NTNU
Norwegian University of
Science and Technology

Optimal Repositioning in Bike Sharing Systems

Hans Martin Espegren
Johannes Kristianslund

Industrial Economics and Technology Management
Submission date: June 2016
Supervisor: Kjetil Fagerholt, IØT
Co-supervisor: Henrik Andersson, IØT

Norwegian University of Science and Technology
Department of Industrial Economics and Technology Management

Preface

This master's thesis is the final work for our Master in Science at the Norwegian University of Science and Technology, Department of Industrial Economics and Technology Management. It was written in spring 2016 and is a continuation of our work for the specialization project done in the fall 2015.

We would like to thank our supervisors Professor Kjetil Fagerholt and Professor Henrik Andersson for invaluable feedback, guidance, and discussions, and Magnus Hausken from Optimeering AS for his support. Further, we would like to thank Russell Meddin, the editor of *The Bike-sharing World Map*, for answering our enquiries and sharing valuable statistics about bike sharing worldwide. We would also like to express our gratitude to Urban Infrastructure Partner and especially Axel, Lars, Johan, and Liisa, kindly providing us with a desk at their office, invaluable data, and the possibility to work in close proximity to the operation of the bike sharing system in Oslo. They have given us a far better understanding of the system through numerous interesting discussions. Finally, we would like to thank Kommunenes Sentralforbund for granting us a scholarship, enabling us to visit the world's largest bike sharing system in Hangzhou, China.

Oslo, 2016-06-08

Hans Martin Espegren & Johannes Kristianslund

Cover photo: Illustration by John Arne Berg for Oslo Bysykkel. Used with permission.

Abstract

This thesis examines the static bicycle repositioning problem (SBRP) and the dynamic bicycle repositioning problem (DBRP) which deal with optimal re-balancing of bike sharing systems (BSSs) during the night and during the day, respectively. The DBRP is an extension of the SBRP where demand throughout the planning period is taken into account. Our analyses show that the demand depends on the location of the stations, the day of week, the time of day, and the weather.

Articles and proceedings discussing the two problems are reviewed in an exhaustive literature survey. To our knowledge, a survey of this scope on the operational level literature has not previously been conducted. We present new and improved mathematical models for the two problems. These models make fewer simplifications than many existing models, e.g. they allow a heterogeneous fleet, multiple visits to each station, and non-perfect re-balancing. The DBRP-model stands out by giving complete routes and specific loading instructions, and by considering the timing of events. It does not however consider the stochasticity in demand.

The models are thoroughly tested on a number of instances that are generated based on the BSS in Oslo, Norway. Testing of the SBRP-model shows that a strengthened MTZ-formulation yields the best results for eliminating subtours, that the symmetry breaking constraints handling visit sequence are effective, and that variable reduction is powerful as it removes symmetry around the depot. For the DBRP-model, the main results from the SBRP-model are applicable. Additionally, we show that the length of the planning horizon influences the routing decision and the value given to key input parameters. The models may serve as decision support together with cost-benefit analyses, for instance when deciding the number of service vehicles and their capacity. Increasing the fleet size and the vehicle capacity improve the objective value for both models. For the SBRP-model, more available time would further improve the solution. The computational effort needed to find the optimal solution is however shown to peak at a problem specific time limit for re-balancing.

The computational effort depends on the number of stations considered and the maximum number of possible visits to each station. While the SBRP-model is solved to optimality within reasonable time for 15 stations, the DBRP-model is only solvable with up to eight stations, indicating that alternative solution methods should be considered. Six rules of thumb (ROTs) for the DBRP are developed, evaluated, and compared with the DBRP-model through simulations. For small instances, the DBRP-model proves effective, while a simple ROT considering the number of times a station fails in meeting customer demand and the driving time seems preferable for larger instances.

Sammendrag

Denne oppgaven studerer det statiske (SBRP) og det dynamiske (DBRP) sykkelflyttingsproblemet. Disse problemene oppstår ved bruk av servicebiler til rebalansering av bysykkelsystemer om natten (statisk) og om dagen (dynamisk). DBRP kan ses på som en utvidelse av SBRP, og tar høyde for etterspørsel. Vi viser at denne etterspørselen avhenger av ukedag, klokkeslett, vær og stasjonenes lokasjon.

I en omfattende litteraturstudie sammenstiller vi tidligere forskning som har blitt gjort på dette området. Så vidt vi vet har det aldri blitt utført en litteraturstudie i denne størrelsesorden tidligere. Vi presenterer to nye, forbedrede matematiske modeller for sykkelflyttingsproblemene som er mer realistiske enn mange tidligere modeller. De tillater blant annet en heterogen servicebilpark, flere besøk til hver stasjon og ikke-perfekt rebalansering. DBRP-modellen skiller seg i tillegg ut ved å gi komplette ruter og spesifikke lasteinstruksjoner, samt ved å ta hensyn til tidsperspektivet. Denne modellen tar imidlertid ikke hensyn til etterspørselsusikkerhet.

Modellene er grundig testet på instanser konstruert basert på bysykkelsystemet i Oslo. Testing av SBRP-modellen viser at en styrket MTZ-formulering er den metoden som fungerer best for eliminering av subturer, at de symmetribrytende restriksjonene for besøksrekkefølge er effektive og at variabelreduksjon er nyttig fordi det fjerner symmetri i tilknytning til depotet. De fleste resultatene fra testingen av SBRP-modellen gjelder også for DBRP-modellen. I tillegg viser vi at planleggingshorisontens lengde påvirker beslutningen om hvilke stasjoner som skal besøkes. Sammen med kost-nytte-analyser, fungerer modellene godt som beslutningsstøtte for systemoperatørene. Flere servicebiler og økt kapasitet på disse forbedrer objektivverdien for begge modeller. I tillegg vil en økt tidsgrense for rebalansering forbedre objektivverdien i SBRP-modellen ytterligere. Det viser seg imidlertid at problemkompleksiteten når et toppunkt for en problemspesifikk tidsgrense.

Problemkompleksiteten avhenger av antall stasjoner i problemet og maksimalt antall mulige besøk til hver stasjon. SBRP-modellen kan løses til optimalitet for inntil 15 stasjoner, mens DBRP-modellen bare kan løses for åtte. Alternative løsningsmetoder må derfor vurderes for å kunne løse problemer av realistisk størrelse. Vi har utviklet seks tommelfingerregler for DBRP. Disse er testet og sammenlignet med DBRP-modellen gjennom tallrike simuleringer. DBRP-modellen fungerer godt for små problemer, mens en enkel tommelfingerregel som vurderer i hvilken grad etterspørselen innfris opp mot kjøretid, viser seg å være effektiv for større problemer.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

As urbanization proceeds throughout the world, public decision makers are looking for effective, affordable, and environmentally friendly means of transportation. Bike sharing fulfills these criteria for short distance traveling within city centres and consequently, bike sharing is getting increasing attention from both governments and the public. In this thesis we study the operation of bike sharing systems (BSSs), focusing on the logistics of the service vehicles used in order to re-balance the system. To get a better understanding of the challenges and possibilities of bike sharing, we start by introducing the concept and its historical development. Further, we present the purpose and the structure of the thesis.

1.1 Background

The last ten years, the number of BSSs worldwide has increased from 17 to 1,029. In this section, we introduce our definition of a BSS, and describe its historical growth and development. Data from the Oslo BSS is extensively used through the thesis and this system is therefore described in some detail.

1.1.1 Bike Sharing Systems

A BSS is a public system for automatic or semi-automatic lending of bicycles (often called *city bikes*) for use within a restricted time period and area. A bike can be lent at one station and delivered at another. This is called A to B service. Note that our definition of a BSS does not include manned (library) systems, bike programs that are A to A or university systems unless automated. Using the bikes may be free of charge, have a fee per hour, or have a monthly or yearly subscription fee. Common for most BSSs is that they are relatively cheap and often partly financed through advertisements on the bikes and the stations.

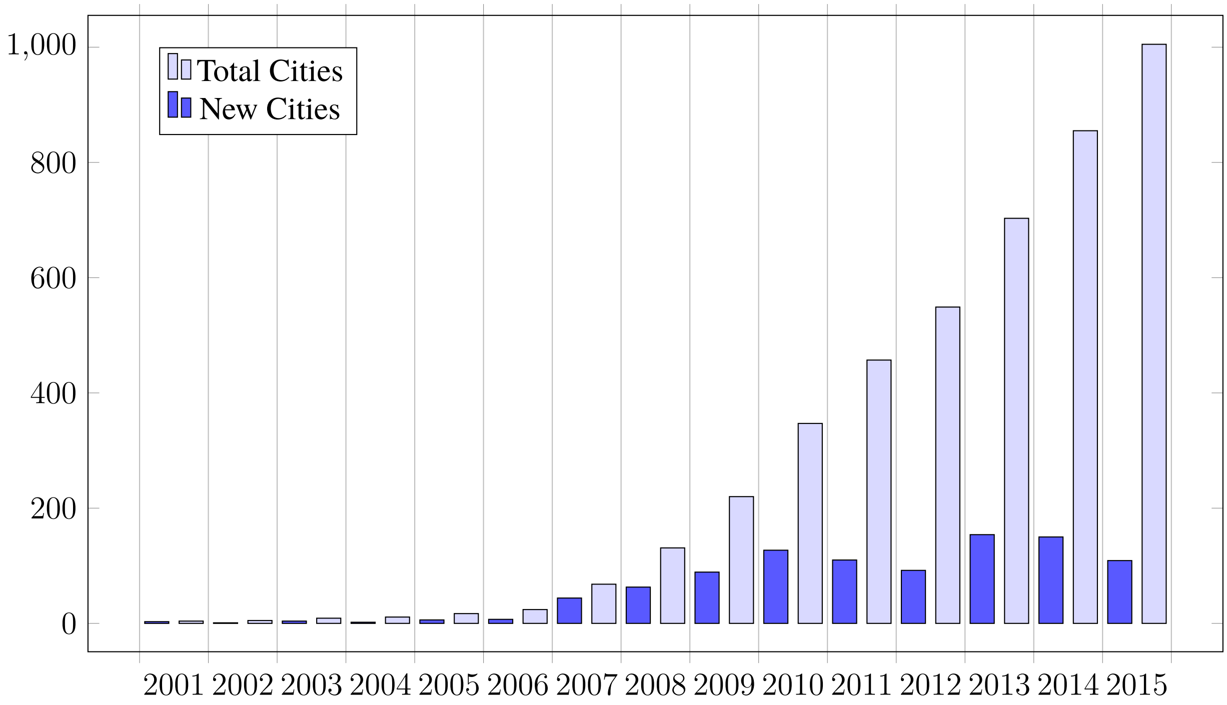
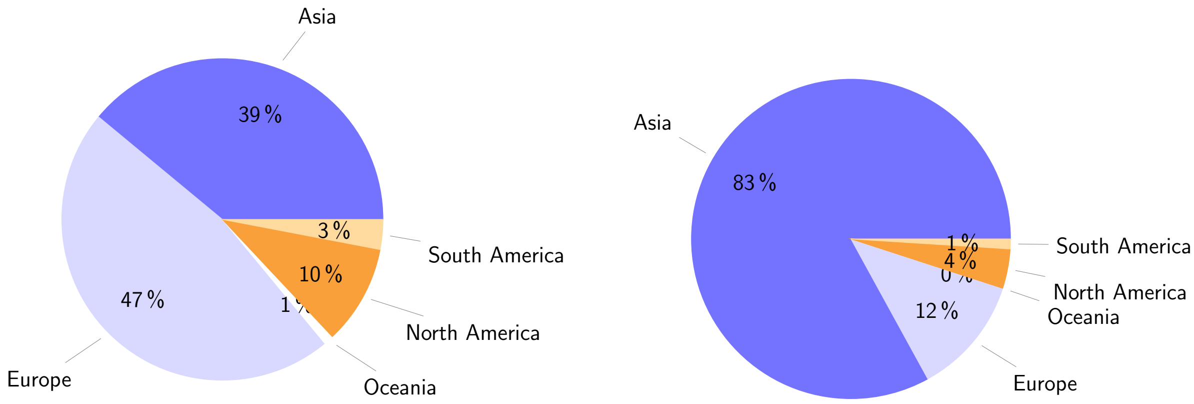


Figure 1.1: Worldwide development in number of cities with a public BSS, 2000-2015 (DeMaio and Meddin, 2016)

Bike sharing is increasingly popular in urban areas around the world. Currently there are 1,029 cities with an active BSS and 319 with a system under planning or construction (DeMaio and Meddin, 2016). Figure 1.1 shows the expansion of bike sharing the recent years, expressed as number of cities in the world with a public BSS.

Public bike sharing emerged in Europe, which remains the continent with most BSSs (Figure 1.2a). In recent years, however, many new systems have been opened in Asia, especially in China. Some of the BSSs in China are very large, which can be seen from Figure 1.2b. The biggest BSS in the world is *Hangzhou Public Bicycle Service* in Hangzhou, China, with more than 84,000 bikes and more than 3,000 stations (DeMaio and Meddin, 2016). The government in Hangzhou plans to expand



(a) Share of cities with BSSs.
Total: 1,029 cities

(b) Share of bikes.
Total: 1,328,100 bikes

Figure 1.2: Distribution of BSSs per continent (DeMaio and Meddin, 2016)

the system to 175,000 bikes by 2020 (Wikipedia, 2015). Figure 1.3 shows the authors of this thesis while testing the BSS in Hangzhou.

Figure 1.3: Testing the BSS in Hangzhou. Picture from private archive.

1.1.2 Four Generations of Bike Sharing

The following section is based on three articles by DeMaio (2009), Shaheen et al. (2010), and Midgley (2011), that independent of each other describe the origination and historical development of bike sharing.

The first bike sharing program began in 1965 in Amsterdam, when a Dutchman named Luud Schimmelpennink painted ordinary bikes white and placed them throughout the city for the public to use. This concept is now known as the first generation of bike sharing, and is characterized by bikes painted in a specific color without locks or stations, free of charge, and with no need for membership or registration. A bike could be picked up where found, ridden for as long as needed and then left unlocked for someone else to use. Despite good intentions, the project was unsuccessful. After just a few days all bikes were either stolen, broken, or thrown in the canals.

It took many years before another bike sharing program evolved. In the beginning of the 1990s in Denmark, a few modest scale projects were staring up in two small cities, before ultimately the famous Bycyklen started up in Copenhagen in 1995. Bringing with it the second generation of bike sharing, two new concepts were introduced. First, the system included docking stations in which bikes had to be locked, borrowed, and returned. Second, to reduce vandalism and theft, customers had to pay a coin deposit when borrowing, which they got back when the bike was safely returned. The bikes were customized for their purpose as city bikes and had spots for advertisements.

Even though the second generation introduced coin deposits, it did not eradicate thefts, seeing as the customers remained anonymous. Hence, the most important contribution of the third generation was to introduce a form of membership, often including a magnetic card used to borrow bikes at the stations. This made it possible to keep track of each borrower, and combined with personal information and credit card information, theft became much less attractive. A couple of small membership-based bike sharing programs existed before 2005, but it was with the system in Paris called *Vélib'*, which opened 2005, that the third generation got a foothold. *Vélib'* was a huge success and got massive attention worldwide. Indeed, this is presumed to be the main reason why the popularity of bike sharing exploded after 2005 (Figure 1.1).

One may argue that a fourth generation of bike sharing has evolved in recent years. This generation makes use of smart phones and Internet, and may be considered a part of the development towards the so-called *Internet of things* (Morgan, 2014). The docking stations, and even each bike, may be connected to the Internet, sending real time information to both users and operators through smart phone applications. Movable docking stations, solar-powered docking stations, and electric bikes are all possible extensions in the fourth generation. Again, Copenhagen has taken on an important role with their new version of Bycyklen, launched in 2015. This system has 1,860 electrical ”smart bikes” equipped with a tablet at the handlebar. Using the tablet, you can log on to the bike, accessing your private information. It also sports a built-in navigation, real-time integration to the public transportation network, and many other functions (Go Bike, 2014).

In Table 1.1, the different generations of bike sharing systems are listed and compared by their main characteristics. Note that there has been a smooth transition between the third and fourth generation, and in some cases it may be hard to differentiate between these two. Today the third and fourth generation are prevalent. The second generation is almost completely gone, and examples of first generation may only be found in some restricted areas such as campuses and parks.

Table 1.1: Specifics of the different bike sharing generations (DeMaio, 2009), (Shaheen et al., 2010), and (Midgley, 2011)

1st Generation	**2nd Generation**	**3rd Generation**	**4th Generation**
1965	1995	2005	2014
• Distinguishable bikes	• Distinguishable bikes	• Distinguishable bikes	• Distinguishable bikes
• Unlocked bikes	• Docking stations	• Docking stations	• Docking stations
• Free of charge	• Small deposits	• Price per hour and/ or subscription fee	• Price per hour and/ or subscription fee
		• Membership and tracking of usage	• Membership and tracking of usage
			• ”Smart bike”, apps, real time availability
			(• Electrical bikes)

1.1.3 Bike Sharing Systems in Norway

Today there are five cities in Norway with an active BSS, see Table 1.2. The largest BSS in Norway is the one in Oslo, currently consisting of 800 bikes and 100 stations (Høgåsen-Hallesby, 2016). The rights to operate the systems in Trondheim, Oslo and Drammen are contracted to Clear Channel Norway AS (Clear Channel Norge, 2015). In Lillestrøm, another outdoor advertiser, JCDecaux Norge AS, owns the rights to operate the system (DeMaio and Meddin, 2016). In Stavanger, Forus Næringspark has developed a system with electrical bikes, similar to the one in Copenhagen (Andersen, 2015). In all five cities, the municipalities donate spots for the stations and are in close cooperation with the operators. The new systems in Oslo and Stavanger are a part to the forth generation of bike sharing, while the remaining BBSs in Norway could be classified as third generation systems, in accordance with Table 1.1. Figure 1.4 pictures a station in the BSS in Oslo.

The BSS in Oslo was renewed just before the season of 2016, and now offers new bikes, stations, service vehicles, app, and subscription-system. The bikes are lighter, faster, and more comfortable, and the app shows real-time availability of bikes and locks and lets the users unlock bikes directly from the mobile phone. The capacity of the service vehicles has more than doubled from 10 to 23 bikes. In 2015, more than one million trips were made throughout the year with the city bikes in Oslo (Oslo Bysykkel, 2016). This equals about 5,000 trips every day and implies that an average bike is used between three and four times per day. As of June 8th 2016, more than 538,000 trips

Table 1.2: Bike sharing in Norway
*The BSS in Oslo is expanding and will be finished in April 2017. The first numbers represent the current size, the seconds are for April 2017.
**The BSS in Stavanger is under development and consists of electrical bikes

City	# stations	# bikes
Trondheim	20	140
Lillestrøm	5	50
Oslo*	100/300	800/3,000
Drammen	16	150
Stavanger**	13/50	50/300

are made, indicating that the usage is nearly doubled compared to the year before. On the peak day of May 25th, 13,033 trips were completed, resulting in an average of 15 trips per bike. Moreover, before the system reopens April 1st 2017, it will undergo a major expansion and triple in size to about 3,000 bikes and almost 300 stations (Høgåsen-Hallesby, 2016). Urban Infrastructure Partner owns and operates the system, ShareBike delivers the bikes and IT-system, while Clear Channel will only be doing matters related to advertising (Oslo Bysykkel, 2016).

Figure 1.4: A full station at Paléhaven in Oslo. Picture from private archive.

1.1.4 The Service Vehicles

For the system to function well, it is crucial that bikes are available at the stations when someone wants to lend, and that there are free slots available when someone wants to return a bike. To achieve this, most BSSs use service vehicles to re-balance the system, i.e. move bikes from (nearly) full stations to (nearly) empty stations. Figure 1.5 shows such a service vehicle with a capacity of 23 bikes. In most BSSs, the re-balancing happens both during the day, when the bikes are used, and during the night when the system is closed. While re-balancing during the day is a dynamic problem, the overnight re-balancing is considered static.

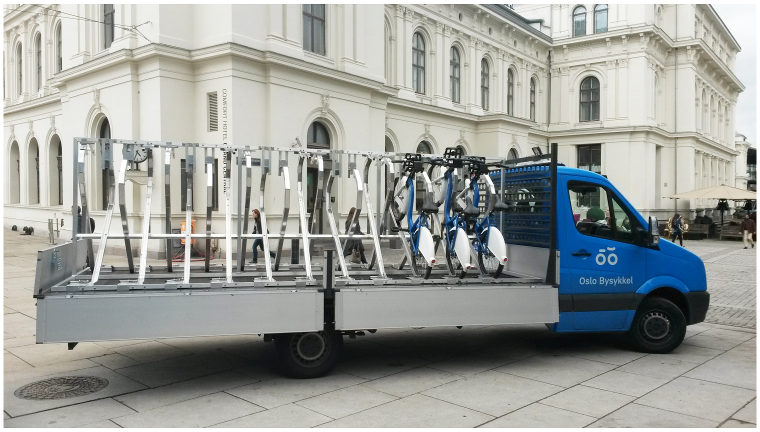


Figure 1.5: A service vehicle in front of Østbanehallen in Oslo. Picture from private archive.

1.2 Purpose

Numerous challenges in relation to planning and operation of BSSs could be modeled and solved by the use of mathematical programming. One of these is the re-balancing problem, i.e. finding routes for the service vehicles, as well as pickup and delivery quantities at each station, so that the system is well positioned to meet customer demand. Finding better solutions to the re-balancing problem can save resources for the operators. They may achieve more effective use of their limited resources, such as bikes and service vehicles. Consequently, customers would benefit from reduced problems with full or empty stations. Overall, this can contribute to increased utilization of bike sharing worldwide and thus lead to greener cities and healthier populations.

This thesis uses the BSS in Oslo as a base case. At present, the use of mathematical tools for decision support for re-balancing in Oslo and in most BSSs are nearly absent. There is however an increased interest in the use of optimization to support the system operators. As solving the re-balancing problem is a quite novel field in operations research, many of the existing models lack realism, and a survey of all the literature on the topic is missing in the literature.

The main objectives of this thesis is threefold. 1) to present an exhaustive literature survey on the optimization problems arising when re-balancing BSSs, including a comparison of the existing models, 2) to propose new mathematical models of the problems that capture more real-life aspects, and 3) to compare and evaluate simple rules of thumb proposed by the system operator to determine the next station visit.

1.3 Structure of the Report

This thesis elaborates on the optimization of bike repositioning in order to re-balance BBSs. Our previous work on the same issue resulted in an article named *The Static Bicycle Repositioning Problem - Literature Survey and New Formulation*, which is submitted to the 7th International Conference on Computational Logistics (ICCL). This article is found in Appendix B and is referred to throughout the thesis to avoid needless repetition.

Chapter 2 provides a study of the literature utilizing optimization models for different aspects of bike sharing. The main topic of this thesis is bike repositioning, and all prior research available on this topic is presented. To facilitate the reading of the quite extensive survey, the articles and proceedings are listed in comparative tables.

To get a deeper understanding for the necessity of re-balancing, a short analysis of the demand for bikes and locks is presented in Chapter 3, using the BSS in Oslo as a basis. Chapter 4 contains a detailed description of the problems at hand. Different aspects of the problems are elaborated, and the main restrictions, decisions, and assumptions are pointed out. In Chapter 5, the mathematical models are presented with all necessary constraints. Symmetry breaking constraints and valid inequalities are suggested.

The mathematical models are tested on a handful of instances. Different formulations, constraints, and extensions of the mathematical models are thoroughly tested, and both the quality of the solution and the computational time are evaluated and discussed. Chapter 6 presents the test instances and the key input-data, while the results are provided in Chapter 7. The latter also illustrates how the models can be used as decision support for the system operators.

Both models, and in particular the model for through-the-day re-balancing, have difficulties with solving real-sized instances. Therefore, simple methods for determining the next station visit for the DBRP are proposed by the system operator in Oslo. In Chapter 8, these methods are compared to the mathematical model, and tested and evaluated through a simulation. Finally, the main findings and conclusions are presented in Chapter 9, and suggestions to further work listed in Chapter 10.

CHAPTER 2

LITERATURE SURVEY

A vast amount of research on BSSs has been published throughout the last decade. This includes a significant number of studies using operational research (OR) and optimization. To lay the foundation for the problem at hand, we will in this chapter elaborate on the literature applying OR and optimization to BSSs. The focal point of this survey is the research on the re-balancing operations.

2.1 Concepts

Before presenting the literature, a few concepts are introduced. The first concept is *state*, i.e. a distribution of bikes throughout the system, expressed as a specific number of bikes at each station. The *optimal state* is the desired distribution of bikes at the end of the planning period, while the *initial state* is the distribution at the beginning of the planning period. After re-balancing the system using the service vehicles, we get the *final state*. The difference in number of bikes between the final and the optimal state is called *deviation*, and total deviation is the sum of deviations over all stations. Every time the system is unable to meet the demand, we get a *violation*, i.e. the event that there are no bikes available when someone wants to lend, or that there are no free slots when someone wants to return a bike.

2.2 Planning Levels

We divide the possible optimization problems regarding BSSs in three levels, in accordance with Vogel et al. (2015); a strategic, a tactical and an operational level. This classification is illustrated in Figure 2.1. The strategic level contains problems arising when constructing the system, including finding the optimal number and size of stations, bikes, and vehicles in a city, as well as finding the best locations for the stations. On the tactical level, the configuration of the system is considered given, and relevant problems include finding the optimal state. The operational level deals with the specific routes and the loading and unloading quantities to re-balance the system. The literature study is most extensive for the operational level, as this is the focal problem of this thesis.

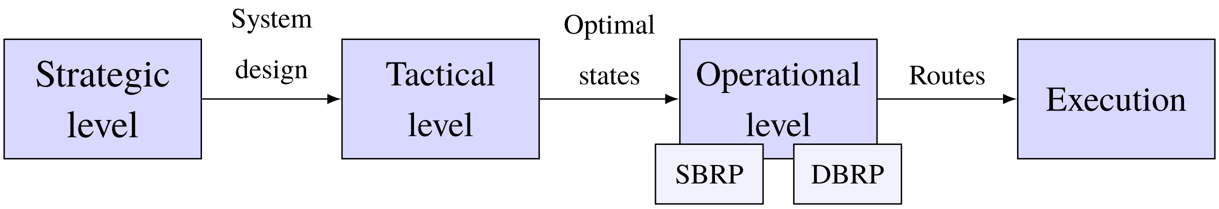


Figure 2.1: Planning levels of BSS optimization

2.3 The Strategic Level

On the strategic level there are three main problems to be solved. The first is to find the optimal number of docking stations and their locations. García-Palomares et al. (2012) use a GIS-based method to calculate the spatial distribution of the potential demand, and locate stations using location–allocation models. Romero et al. (2012) present a methodology to simultaneously model private car and public bicycle transport modes and their interactions. They use this method to optimize the location of docking stations in a BSS. Lin and Yang (2011) create a model that attempts to determine the number and location of docking stations, using a network structure of bike paths connected between the stations and travel paths for users between each pair of origins and destinations. Dell'Olio et al. (2011) analyse the personal profiles of potential system users. This information is applied to find demand patterns, optimal tariffs to charge and optimal locations for the stations.

The second main problem on the strategic level is to find the optimal number of bikes in the system, and the number of slots at each station. Fricker and Gast (2014) study the stations where most problems arise in the BSS in Paris called *Vélib'*. They quantify the influence of changing the station capacities and compute the optimal number of bikes in the system in terms of minimizing the proportion of problematic stations. A problematic station is a station where violations occur regularly. Fricker and Gast (2014) conclude that the total number of bikes that corresponds to the best performance is half of the total number of slots, plus a few more depending on system parameters. The third main problem is to detect the ideal number of service vehicles, but no studies are found that address this issue.

To solve the strategic level problems, a thorough analysis of the demand patterns is needed. This can be done using both demographic data (information about where people live, work etc.) and historical data from public transport or an existing BSS.

2.4 The Tactical Level

The objective on the tactical level is to find the desired distribution of bikes between the stations at any given time, i.e. finding the optimal state. When solving this problem, the parameters from the strategic level are considered given, i.e. it is impossible to change the number and location of stations, the number of bikes in the system, or the number of slots at each station. To find good solutions on the tactical level it is crucial to make accurate forecasts of the demand. Because of demand uncertainty, the solution is sometimes given in intervals, e.g. the optimal state at a given station is 5-8 bikes. The solution from the tactical problem is necessary to make good decisions at the operational level.

Vogel et al. (2011) use data mining and knowledge discovery in databases to gain insight into the bike activity patterns and use this information to find the optimal state. Raviv and Kolka (2013) introduce an inventory model for managing inventory at bike sharing stations. They create a *user dissatisfaction function* (UDF) based on the occurrence of violations. Using discrete time periods and a Poisson process gives an approximation of the UDF that can be used to find optimal inventory levels. They also propose that the model may be a building block in a greater optimization model.

Schuijbroek et al. (2013) model stochastic demand using a queue system and define service level requirements at each station. The output of the model is intervals for optimal state to obtain the desired service level. Further, they combine their inventory model with a vehicle routing model, hence combining two problems that usually are solved separately. Parikh and Ukkusuri (2015) develop a mixed integer program (MIP) model for finding the optimal state. The objective is to minimize the expected penalty, which is calculated as a function of expected violations for all stations over the planning horizon. Like the other studies, they assume that the demand at one station is independent of the others. With this simplification, the model is solved to optimality for instances of realistic size.

O'Mahony and Shmoys (2015) analyze the best placement of bikes utilizing system usage data. They cluster the stations with similar kind of demand together, to later label each cluster with a desired level of bikes. For example self-balancing stations are put together. A self-balancing station maintains a relatively constant amount of bikes throughout the day, because demand and supply balance each other out. The clustering is done to decrease the amount of necessary computations.

The tactical level also contains problems regarding the detection of broken bikes in the system. This is not a widely explored area, but Kaspi et al. (2015) have developed a model that computes the probability that a given bike is unusable, based on the time since the last rental, other available bikes at the station etc.

2.5 The Operational Level

Before elaborating on the literature on the bicycle repositioning problems at the operational level, a general classification of routing problems is presented, and different groups of problems introduced.

2.5.1 Routing Problems

In accordance with both Toth and Vigo (2014) and Psaraftis et al. (2016), we classify Vehicle Routing Problems (VRPs) as either *static* or *dynamic*, and *stochastic* or *deterministic*. Psaraftis et al. (2016) define a VRP to be dynamic if the input of the problem is received and updated concurrently with the determination of the route. If all problem inputs are received before route determination, the VRP is static. A VRP is deterministic if all inputs are known with certainty and there are no stochastic inputs. The stochastic VRP on the other hand, is characterized as a VRP where one or more of the parameters are stochastic, meaning that some future events are represented by random variables with known probability distributions.

This classification gives four possible combinations, illustrated in Table 2.1. Based on a thoroughly description of the different combinations by Toth and Vigo (2014), a short resume is given in the table.

Another important characterization is done by Ritzinger et al. (2015), who separate the dynamic and stochastic VRPs further into two groups. The first group, *preprocessed decisions*, uses approaches where solutions are computed before the execution of the plan. The second group, *online decisions*, consists of approaches where solutions are computed as soon as a dynamic event occurs.

Table 2.1: Classification of VRPs

	Deterministic	**Stochastic**
Static	*Static and Deterministic.* All input parameters are known with certainty and in advance, and are assumed not to change during operation. The problem may be solved once and before the beginning of the planning period	*Static and Stochastic.* Some input parameters are random or stochastic, and the actual values are revealed during the execution of the routing process.
Dynamic	*Dynamic and Deterministic.* Some or all input data are not known beforehand, but become available over time. Only probabilistic information is available for future events. Optimization can only be performed as new information arrives.	*Dynamic and Stochastic.* Dynamic problems where part of the unknown input data is in the form of stochastic information (e.g., forecasts, range values, and prescribed distributions).

A survey of VRPs in general is not done here, but note that dynamical VRPs may, as described above, be separated into deterministic and stochastic problems. Thoroughly literature surveys have been conducted on both kind of problems, and we refer to Toth and Vigo (2014), Ritzinger et al. (2015), Psaraftis et al. (2016), Berbeglia et al. (2010), and Pillac et al. (2013) for extensive elaborations on the subject.

In the context of BSSs, the operational level deals with the actual transportation and relocation of bikes by the service vehicles. This operation is in the literature referred to as both repositioning and re-balancing. A solution contains optimized routes for all service vehicles, including the number of bikes that should be picked up and delivered at each station. Using graph theory, each station could be considered a node, and the roads between them arcs. It is common to divide the operational level in two: static and dynamic problems, in accordance with the classification above. Raviv et al. (2013) name the two problems *static bicycle repositioning problem* (SBRP) and *dynamic bicycle repositioning problem* (DBRP). In SBRP the system is considered static, while in DBRP the demand during the operating time is taken into account. The DBRP may be modeled as either deterministic or stochastic. In the following, the literature on the SBRP and the DBRP are examined.

2.5.2 The Static Bicycle Repositioning Problem

The SBRP is typically used for overnight re-balancing, and the problem has to be solved once at the beginning of every night. In most cities, the demand at night is very low, or the system is closed. It is therefore not necessary to take the demand forecast for the operating period into account; the problem is static and deterministic. Hence, the SBRP can be classified as a *static many-to-many one-commodity pickup and delivery problem with selective pickup and selective delivery*, according to Berbeglia et al. (2007). This variant of the pickup and delivery problem (PDP) is not known to have been studied outside the BSS-literature, but it bears similarities to the one-commodity pickup and delivery traveling salesman problem (1-PDTSP) introduced by Hernández-Pérez and Salazar-González (2003). There are three main differences between the SBRP and the 1-PDTSP; the SBRP allows use of more than one vehicle and splitted loads, it has selective pickup and selective delivery, and the objective of the SBRP is not always to minimize costs, but might as well be to minimize deviations.

A comparison of the studies on the SBRP is presented in Table 2.3, and the studies corresponding to the numbers in the table header are listed in Table 2.2. For an extensive literature survey on the SBRP, the reader is referred to our article found in Appendix B.

Table 2.2: Articles overview for Table 2.3

1	Benchimol et al. (2011)
2	Chemla et al. (2013)
3	Raviv et al. (2013)
4	Rainer-Harbach et al. (2013a) & (2013b)
5	Schuijbroek et al. (2013)
6	Ho and Szeto (2014)
7	Dell'Amico et al. (2014)
8	Sörensen and Dilip (2014)
9	Gaspero et al. (2015)
10	Erdoğan et al. (2014)
11	Erdoğan et al. (2015)
12	O'Mahony and Shmoys (2015)
13	Forma et al. (2015)

Table 2.3: Comparison of SBRP articles. The numbers in the table header correspond to the studies in Table 2.2

Parameter	1	2	3	4	5	6	7	8	9	10	11	12	13
Multiple vehicles	No	No	Yes	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Yes
Heterogeneous fleet	n/a	n/a	Yes	Yes	Yes	n/a	n/a	No	Yes	n/a	n/a	No	Yes
One vehicle can make multiple visits to a station	Yes	Yes	No/Yes two models	Yes	Yes	No	No	No	Yes	No	Yes	No	No
Several vehicles can visit the same station	n/a	n/a	Yes	Yes	Yes	n/a	n/a	No	Yes	n/a	n/a	No	No
Modeling loading / unloading	No cost for l/unl.	No cost for l/unl.	Depend on quantity	Average time for st.	No cost for l/unl.	Depend on quantity	No cost for l/unl.	Time for each request	Average time for st.	Depend on quantity	No cost for l/unl.	Average time for st.	Depend on quantity
Selective pickup/delivery	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Stations used as temporal inventories	No	Yes	Yes	No	Yes	No	No	No	Yes	No	Yes	No	Yes
Allows non-perfect rebalancing	No	No	Yes	Yes	No	Yes	No	Yes	Yes	No	No	Yes	Yes
Objective function	Min. cost	Min. cost	Min. dev. and cost	Min. dev & operations	Min. longest route	Min. penalty func.	Min. cost	Min. rejected requests	Min. dev. and cost	Min. time and cost	Min. cost	Max. rebal. nodes	Min. dev. and cost
Sub-tour elimination	Many new var. & cons.	Many new var. & cons.	MTZ / Time	Heuristics	Time	MTZ	MTZ / Sep. & cut	Time	Time / Route	Separation & cut	Separation & cut	A variant of MTZ	MTZ
Solution method	n/a	B-&-C & Tabu search	Heuristics & exact	PILOT, GRASP, VNS	Cluster first route second	Tabu search	Exact: B-&-C	n/a	LNS	Exact: Benders' decomp.	Exact: Benders' cut	Heuristics & exact	Cluster first route second
Size of solvable instances	n/a	1v., 100st.	1v., 60st.	21v., 700st.	5v., 135st.	1v., 400st.	1v., 50st.	n/a	6v., 240st.	1v., 50st.	1v., 60st.	5v., 100st.	3v., 200st.
Based on article (table header as reference)	n/a	1	n/a	n/a	n/a	3	n/a	n/a	4	n/a	2 & 9	n/a	3

2.5.3 The Dynamic Bicycle Repositioning Problem

DBRP is also a problem regarding the re-balancing of the system, but where demand during operating time is taken into account. An important aspect of the DBRP is to prioritize which bikes to move first.

The first to address the DBRP was Contardo et al. (2012), that describes the problem using a time-space network. Considering deterministic demand, the objective is to minimize the number of violations during the planning period. To solve real world instances, they propose an iterative solution procedure using both Dantzig-Wolfe and Benders decomposition.

Caggiani and Ottomanelli (2012, 2013) propose a decision support system based on neural network. The system both forecasts the demand in each time interval and finds an optimal relocation matrix. Their first article uses variable time intervals, while the second uses constant intervals. Constant intervals prove to be better in situations with low demand, while variable intervals are better with increasing congestion. The length of an interval is typically 10 minutes, and they assume that the service vehicle makes one movement within one time interval.

Nair (2010), Nair and Miller-Hooks (2011), and Nair et al. (2013) solve the DBRP for the Vélib'-system in Paris by simplifying it to a static and stochastic problem for the re-balancing. The model is at an aggregated level, i.e. it does not consider actual routing. Demand is represented using a random variable with known probability distribution. The objective is to find the re-balancing operations that minimize the planned number of violations, as well as the cost of re-balancing, given that the actual number of violations does not exceed the planned number of violations with a given probability.

A decomposition method is introduced by Sörensen and Dilip (2014), consisting of a request generation algorithm and a bike request scheduling problem (BRSP). The request generation algorithm uses various data to generate repositioning requests. However, the details of this algorithm are not included in the study. A request includes the location and number of bikes to be picked up or delivered, a time window, and an importance weight. The BRSP determines which requests to execute, and assigns them to vehicles. The objective is to minimize the total weight of rejected requests, without considering the actual routing.

Vogel et al. (2014) model the DBRP by disctretizing in quite long time periods of for example one hour. The number of bikes to move in each period are determined. The model provides routing for each time period, but there is no connection between the routing in the different periods. The article also introduces a hybrid metaheuristic, integrating a large neighborhood search (LNS) with exact solution methods. This hybrid metaheuristic worked well on the quite small Vienna BSS (59 stations, 627 bikes), but the authors claim it might be challenging to apply it to larger instances.

Kloimüllner et al. (2014) model the DBRP using continuous time. The objective is to find effective routes that minimize violations during the day, as well as deviations from the optimal state at the end of the day. The focus of the article is to develop and compare different construction heuristics. They conclude that a preferred iterative look ahead technique (PILOT) heuristic is useful when solutions are needed quickly, while a variable neighborhood search (VNS) is better for longer runs.

Angeloudis et al. (2014) split the problem into two sub-problems; a multiple traveling salesman problem (m-TSP) determining the routes, and a flow assignment problem for each vehicle determining the pickup and delivery quantities. Through iterations between the two problems, they find the shortest, feasible tour that minimizes total deviation. They study a quite long time period, and do not consider the timing of demand and re-balancing events within this period. By using m-TSP, all stations have to be visited exactly once.

Seo et al. (2015) use a Markov decision process to model stochastic demand. There is only one vehicle, which has infinite capacity, and during one time step the vehicle can move from one station to another. Thus, the model does not appear to be complete with respect to routing. The objective is to minimize a non-linear penalty function.

Regue and Recker (2014) split the DBRP into four sub-problems. The first two problems are at the tactical level and are typically solved once every week. These problems provide a demand forecast and an optimal inventory interval for each station in every time period. Time periods are typically 20 minutes. Further, they present a sub-problem determining the reallocation needs using the optimal inventory levels and demand forecast. This sub-problem minimizes expected violations in the following periods, given that the probability of a violation at each station is below a certain level. Lastly, they use a VRP to find the routes for each vehicle, maximizing the utility of the visited stations in the current period. The reallocation needs problem and the VRP are resolved every time a vehicle finishes its route at the end of a time period.

O'Mahony and Shmoys (2015) classify all stations as either consumers, producers or balanced nodes, and pair up consumer and producer-nodes in a pairing problem. Their model does not take demand or time into account, but minimizes the longest distance between a given number of pairs.

Brinkmann et al. (2015a) present a complete model of the DBRP, using an arc-formulation and discrete time-intervals. The time intervals are short, implying that the vehicles drive from one time-space location to another. Demand is considered deterministic, and the objective is to minimize the squared gap between the final state and target interval at each station for each time interval. The instances are solved both by a decomposition approach and a heuristic.

Brinkmann et al. (2015b) propose a short term relocation (STR) strategy looking only one move ahead. The strategy states that each vehicle should always visit the nearest, feasible, unbalanced station. Using simulation, the STR strategy is compared with a heuristic from Brinkmann et al. (2015a) (referred to as a long term relocation (LTR) strategy). While the LTR uses target fill levels to make decisions, STR prioritizes stations with a certain urgency and therefore a high risk of immediate violations. They conclude that the STR strategy outperforms the LTR strategy.

2.5.4 Comparison of DBRP Studies

Table 2.5 shows a comparison of the studies on the DBRP. The studies corresponding to the numbers in the table header are listed in Table 2.4. From the table we see that six studies present models for preprocessed decisions, while four studies focus on online decisions. Two studies (Nair et al. (2013) and Brinkmann et al. (2015a)) provide models that allow both preprocessed and online decisions. Note that all models for preprocessed decisions could be re-solved at any time to get an updated route, but in general it is not intended to solve these models often.

Table 2.4: Overview of the studies from Table 2.5

1	Contardo et al. (2012)
2	Caggiani and Ottomanelli (2012)
3	Nair et al. (2013)
4	Sörensen and Dilip (2014)
5	Vogel et al. (2014)
6	Kloimüllner et al. (2014)
7	Angeloudis et al. (2014)
8	Seo et al. (2015)
9	Regue and Recker (2014)
10	O'Mahony and Shmoys (2015)
11	Brinkmann et al. (2015a)
12	Brinkmann et al. (2015b)

Only three studies provide truly stochastic models. Nair et al. (2013) and Regue and Recker (2014) both consider a random demand with a known probability distribution, while Seo et al. (2015) use a Markov chain to model pickup and delivery at the stations. Of the remaining articles, six use

Table 2.5: Comparison of DBRP studies. The numbers in the table header correspond to the studies in Table 2.4

Parameter	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Preprocessed / Online	Preprocessed	Online	Preprocessed / Online	Online (for each new request)	Preprocessed	Preprocessed	Preprocessed	Preprocessed	Online	Preprocessed	Preprocessed / Online	Online. Solved at every node.
Modeling demand	Expected value	Expected value	Random var., known prob. dist.	Request generating algorithm	Expected value	Expected value	Expected value	Markov chain. Stoch.	Random var., known prob. dist.	No demand	Expected value	Future demand neglected
Modeling time	Discrete (time-space-network)	Discrete (short intervals)	Discrete (long intervals)	Continuous	Discrete (long intervals)	Continuous	Discrete (long intervals)	Continuous	Discrete (long intervals)	No time	Discrete (time-space-network)	No time
Re-balancing	Start each re-balancing operation in one time interval, end in another	One re-balancing operation within each time interval	Multiple re-balancing operations within one time interval	Continuous time	Multiple re-balancing operations within one time interval	Continuous time	Multiple re-balancing operations within one time interval	Continuous time	Multiple re-balancing operations within one time interval	n/a	Start each re-balancing operation in one time interval, end in another	n/a
Objective function	Min. violations	Min. relocation costs and lost users cost	Min. relocation costs and violations	Min. weight of unserved requests	Min. weight of # bikes moved, time usage and deviations	Min. weight of violations, deviations, handling and time	Min. time. Min. deviations through sol. algorithm	Min. penalty function and time used	(1) Min. dev. for all periods(weighted) (2) Max. utility of visits	Min. maximal distance between any pair	Min. deviation squared	Go to closest, feasible, unbalanced station
Complete routing	Yes	Yes	No	Yes	No	Yes	Yes	No	Yes	No	Yes	No
Multiple vehicles	Yes	Yes	n/a	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Yes
Multiple visits to a station	Yes	No	n/a	Yes	Yes	No	No	No	Yes	No	Yes	n/a
Modeling handling	No	No	n/a	Time for each request	Yes	Yes	Yes	No	No	No	No	No
Selective pickup/delivery	Yes	Yes	Yes	No	Yes	Yes	Yes	n/a	Yes	No	Yes	Yes
Solution algorithm	Iterative procedure, Dantzig-Wolfe & Benders	Branch & bound	Various solution methods to handle non-linearity	n/a	Both exact and using LNS	Greedy construction heuristic, PILOT, VND and GRASP	Exact solution. Iterative algorithm	Exact	Exact solution, two sub-problems	Exact and greedy heuristic	Decomposition and VNS	STR heuristic
Size of solvable instances	100 st., 5 v., 60 periods	5 st., 1 v., 3 days	n/a	n/a	59 st., 1 v., 24 periods	90 st., 5 v., 8 hours	30 st., 4 v., 3 hours	10 st.	61 st., 2 v.	100 st., 5 v.	59 st., 2 v.	59 st., 2 v.
Contribution	First study on DBRP	Use of neural network to forecast demand	Modeling stochastic demand	Request algorithm approach	Hybrid meta-heuristic	Construction heuristics	Two sub-problems: m-TSP and flow assignment	Markov chain	Two sub-problems: Reallocation needs and VRP	Pairing of producer and consumer stations	Realistic/ complete model	Short term relocation strategy (STR)

expected values, two neglect future demand (O'Mahony and Shmoys (2015) and Brinkmann et al. (2015b)), and Sörensen and Dilip (2014) assume that future demand is taken care of by their request generating algorithm.

Time could be modeled in different ways, and the chosen method greatly affects the mathematical formulation. Three studies use continuous time, where time is a variable, and two studies do not consider time at all. The remaining studies consider discrete time, with either long, short, or very short time intervals. When the time intervals are very short (Contardo et al. (2012) and Brinkmann et al. (2015a)), e.g. one minute, the graph is considered a time-space-network where the arcs go between different time-space nodes. This modeling method results in a huge amount of nodes and arcs in the graph. Caggiani and Ottomanelli (2012) present the only article making use of short intervals, e.g. 10 minutes, and assume that each service vehicle manages to do exactly one re-balancing operation during the interval. Using this method, they assume that the driving times between all pairs of stations are equal. The remaining four studies using discrete time, use long time intervals, e.g. one hour. Here, the service vehicles may do multiple re-balancing operations during each time period. When choosing long time intervals, it is not possible to control whether demand occurs before or after the station is visited by a service vehicle in the course of one interval. Further, it could be argued that when using long time intervals in a dynamic setting, it is only worth considering the first interval, and hence neglecting time.

When considering the objective function, no studies are identical. Typically, violations and/or deviations are minimized, often in combination with time usage. Contardo et al. (2012) and Nair et al. (2013) minimize violations, while Vogel et al. (2014), Angeloudis et al. (2014), Regue and Recker (2014), and Brinkmann et al. (2015a) minimize deviations. Kloimüllner et al. (2014) minimize both violations and deviations and give them different weight. Seo et al. (2015) use a non-linear objective function depending on the number of bikes at each station, while the remaining four studies use objective functions that are only slightly related to the number of bikes at the stations. Note that in the models only minimizing deviations, violations can occur, but are not treated differently than deviations. The models only minimizing violations do not consider an optimal state at all.

Five of the studies do not include complete routing, e.g. the models are incomplete with respect to sub-tours. In these studies, the routing is typically handled by the solution algorithm. Three studies allow only one vehicle, and as little as four studies take the time/cost for handling into account. Five studies allow multiple visits to each stations, while the others do not. Allowing only one visit to each station may be a severe simplification, depending on the length of the planning horizon.

Two models do not allow selective pickup and delivery. In Sörensen and Dilip (2014), each request must either be served completely or not at all, while O'Mahony and Shmoys (2015) do not consider loading quantities at all.

All but two studies (Nair et al. (2013) and Sörensen and Dilip (2014)), include computational experiments on either theoretical or real instances. The majority uses some kind of heuristic to solve the instances. The studies that use exact methods, fail to find the optimal solution when the problem size increases and only yield upper and lower bound solutions. Angeloudis et al. (2014) split the problem into two sub-problems; a m-TSP and a flow assignment problem, allegedly making it easier to solve. Another splitting into sub-problems is attempted by Regue and Recker (2014), who split the problem into a reallocation needs-problem, and a VRP.

CHAPTER 3

DEMAND ANALYSIS

A challenge for all BSSs is the fluctuating demand for both bikes and locks. Understanding what drives the demand and enabling to make accurate forecasts is essential for satisfactory re-balancing. In this chapter, we look at how historical data may be used to estimate important parameters, and how they may be applied in the practical operation of a BSS.

3.1 Analysis of Historical Data

Historical data for all bike trips and inventory levels at all stations are available for the system operators. In this section, we analyze historical data from the BSS in Oslo, discuss how some essential system parameters affect the demand, and demonstrate the necessity of re-balancing.

3.1.1 System Usage

The system usage varies in the course of a day. Figure 3.1 shows the number of bikes in use on a daily basis, averaged over four weeks in May 2016. Weekdays and weekends are shown separately, as the usage typically differs. Analyses show that the different days within the two groups internally are very similar. Even though the values are averaged over a period of four weeks, the curves are not smooth, mainly because of missing data points. The tendency is however clear, and the morning peak on weekdays stands out as the main difference between the two, showing that fewer people work in the weekends. Another observation is that the afternoon peak is higher than the morning peak, which probably could be explained by two factors. First, the usage is more evenly distributed throughout the system in the afternoon, resulting in fewer empty stations and a better ability to meet customer demand. In the mornings, the vast majority of users ride the bikes from the residential areas down to the city center, resulting in many empty stations and a limited supply of bikes to the potential users. Secondly, the average trip length is longer in the afternoon than in the morning, resulting in many simultaneous trips. Preliminary analyses indicate that the average trip in the afternoon is three times longer than in the morning. The graph in Figure 3.1 must therefore not be confused with the number of trips completed.

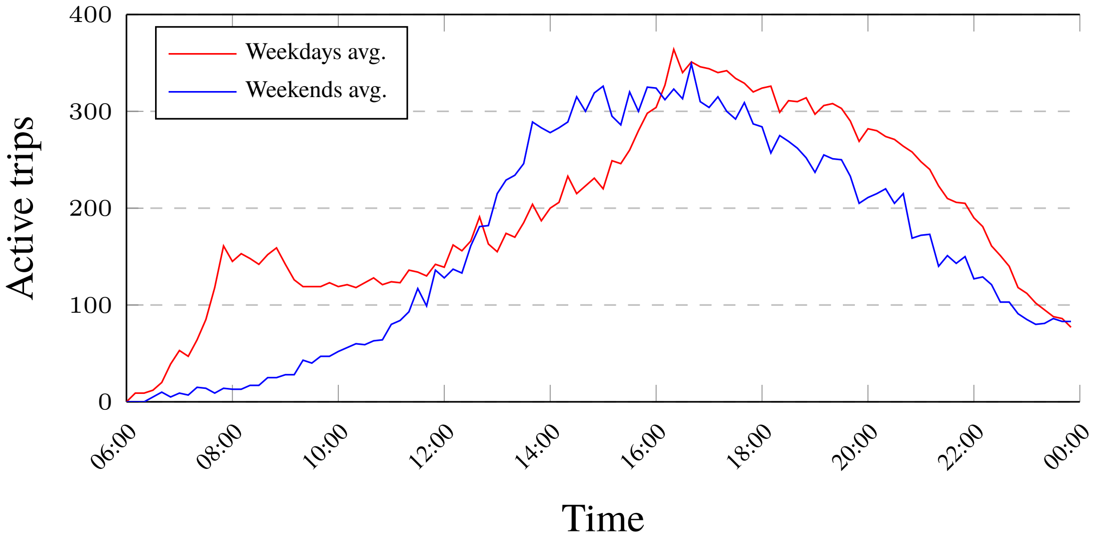


Figure 3.1: Average active trips for four weeks in the Oslo BSS

In the four weeks of May, the total number of bikes in the system varied between 700 and 900. This indicates that during peak hours, about half of the bikes in the system are used. In the afternoon on days with fair weather up to 75% of the bikes in the system are in use at the same time. During these hours there is not much work for the service vehicles, as there quite simply are few bikes to rebalance. During the morning peak, on the other hand, many short trips are carried out, the majority of which are going in the same direction. This is a hectic period for the system operator and a large part of the total violations occur during these hours according to the COO of Oslo Bysykkel (Felin, 2016).

Demand obviously depends on the weather. This is depicted in Figure 3.2 where the number of bikes in use are plotted for both a rainy and a sunny day. On Tuesday May 3rd there was light rain the entire day and an average temperature of 7.6 degrees Celsius, whereas one week later, Tuesday May 10th, the sky was clear and the average temperature was 16.1 degrees Celsius. 7,613 trips were completed May 3rd, while 11,915 were completed May 10th, implying 56% more trips due to improved weather. Knowing this, we see from Figure 3.2 that the number of trips are higher and the trip lengths are longer on days with good weather than on days with bad weather. These results suggest that the need for re-balancing heavily depends on the weather.

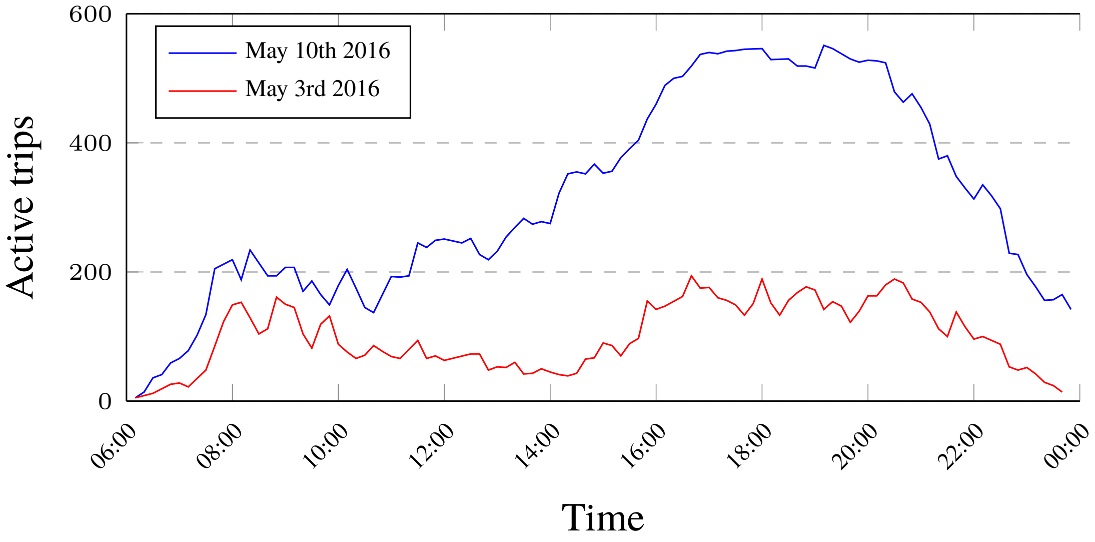


Figure 3.2: Active trips for two days with different weather in the Oslo BSS

3.1.2 Inventory Level

In the following, demand at two different stations in the Oslo BSS is depicted to illustrate some typical characteristics of demand. Figure 3.3 shows the inventory level in number of bikes at the Paléhaven station on May 27th 2016. The station is situated in walking distance from both the central station and the business area called Barcode. It is one of the largest stations in the system with 48 slots and is actively used, but during this recorded day the station gets neither full nor empty. The station is partly self-balanced, but deliveries to the station are generally higher than pickups. During the day, 184 bikes are delivered to the station and 120 bikes are picked up. To ensure that there are available slots, the service vehicles visit the station three times to remove in total 63 bikes. This is clearly visible in the figure, as the inventory level drops at about 10am, 4pm, and 10pm. There is a high demand for slots in the morning, as people use the city bikes to get to work in the business area Barcode, and the inventory level rises significantly. The inventory level at 12pm is about the same as in the morning, indicating that the station is well prepared for the demand the following day.

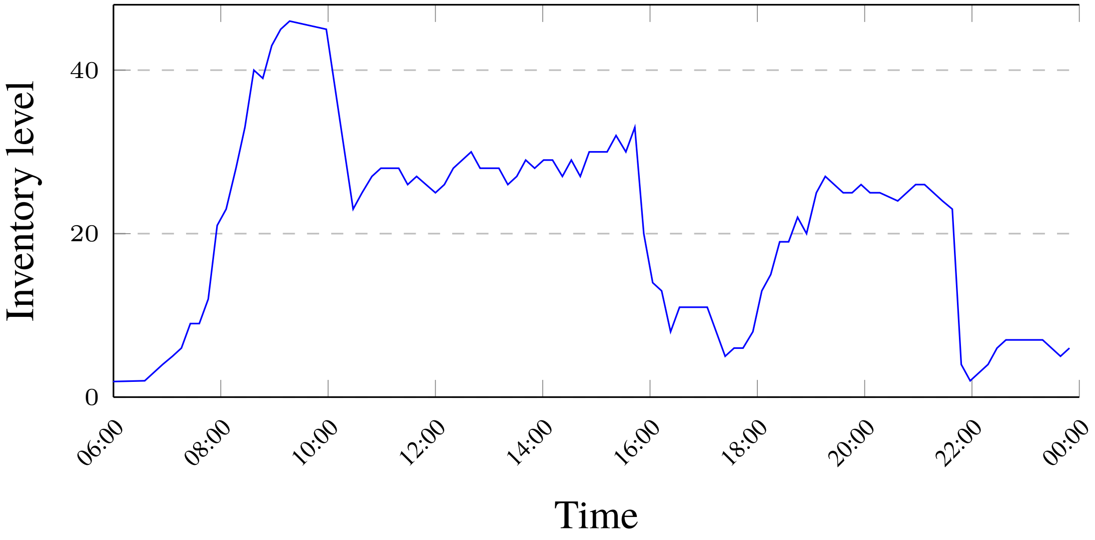


Figure 3.3: Inventory level at Paléhaven, May 27th 2016

In Figure 3.4, the inventory level on May 27th 2016 for the station at Bislett Stadium is shown. This station is situated in a residential area called Bislett, and is at a higher altitude than the city center. The station has a capacity of 24 bikes and is one of the busiest stations in the Oslo BSS. Demand for bikes is especially high in the morning, as can be seen from the figure. The service vehicles visit the station four times before 10am and still fail to fulfill the demand in this period. All the bikes delivered by the service vehicles are picked up by the users within few minutes. During the course of this day, the service vehicles delivered in total 119 bikes to the station through seven station visits. 287 bikes are delivered to the station by the users, and 417 bikes are picked up. Overall, the demand for bikes is significantly higher than the demand for slots and the station is in great need of deliveries by the service vehicles. Note that it may be difficult to estimate the total demand for bikes in the morning, as the station is often empty. At the end of the day the station is nearly empty, even though it is clearly beneficial that the station is full in the morning. This implies that the station is badly prepared for the demand the following day.

The demand profile for the two stations are clearly different. Both stations are quite busy, but the demands are of opposite kind. Demand at Bislett is higher, especially in the morning, and is harder to fulfill than at Paléhaven. Differences between the stations can be explained by their geographical location and the area they are situated in. Paléhaven is close to both the central station and the business area, while Bislett is situated in a residential area at a higher altitude than the city center.

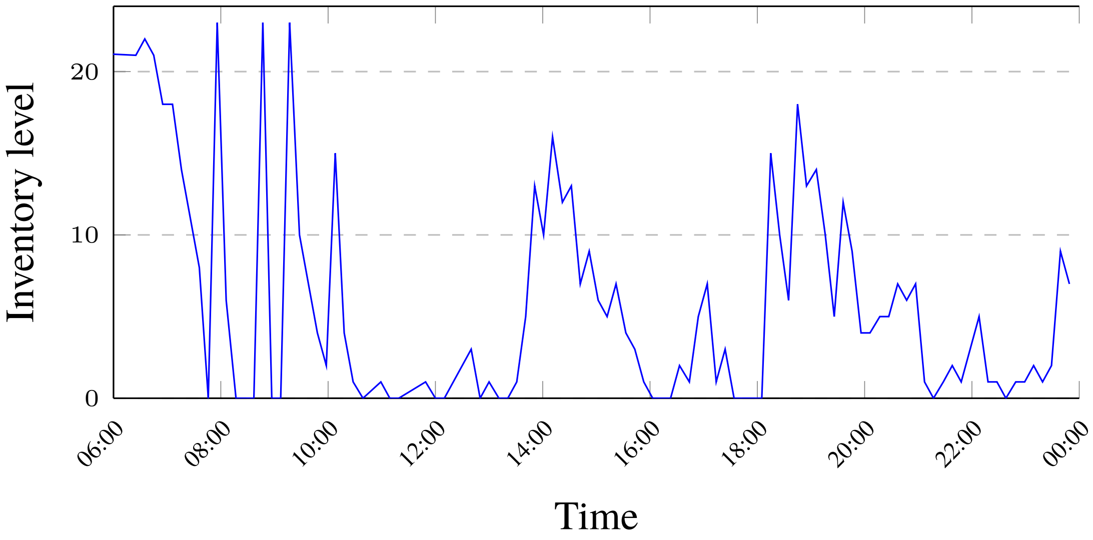


Figure 3.4: Inventory level at Bislett Stadium, May 27th 2016

3.2 Practical Application of Demand Data

Historical data may be used to estimate the expected demand, the variance, the number of violations, and the probability of violations during a time period. This information is valuable for the operator of the BSS in order to determine which stations to visit and the number of bikes to pick up or deliver. In the following, we assume net demand over a given time period (number of bikes picked up $-$ number of bikes delivered) at all stations to be normal distributed.

3.2.1 Expected Demand

The expected number of bikes that are picked up or delivered by the users at a station, i, over a given time period, hereby called expected demand for a time period, $E\left(D_i\overline{T}^D\right) = \mu_{D_i}$, is essential data for the operators. This corresponds to $D_i\overline{T}^D$ in the mathematical formulation in Section 5.2. D_i is the demand in bikes per minute at station i and $\overline{T}^D$ is the time limit for re-balancing. Note that the demand is negative when bikes are being picked up by the users. The notation used in this section is in accordance with the notation presented in Section 5.1.1 and 5.2.2.

The demand may be estimated using historical data for similar periods, assuming that they are representative for the given period. Important aspects to consider for such an approach are, as shown in Section 3.1, the geographical location of the station, the day of week, the time of day, and the weather, as the demand is greatly dependent on these parameters. To make an accurate estimate of the expected demand, a large number of representative historical observations are needed. If this data is available, many alternative forecasting methods could be used, such as last value, moving average, and exponential smoothing.

A challenge when estimating the expected demand is to handle the periods when stations are either full or empty. No one delivers bikes to a station that is full, or picks up bikes from a station that is empty. This does not mean that demand in these periods are zero, and methods for estimating the right demand for these periods must be applied. One alternative is to assume the same demand as the minutes before a station gets full or empty.

3.2.2 Variance

In addition to the expected demand for a period, an estimate for the variance, $\text{Var}\Big(D_i\overline{T}^D\Big) = \sigma^2_{D_i}$, is needed. The expected demand yields important information, however, this number is only really useful when considered with the variance. The variance indicates the spread of the data used to estimate the expected value, and a large variance indicate that the historical observations are dispersed. Calculating the variance in a data set is easily done using equation (3.1), where σ^2 is the variance, μ is the mean, X are all observations in the data set, and N is the number of observations.

$$\sigma^2 = \frac{\sum_{i=1}^{N}(X_i - \mu)^2}{N} \tag{3.1}$$

3.2.3 Number and Probability of Violations

Knowing the inventory levels and expected demand over a period for all stations, it is possible to calculate what we call the number of violations. This expression is used throughout the thesis and represents the expected number of violations if the demand is deterministic, i.e. without taking the variance into account. Calculating the number of violations is done by adding the expected demand for a period, μ_{D_i}, to the inventory level at the beginning of the planning period, S_i^0, and compare with the station capacity, Q_i^S. Equations (3.2-3.3) show the calculations for the number of violations. Note that a station may be expected to get violations either because it is full, v_i^F, or empty, v_i^E.

$$E(v_i^F) = max\Big\{0, S_i^0 - \mu_{D_i} - Q_i^S\Big\} \tag{3.2}$$

$$E(v_i^E) = max\Big\{0, \mu_{D_i} - S_i^0\Big\} \tag{3.3}$$

Another interesting measure is the probability of violations, which can be found based on the inventory level, expected demand, and the variance. By assuming that net demand over a given time period is normal distributed, the statistical formulas presented below, (3.4-3.8), enables to identify the probability of a violation occurring during a specific time period. The normal distribution is chosen because its shape seems reasonable and its characteristics are favourable. The probability of violations is found by adding up the probability of violations caused by the station being full and the station being empty. Even though a station is expected to get full, there is a small probability that the station becomes empty as long as $\sigma^2_{D_i} > 0$.

To compute the probability of violations, we use the cumulative distribution function (CDF), i.e. the probability that a random variable X will take a value less than or equal to a given value x. Assuming a mean, μ, and standard deviation, σ, different from 0 and 1, respectively, the CDF for the generic normal distribution is given by equation (3.4). Equation (3.5) gives the probability that the value of the normal random variable X will exceed x.

$$P(X \leq x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{x-\mu}{\sigma}} e^{\frac{-t^2}{2}} \mathrm{dt} \tag{3.4}$$

$$P(X > x) = 1 - P(X \leq x) \tag{3.5}$$

Here, X_i represents the actual net demand in number of bikes for the planning period at station i, and is defined to be positive when bikes are picked up by the users. As before, μ_{D_i} and $\sigma^2_{D_i}$ is the expected demand and the variance in demand, respectively. Using equations (3.4-3.5), we get equation (3.6), which is the probability of violations caused by the station being empty. This happens when the actual net demand for the period, X_i, exceeds the initial inventory level, S^0_i.

$$P\big(v^E_i > 0\big) = P\big(X_i > S^0_i\big) = 1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{S^0_i - \mu_{D_i}}{\sigma_{D_i}}} e^{\frac{-t^2}{2}} \mathrm{dt} \tag{3.6}$$

Correspondingly, equation (3.7) expresses the probability of violations caused by a full station, i.e. the probability that more bikes are delivered through the period than there are free slots at the station. A negative value for X_i implies that there is a net delivery of bikes by the users to the station.

$$P\big(v^F_i > 0\big) = P\big(X_i \leq S^0_i - Q^S_i\big) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{S^0_i - Q^S_i - \mu_{D_i}}{\sigma_{D_i}}} e^{\frac{-t^2}{2}} \mathrm{dt} \tag{3.7}$$

By combining equations (3.6) and (3.7), we get equation (3.8), which gives the total probability of violations at a station.

$$P\big(v_i^F + v_i^E > 0\big) = 1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{S_i^0 - \mu_{D_i}}{\sigma_{D_i}}} e^{\frac{-t^2}{2}} \mathrm{dt} + \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{S_i^0 - Q_i^S - \mu_{D_i}}{\sigma_{D_i}}} e^{\frac{-t^2}{2}} \mathrm{dt} \tag{3.8}$$

Both the number of violations and the probability of violations may be used by the system operator when re-balancing the system. They may for instance be used to sort the stations by urgency or ignore the stations with the lowest values. An advantage with the number of violations is that it is easy to calculate or estimate. The probability of violations, on the other hand, is more complex, but captures more realism by considering the variance. Decision rules and prioritization of which stations to visit first can be made based on this information and are elaborated on in Chapter 8. One simple decision rule is for instance to always visit the station with the largest probability of violations.

3.2.4 Optimal State

The demand forecast may also be used to compute the optimal state, which is an important input for both the SBRP-model and the DBRP-model. We suggest an approach based on the probability of violations (equations (3.6-3.7)). We define the optimal state at a station to be the point where the probability of violations because the station is full equals the probability of violations because the station is empty, i.e. equation (3.9) is fulfilled. This method is quite similar to Parikh and Ukkusuri (2015) and utilizes both the expected demand and the variance to set the optimal state to a level where the system is well prepared for future demand.

$$1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{S_i^0 - \mu_{D_i}}{\sigma_{D_i}}} e^{\frac{-t^2}{2}} \mathrm{dt} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{S_i^0 - C_i^S - \mu_{D_i}}{\sigma_{D_i}}} e^{\frac{-t^2}{2}} \mathrm{dt} \tag{3.9}$$

Using this approach, the optimal state is often close to full when there is a positive demand for bikes, and close to empty when there is a negative demand. However, because of the variance (uncertainty), the optimal state will rarely indicate entirely full or entirely empty stations. A higher variance implies that the optimal state should be the further away from the capacity limits.

CHAPTER 4

PROBLEM DESCRIPTION

Based on the literature survey of the operational level in Section 2.5, we present our understanding of the SBRP and DBRP seeking to consolidate the literature. Different aspects of the problems are elaborated, and the main assumptions, restrictions, and decisions are highlighted. The two problems are mostly presented together, as the main structure is similar. Differences between the problems are pointed out when necessary.

4.1 The Models' Objectives

The objective of the SBRP is to minimize the total deviation at the end of the planning period. It may be more important to reach the optimal state on some stations than others. This could be handled by giving the stations different weight based on their importance. However, such weighting is not done here. In addition to minimizing the total deviation, it is desirable to re-balance the system using as little time as possible when considering the SBRP. The time usage should be considered in the objective function, but this part should be given a much lower weight than the deviations. When comparing deviations with time usage in the objective function, it is important to be aware of the fact that the terms have different scales.

The objective of the DBRP is to minimize the total deviation at the end of the planning period, as well as the sum of violations occurring during the planning period. The sum of violations should be given more weighting than the total deviation, indicating that one violation is worse than one deviation. Deviations are included in the objective function to do the re-balancing so that the system is well prepared for the demand in later periods. If only violations are minimized, the system may perform well in the first period, but not necessarily in the longer run. Note that time usage could also be considered in the objective function of the DBRP.

4.2 Routing

Considering the fact that the SBRP is static and deterministic, optimal routes for the entire planning horizon could be determined at the beginning of the period. Hence, the complete routes determined at the beginning of the overnight re-balancing period do not need to be updated during operation. During the day, on the other hand, the problem is dynamic and stochastic, and new information is received during operation. The optimal routes could then be updated using either online or preprocessed decisions, as introduced in Section 2.5.1.

A given number of service vehicles are used for both problems. For the SBRP, all vehicles start and finish empty at the depot. The vehicles in the DBRP may however start and finish anywhere in the system, and with a varying number of bikes. The starting point and initial number of bikes in each vehicle are inputs to the model. The vehicles may either start at a station, or be on its way to a station, with a given driving time to arrival.

For every vehicle, a complete route has to be decided, which is done by determining the order of station visits. For every station visit, the number of bikes to pick up or deliver must be determined. These numbers have to be within the already mentioned capacity and loading limits. Several vehicles can visit the same station, and a single vehicle can visit the same station several times.

A station with too many bikes in the initial state compared to in the optimal state is defined as a *pickup station* for the SBRP. Correspondingly, stations with too few bikes are defined as *delivery stations*. For the DBRP, all stations are defined as either pickup or delivery stations by comparing the optimal state with the expected final state, given by the initial state and the demand. It is only allowed to pick up bikes at pickup stations and deliver bikes at delivery stations. This is a simplification of the problem, excluding solutions where the stations are used as temporary depositories. Since this opportunity is considered to be of negligible value, it is omitted to get a simpler problem formulation.

4.3 Time Usage

The service vehicles use a given amount of time to drive between the different stations. We presume that the time usage is constant and independent of the hour. This is not necessarily a bad assumption for the SBPR, as it is used for overnight operation, when traffic is limited. For the DBRP, however, this assumption is obviously a simplification. In addition to driving time, each vehicle uses a fixed parking time at each station visit. Time used to load and unload bikes at a station, the so-called handling time, is proportional to the number of bikes handled at the station.

The static problem is analog with overnight repositioning. However, the operator has restricted time for static repositioning, a static time limit. Consequently, the time available might be insufficient to make the total deviation equal to zero, and thus the time limit for operation is an essential constraint.

In the dynamic version of the problem, the planning period is restricted by a dynamic time limit. When operating in a dynamic setting, it is only worth considering a limited planning horizon, depending on the distribution and predictability of demand.

4.4 Capacities

Since there is a limited number of slots at each station, all stations have restricted capacity. Consequently, the maximum number of bikes that can be brought to a station equals the total number of slots minus the current amount of bikes at the station. Note that there may be different number of slots at different stations.

Furthermore, each vehicle has a restricted carrying capacity. The vehicles may be identical or have different capacity. If the capacities differ, the fleet is said to be heterogeneous.

For the DBRP there are loading limits at each station that are more restrictive than the station capacities. The difference between the loading limits and the station's capacities is called safety margin. This is typically quite small and could also be set to zero. Safety margins are introduced to avoid violations right after service at a station because of unpredictable demand. Such safety margins are not necessary for the SBRP, because of the deterministic setting and the fact that expected violations are not part of the objective function.

4.5 Demand

Demand is not considered in the SBRP, only in the DBRP, according to the definition of the problems. For the DBRP, there is a demand for bikes at all stations. This demand could be modeled in various ways. One method, considering the stochasticity of demand, is to utilize probability distributions for the demand at all station, e.g. assuming net demand over a given time period to be normal distributed. This results in a complex, non-linear formulation. Alternatively, the problem could be simplified by assuming that there is a constant demand at each station, which is known for all time periods of the day.

4.6 Example Problem

Figures 4.1 and 4.2 illustrate an example problem for the SBRP. The first figure shows the characteristics of the problem, and the second shows a feasible solution. Twelve stations are to be re-balanced by two service vehicles between 23:00 and midnight. In the initial state, some stations have too many bikes (pickup stations) and some stations too few (delivery stations). In the parenthesis (x,y) above the stations, x is the initial number and y the optimal number of bikes at the station. Note that station 2 is initially in balance, and can be removed from the problem a priori. The two main limiting restrictions in this example are time, and vehicle capacities. There is only one hour available, and the two service vehicles can carry at maximum 15 bikes each. Because of the limited capacity, station 3 ends up with one bike extra and station 4 ends up one bike short. As a result of the limited time available, visiting station 9 is not prioritized. Hence, station 9 ends up with 14 instead of 15 bikes, which consequently leads to station 8 ending up with one additional bike. In the final state, after the re-balancing, the system is close but not equal to the optimal state.

An important difference between the SBRP (as illustrated) and the DBRP, is the fact that in the DBRP the vehicles may start and finish non-empty at whichever station. Further, the continuous demand in a dynamic setting is not taken into account in this example. Nevertheless, the main structure in routing and pickup and delivery decisions look quite similar for both problems.

4.7 Conceptual Overview

The SBRP is characterized by a static system where there is no demand during the planning time. For the DBRP, demand is taken into account, resulting in a dynamic problem.

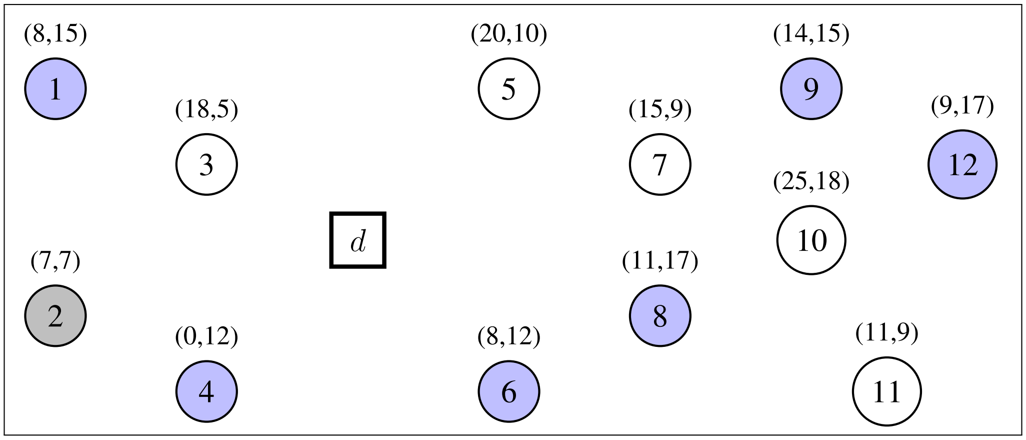


Figure 4.1: Illustration of the SBRP. There are 12 stations and one depot. The first number in the parentheses is the initial amount of bikes at the station, and the second number is the optimal amount. Pickup stations are white, delivery stations are blue and initially re-balanced stations are grey.

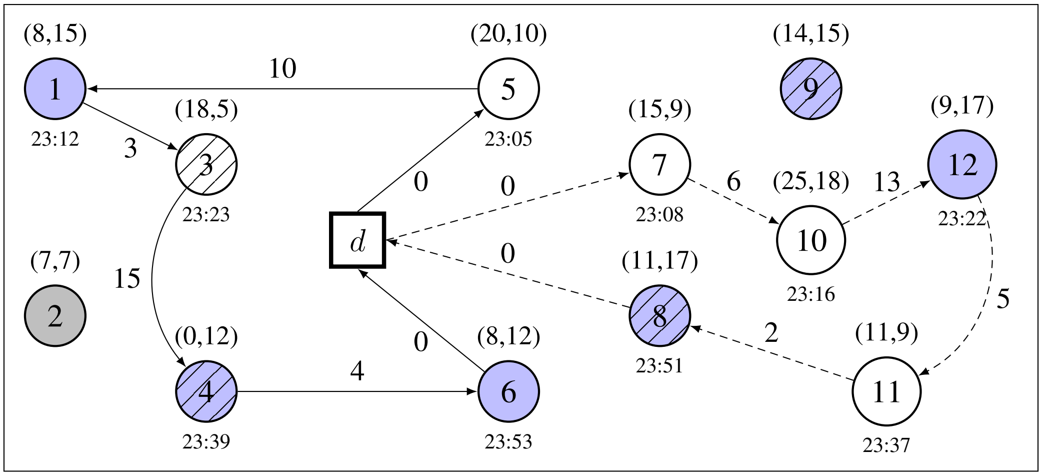
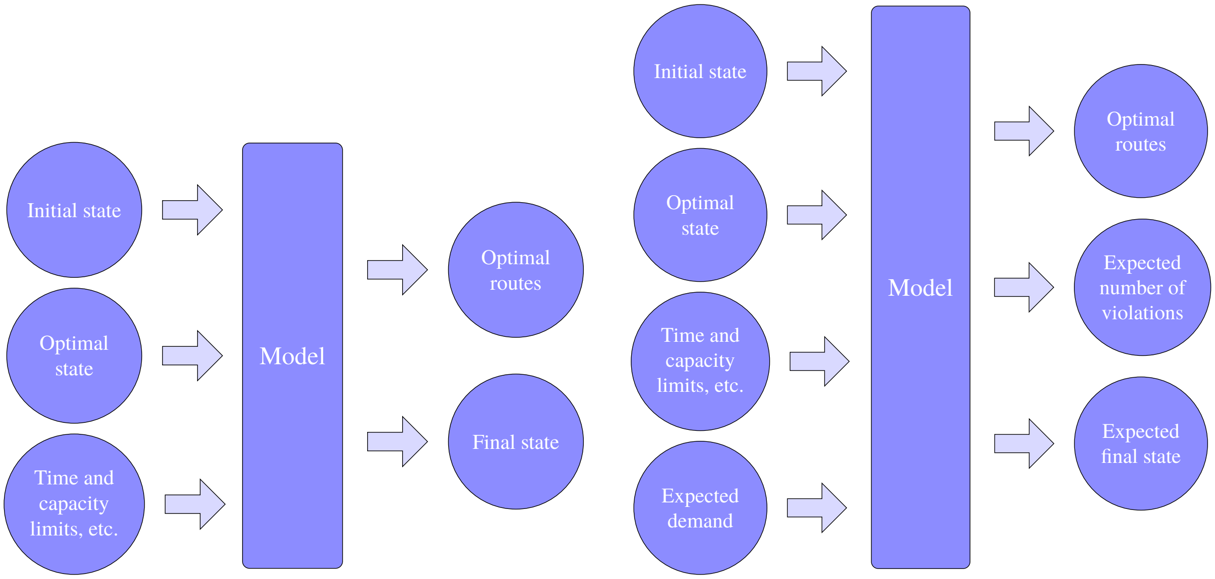


Figure 4.2: A feasible solution for two service vehicles, plotted together with the number of bikes moved between the stations. A station is filled with lines if it is not perfectly re-balanced.

The objective of the SBRP is to find effective pickup-and-delivery routes for the service vehicles that minimize the deviation between the final and the optimal state. For the DBRP, the objective is to minimize both the total expected deviation at the end of the planning period, and the sum of expected violations during the period. Limited time and capacity at both the stations and the vehicles constrain the problems, and for the DBRP the expected demand is an important input. A conceptual overview of the models are shown in Figures 4.3a and 4.3b.



(a) The SBRP

(b) The DBRP

Figure 4.3: Conceptual overview of the problems

CHAPTER 5

MODEL FORMULATION

In Chapter 4, the SBRP and the DBRP are presented. In this chapter, we propose new mathematical formulations to solve the two problems, starting with the SBRP. Many models in the literature lack realism, as pointed out in Section 2.5. Therefore, the models presented here are more detailed and capture more real-life aspects than the majority of existing models. We call our mathematical formulations *the SBRP-model* and *the DBRP-model*, as they are models for solving the SBRP and the DBRP, respectively. The two models bear many similarities and differences between them are highlighted through the text.

5.1 Model for the Static Bicycle Repositioning Problem

In this section, the notation and constraints for the SBRP-model are introduced, and some alternative formulations discussed. To reduce the computational time used to solve the model, we suggest several symmetry breaking constraints and valid inequalities.

5.1.1 Notation

All notation used in the SBRP-model is listed in Table 5.1. The problem is defined on a set of stations, $\mathcal{N}$, indexed by i and j and k. Note that the depot is included in this set. $\mathcal{V}$, the set of vehicles, is indexed by v. The time a vehicle uses between station i and j is defined as T_{ij}. T^P is the time used for parking at a station and T^H is the handling time used for loading or unloading a bike. The time limit for re-balancing a static system is $\overline{T}^S$. We call this the static time limit. The capacity of vehicle v is denoted Q^V_v. Capacities for the stations are not needed in the SBRP-model, because it is only allowed to pick up bikes at pickup stations and deliver bikes at delivery stations, and because the penalties in the objective function are equally weighted for both shortage and excess.

S^0_i, O_i, and s^E_i describe the initial, optimal, and final state for station i, respectively, of which the first two are inputs, and the third is a variable to be determined in the model. S^0_i is the initial number of bikes at station i at the beginning of the re-balancing period. O_i, is the desired number of bikes at station i at the end of the period. The optimal state could for instance be calculated using the approach presented in Section 3.2.4. The final state, represented by variable s^E_i, is the number of bikes at station i after the re-balancing. Parameter α indicates the relative importance of deviations to time usage, where $\alpha = 1$ implies only considering deviations.

Every station i has a set of possible visits $\mathcal{M}_i$. Hence the maximum number of times station i can be visited is limited to $|\mathcal{M}_i|$. The formulation uses the arc flow variables x_{imjnv}, similar to Christiansen (1999). This variable equals 1 if vehicle v drives directly from station visit (i, m) to station visit (j, n). Here, m and n are the visit numbers. The first time a station is visited, $m = 1$, the second time, $m = 2$, etc. This kind of arc flow variables makes multiple visits possible and eliminates the need for big-M formulations in the loading and capacity constraints. Figure 5.1 demonstrates the x_{imjnv}-variables in a small example problem.

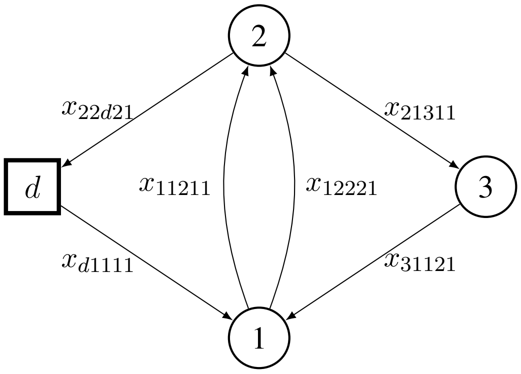


Figure 5.1: Illustration of the arc-flow variables x_{imjnv}, with one vehicle $v = 1$. The figure shows which variables x_{imjnv} that take the value 1. Vehicle $v = 1$ visit the stations in the following sequence: $d \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow d$.

Table 5.1: Notation used in the mathematical formulation for the SBRP

Sets	
$\mathcal{N}$	Set of stations
$\mathcal{V}$	Set of vehicles
$\mathcal{M}_i$	Set of possible visits at station i
Indices	
i, j, k	Station $i, j, k \in \mathcal{N}$
v	Vehicle $v \in \mathcal{V}$
m, n	Visit number $m, n \in \mathcal{M}_i$ at station i
Parameters	
T_{ij}^{D}	Driving time between stations i and j
T^{P}	Time used for parking a vehicle
T^{H}	Handling time used for loading or unloading a bike
$\overline{T}^{S}$	Static time limit for the operation of service vehicles
Q_v^{V}	Capacity of vehicle v
J_i	1 if station i is a pickup station, –1 if it is a delivery station, and 0 if it is initially in balance
α	Weight on deviations in the objective function relative to time usage
$\overline{A}$	Maximum number of station visits for a vehicle
S_i^{0}	Initial state, number of bikes at station i
O_i	Optimal state, number of bikes at station i
Variables	
x_{imjnv}	1 if vehicle v is driving directly from station visit (i, m) to station visit (j, n), 0 otherwise
f_{ijv}	Total number of bikes carried by vehicle v between stations i and j
q_{iv}	Number of bikes either picked up or delivered at station i by vehicle v
s_i^{E}	Final state, number of bikes at station i at the end of the planning period
u_{imv}	The sequence number in which station visit (i, m) is made by vehicle v
w_{imjnv}	1 if vehicle v makes station visit (i, m) before station visit (j, n), 0 otherwise

The variables f_{ijv} are defined as the total number of bikes carried by v on the arc (i, j). Since f_{ijv} are not indexed by visit number, they represent the sum of all bikes transported by vehicle v on the arc. The variables q_{iv} keep track of the number of bikes loaded or unloaded at station i by vehicle v. q_{iv} is always a positive number. The parameter J_i is equal to 1 if station i is a pickup station, and –1 if it is a delivery station. If the station is initially in balance, $J_i = 0$, implying that the station do not need to be visited, and can be removed from the problem.

The variables u_{imv} and w_{imjnv}, and the parameter $\overline{A}$ are used in the subtour eliminating constrains and are presented later.

5.1.2 Mathematical Formulation

$$\begin{aligned} \min \quad & \alpha \sum_{i \in \mathcal{N}} J_i(s_i^E - O_i) \\ & +(1-\alpha) \left[\sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}_i} \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \sum_{v \in \mathcal{V}} \left(T_{ij}^D + T^P\right) x_{imjnv} + \sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{V}} T^H q_{iv} \right] \end{aligned} \tag{5.1}$$

subject to:

$$\sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} x_{dvjnv} = 1 \qquad v \in \mathcal{V} \tag{5.2}$$

$$\sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}_i} x_{imd(v+|\mathcal{V}|)v} = 1 \qquad v \in \mathcal{V} \tag{5.3}$$

$$\sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} x_{jnimv} - \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} x_{imjnv} = 0 \qquad i \in \mathcal{N} \setminus \{d\}, m \in \mathcal{M}_i, v \in \mathcal{V} \tag{5.4}$$

$$\sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \sum_{v \in \mathcal{V}} x_{imjnv} \leq 1 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i \tag{5.5}$$

$$\sum_{j \in \mathcal{N}} f_{jiv} + J_i q_{iv} - \sum_{j \in \mathcal{N}} f_{ijv} = 0 \qquad i \in \mathcal{N}, v \in \mathcal{V} \tag{5.6}$$

$$s_i^E + \sum_{v \in \mathcal{V}} J_i q_{iv} = S_i^0 \qquad i \in \mathcal{N} \tag{5.7}$$

$$\sum_{v \in \mathcal{V}} q_{iv} - J_i(S_i^0 - O_i) \leq 0 \qquad i \in \mathcal{N} \tag{5.8}$$

$$f_{ijv} - \sum_{m \in \mathcal{M}_i} \sum_{n \in \mathcal{M}_j} Q_v^V x_{imjnv} \leq 0 \qquad i, j \in \mathcal{N}, v \in \mathcal{V} \tag{5.9}$$

$$\sum_{j \in \mathcal{N}} f_{djv} = 0 \qquad v \in \mathcal{V} \tag{5.10}$$

$$\sum_{i \in \mathcal{N}} f_{idv} = 0 \qquad v \in \mathcal{V} \tag{5.11}$$

$$\sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}_i} \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \left(T_{ij}^D + T^P\right) x_{imjnv} + \sum_{i \in \mathcal{N}} T^H q_{iv} \leq \overline{T}^S \qquad v \in \mathcal{V} \tag{5.12}$$

$$\text{Subtour eliminating constraints} \tag{5.13}$$

$$x_{imjnv} \in \{0,1\} \qquad i,j \in \mathcal{N}, m \in \mathcal{M}_i, n \in \mathcal{M}_j, v \in \mathcal{V} \tag{5.14}$$

$$f_{ijv} \geq 0, \text{integer} \qquad i,j \in \mathcal{N}, v \in \mathcal{V} \tag{5.15}$$

$$q_{iv} \geq 0, \text{integer} \qquad i \in \mathcal{N}, v \in \mathcal{V} \tag{5.16}$$

$$s_i^E \geq 0, \text{integer} \qquad i \in \mathcal{N} \tag{5.17}$$

The objective function (5.1) consists of two terms that are to be minimized. The first term is the deviation in number of bikes between the final state, s_i^E, and the optimal state, O_i, for all stations. Having too many and too few bikes are equally penalized. The second term is the accumulated time used by the service vehicles to obtain the final state. Total time corresponds to the sum of driving time, T_{ij}^D, parking time, T^P, and handling time, T^H. By setting α slightly below one, the most effective routes minimizing the total deviation are found.

Constraints (5.2) and (5.3) force the vehicles to start and end at the depot, d. Symmetry in the depot is handled by stating that vehicle v uses visit numbers v and $v + |\mathcal{V}|$ when leaving and arriving at the depot, respectively. Constraints (5.4) ensure that a vehicle that enters a station visit, leaves the same station visit, while constraints (5.5) make sure all station visits happen at most once.

The loading and unloading constraints (5.6) ensure that the flow of bikes into a station, f_{jiv}, equals the flow out of the station, f_{ijv}, plus the net pickup, q_{iv}. Since the problem is static, only the total net pickup is considered. Constraints (5.7) and (5.8) assign values to the final state, s_i^E. In addition, constraints (5.8) give an upper bound on the net pickup at station i by vehicle v, q_{iv}.

The vehicle capacity constraints (5.9) make sure that a vehicle never carries more bikes along an arc than the vehicle's capacity multiplied by the number of times the arc is traversed. Constraints (5.10) and (5.11) state that the service vehicles must be empty when leaving and returning to the depot. Capacity constraints for the stations are handled implicitly. The total time spent for each vehicle is limited to the static time limit, $\overline{T}^S$, by constraints (5.12). Constraint (5.13) represent subtour elimination constraints. Several possible formulations are presented in Section 5.1.3.

Non-negativity and integer constraints are given by (5.14-5.17). The variables f_{ijv}, q_{iv}, s_i^E all have to take integer values, but if either f_{ijv} or q_{iv} is defined as integer, constraints (5.6-5.7) force the remaining two to be integer as well.

5.1.3 Subtour Elimination

Various methods exist to handle subtours. As shown in Section 2.5.2, the formulation developed by Miller et al. (1960); the so called Miller-Tucker-Zemlin (MTZ) constraints are popular in the literature. Subtour elimination could also be implemented using a strictly increasing time variable, but this formulation would be weaker. Alternatively, subtour elimination could be achieved by generating the constraints as they are needed. This would however imply making a program that detects subtours automatically and generate the necessary cuts through an iterative process.

In the MTZ-constraints (5.18), u_{imv} is the sequence number in which station visit (i, m) is made by vehicle v. Hence, u_{imv} can take values $1, 2, ..., \overline{A}$, where $\overline{A}$ is the maximum number of possible station visits made by vehicle v. Note that u_{imv} is integer. In our problem all tours must begin and end at the depot, and it is not allowed to go directly from the depot to the depot. Hence, constraints (5.18) can be strengthened to constraints (5.19).

$$u_{imv} - u_{jnv} + \overline{A}x_{imjnv} \leq \overline{A} - 1 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, v \in \mathcal{V} \tag{5.18}$$

$$u_{imv} - u_{jnv} + (\overline{A} - 1)x_{imjnv} \leq \overline{A} - 2 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, v \in \mathcal{V} \tag{5.19}$$

The MTZ constraints were originally developed for eliminating subtours in the travelling salesman problem (TSP). In the TSP, each node must be visited exactly once. Hence, the big-M $(\overline{A})$ could equal the number of nodes, $|N|$, and each node would be sequenced from 1 to $|N|$. In our problem, which allows multiple visits to some stations, the total number of visits to be made is unknown. Accordingly, the big-M would often be given a value much larger than the actual number of visits needed in the solution. As a result, the linear relaxation of the problem becomes weak. For the problem at hand, $\overline{A}$ may be set to for example $\overline{A} = \sum_{i \in \mathcal{N}} |\mathcal{M}_i|$, i.e. the sum of the maximum number of possible visits over all stations. This formulation may be strengthened, for instance when $\overline{T}^S$ is relatively small. By making use of the fact that all vehicles must depart from the depot, hence using two visit numbers in the depot and minimum one visit to another station, we can set $\overline{A} = \sum_{i \in \mathcal{N}} |\mathcal{M}_i| - 3\,(|\mathcal{V}| - 1)$.

Various methods exist to strengthen the MTZ-formulation. In a method proposed by Desrochers and Laporte (1991), an extra term is introduced at the left hand side, resulting in a tighter bound for the u_{imv}-variables. The MTZ-formulation with this extension is shown in constraints (5.20).

$$u_{imv} - u_{jnv} + (\overline{A} - 1)x_{imjnv} + (\overline{A} - 3)x_{jnimv} \leq \overline{A} - 2 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, v \in \mathcal{V} \quad (5.20)$$

Given $x_{imjnv} = 1$ we get two constraints for the node pair $[(i, m), (j, n)]$, shown in constraints (5.21-5.22). The result of the new, tighter formulation of the MTZ-constraints is that $u_{jnv} = u_{imv} + 1$ (instead of the original result: $u_{jnv} \geq u_{imv} + 1$).

$$u_{imv} - u_{jnv} \leq -1 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, v \in \mathcal{V} \quad (5.21)$$

$$u_{jnv} - u_{imv} \leq 1 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, v \in \mathcal{V} \quad (5.22)$$

An alternative formulation for eliminating subtours is proposed by Chemla et al. (2013), and is shown in constraints (5.23-5.25). This formulation introduces many binary variables, hoping to achieve a stronger formulation than the MTZ-constraints. The variable w_{imjnv} is binary and takes value 1 if vehicle v makes station visit (i, m) before station visit (j, n), 0 otherwise.

$$x_{kpjnv} + w_{imkpv} - w_{imjnv} \leq 1 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, k \in \mathcal{N}, p \in \mathcal{M}_k, v \in \mathcal{V} \quad (5.23)$$

$$x_{imjnv} - w_{imjnv} \leq 0 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, v \in \mathcal{V} \quad (5.24)$$

$$w_{iminv} = 0 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, n \in \mathcal{M}_i \setminus \{n < m\}, v \in \mathcal{V} \quad (5.25)$$

To assign the right values to the variables w_{imjnv}, constraints (5.23 - 5.24) are introduced. Then, to ensure that no subtours occur, constraints (5.25) state that if vehicle v returns to station i after having made stations visit (i,m), it must make station visit (i, n), where $n > m$. Since there is an upper bound on the number of station visits, this formulation makes sure no subtours occur. The only station vehicles can visit without having to leave, is the depot. Hence all routes must end in the depot.

5.1.4 Symmetry Breaking Constraints

Symmetry means that a problem has many solutions that are mathematically different, but practically identical. Much computational effort may be needed when solving problems with symmetry, as the computer has to evaluate all the symmetrical solutions.

The station visit numbers are one cause of symmetry in the problem at hand. Solutions that in practice are identical, differ only with regard to the visit numbers. For instance would visits to visit number 2 and 3 imply two mathematically different solutions, even though they in practice are identical. Constraints (5.26) remove this symmetry by forcing the lowest station visit numbers to be used first.

$$\sum_{j\in\mathcal{N}}\sum_{n\in\mathcal{M}_j}\sum_{v\in\mathcal{V}}\left(x_{imjnv}-x_{i(m-1)jnv}\right)\leq 0 \qquad i\in\mathcal{N}\setminus\{d\}, m\in\mathcal{M}_i\setminus\{1\} \tag{5.26}$$

Another cause of symmetry is the service vehicles. Using a homogeneous fleet, it is irrelevant which vehicle drives which route. This symmetry can be eliminated by introducing one of the following two sets of constraints. Constraints (5.27) state that vehicle one, $v = 1$, must visit more stations than vehicle two, $v = 2$, and so forth. The second set of constraints (5.28), forces vehicle one to use longer time than vehicle two, and so forth. To assure that the optimal solution is not cut away, only one of these sets of constraints can be imposed. On the other hand, both of these sets of constraints can be combined with (5.26), since they are dealing with different symmetry problems. Note that constraints (5.27-5.28) only apply when $Q_v^V = Q_{(v+1)}^V$.

$$\sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}_i}\sum_{j\in\mathcal{N}}\sum_{n\in\mathcal{M}_j}\left(T_{ij}^D+T^P\right)\left(x_{imjnv}-x_{imjn(v+1)}\right)+\sum_{i\in\mathcal{N}}T^H\left(q_{iv}-q_{i(v+1)v}\right)\geq 0$$
$$v\in\mathcal{V}\setminus\{|\mathcal{V}|\}\;\Big|\;Q_v^V=Q_{(v+1)}^V \tag{5.27}$$

$$\sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}_i}\sum_{j\in\mathcal{N}}\sum_{n\in\mathcal{M}_j}\left(x_{imjnv}-x_{imjn(v+1)}\right)\geq 0 \qquad v\in\mathcal{V}\setminus\{|\mathcal{V}|\}\;\Big|\;Q_v^V=Q_{(v+1)}^V \tag{5.28}$$

5.1.5 Valid Inequalities

Adding valid inequalities is a way of improving the solution of the linear relaxation. Solving the linear relaxation of a problem implies solving a linear program (LP). Using such inequalities may result in a tighter formulation, without removing the optimal solution. Ideally, one wants to find those forming the convex hull, which makes the solution of the linear relaxation integer. A valid inequality is called a cut if it makes the currently best LP-solution infeasible. Such cuts make the LP-solution a better optimistic bound of the optimal IP-solution, and could reduce the computational time in the B-&-B tree. It is difficult to know a priori whether a valid inequality is a cut.

Constraints (5.29) are valid inequalities that force some of the x-variables to take on values closer to one in the linear relaxation of the problem. Presume that $|(S_i^0 - O_i)| = 5$ and that the capacity of the vehicle is $Q_v^V = 15$, then the solution to the LP could give $x_{imjnv} = \frac{1}{3}$ and still allow $q_{iv} = 5$. Enforcing constraints (5.29), would give $x_{imjnv} = 1$ in the linear relaxation if $q_{iv} = 5$ and only one vehicle was visiting the station.

$$\sum_{v \in \mathcal{V}} q_{iv} - |\left(S_i^0 - O_i\right)| \sum_{m \in \mathcal{M}_i} \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \sum_{v \in \mathcal{V}} x_{imjnv} \leq 0 \qquad i \in \mathcal{N} \tag{5.29}$$

The formulation can be tightened using valid inequalities for the 1-PDTSP from Hernández-Pérez and Salazar-González (2007). Differences between the 1-PDTSP and the SBRP are elaborated on in Section 2.5.2. The proposed inequalities are only valid for the case where there is enough time available, so that it is guaranteed to find at least one solution with zero total deviation. Further, stations i, j, and k must be the same type (pickup or delivery) and $|\mathcal{M}_i| = |\mathcal{M}_j| = |\mathcal{M}_k| = 1$. Constraints (5.30) allow maximum one traverse of all the the arcs between i, j and k, if the total quantity to be picked up or delivered at the three stations exceeds the capacity of the service vehicle.

$$\sum_{v \in \mathcal{V}} \left(x_{imjnv} + x_{imkpv} + x_{jnkpv} + x_{jnimv} + x_{kpimv} + x_{kpjnv}\right) \leq 1$$

$$i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, k \in \mathcal{N}, p \in \mathcal{M}_k$$

$$\Big| \; J_i = J_j = J_k, \;\; |\mathcal{M}_i| = |\mathcal{M}_j| = |\mathcal{M}_k| = 1, \;\; |(S_i^0 - O_i) + (S_j^0 - O_j) + (S_k^0 - O_k)| > \max_{v \in \mathcal{V}} \left\{Q_v^V\right\} \tag{5.30}$$

Another set of valid inequalities is derived by Chemla et al. (2013). They prove that the arcs between two stations with both positive or both negative imbalance values need not be traversed more than once. Using this information, constraints (5.31) may be added to the problem.

$$\sum_{v \in \mathcal{V}} \sum_{m \in \mathcal{M}_i} \sum_{n \in \mathcal{M}_j} x_{imjnv} + \sum_{v \in \mathcal{V}} \sum_{m \in \mathcal{M}_i} \sum_{n \in \mathcal{M}_j} x_{jnimv} \leq 1 \quad i \in \mathcal{N}, j \in \mathcal{N} \;\Big|\; J_i = J_j \tag{5.31}$$

5.2 Model to Solve the Dynamic Bicycle Repositioning Problem

In this section, a mathematical model for solving the DBRP is presented. First, we discuss how dynamism is handled, and then we present the notation and constraints. Finally, different symmetry breaking constraints and valid inequalities are suggested.

5.2.1 A Dynamic Problem

The DBRP is a dynamic problem, but in order to solve it we choose to simplify the dynamism in our DBRP-model. The model presented here makes use of a continuous demand and considers the timing of events, but must be re-solved with updated values in order to consider new information.

We model the DBRP by considering a planning horizon of some length, e.g. one hour, but assume that the problem is regularly re-solved with updated values, for instance every time a service vehicle arrives at a station. To simplify, demand is represented using a constant value in bikes per minute for each station. This value could for instance be an expected value derived from historical data, see Section 3.2.1. Using relatively short time periods, constant demand seems reasonable. Given that the problem is frequently re-solved with updated values, we assume that an expected value approach depict the reality sufficiently accurate.

5.2.2 Notation

This section introduces notation that is not already described for the SBRP-model. This notation is found in Table 5.2. The model takes a starting station for each vehicle, v, as input, denoted by $o(v)$. The vehicles can start anywhere in the system, but are always initially on its way to the starting station, with $T_v^{o(v)}$ minutes left to drive. If a vehicle starts exactly at a station, $T_v^{o(v)} = 0$. Each vehicle ends the route at a destination point denoted by $d(v)$. This destination point is an artificial station with driving time equal to zero from all stations. The time limit, $\overline{T}^D$, in the DBRP-model is called the dynamic time limit and is similar to the static time limit, $\overline{T}^S$, in the SBRP-model. Note however that $\overline{T}^D$ is not the total time for operation that day, but a chosen planning horizon.

Q_i^S is the capacity of station i. This parameter was not needed in the SBRP-model, but must be included here to keep inventory levels within the station's capacities. The weighting of the deviations relative to the violations in the objective function is represented by β. D_i represents the net demand in bikes per minute at station i. This parameter is positive when bikes are picked up by the users, and negative when bikes are delivered. Δ_i is the safety margin at station i, implying that the number of bikes after service at station i must be within the interval $\left[\Delta_i, Q_i^S - \Delta_i\right]$.

As in the SBRP-model, all stations are classified as either pickup- or delivery stations. In the SBRP-model, the stations were classified based on the difference between the initial and the optimal state. In the DBRP-model, demand must also be considered when classifying the stations, and the station type is defined in accordance with equations (5.32). If the demand at a station balances the difference between the initial and optimal state, $J_i = 0$.

$$J_i = \begin{cases} 1, & \text{if } S_i^0 - O_i - D_i\overline{T}^D > 0 \\ -1, & \text{if } S_i^0 - O_i - D_i\overline{T}^D < 0 \\ 0, & \text{else} \end{cases} \qquad i \in \mathcal{N} \tag{5.32}$$

For the DBRP-model, new inventory variables are introduced. s_{im} represents the number of bikes at station i when service starts at station visit (i, m). s_i^E is still the final state, i.e. the number of bikes at station i at the end of the planning period. Note however that the inventory variables are not integer in the DBRP-model, because they depend on the demand and the continuous time variables.

Table 5.2: New notation for the DBRP-model

Indices	
$o(v)$	Origin station $o(v) \in \mathcal{N}$, for vehicle v
$d(v)$	Artificial destination station $d(v) \in \mathcal{N}$ for vehicle v
Parameters	
$\overline{T}^D$	Time limit for operation of service vehicles, i.e. length of planning period
$T_v^{o(v)}$	Driving time to the origin station for vehicle v
Q_i^S	Capacity of station i
L_v^0	Initial load on vehicle v
β	Weight on deviations in the objective function relative to violations. $\beta = 1$ and $\beta = 0$ implies considering only deviations and only violations, respectively
D_i	Net demand $\big[$bikes / minute$\big]$ at station i
Δ_i	Safety margin at station i
Variables	
s_{im}	Number of bikes at station i when service starts at station visit (i, m)
t_{im}	Time when service starts at station visit (i, m)
l_{imv}	Load on vehicle v after service at station visit (i, m)
q_{imv}	Number of bikes either picked up or delivered at station visit (i,m) by vehicle v
v_{im}^F	Accumulated violations at a full station i between station visit $(i, (m-1))$ and (i, m)
v_{im}^E	Accumulated violations at an empty station i between station visit $(i, (m-1))$ and (i, m)
v_i^{EF}	Accumulated violations at a full station i between station visit $(i, \lvert\mathcal{M}_i\rvert)$ and the end of the planning period
v_i^{EE}	Accumulated violations at an empty station i between station visit $(i, \lvert\mathcal{M}_i\rvert)$ and the end of the planning period

The time variables, t_{im}, specify the time when service starts at station visit (i, m). l_{imv} is the load on vehicle v after service at station visit (i, m). Note that the pickup and delivery quantities, q_{imv}, are indexed by visit number, m. In the DBRP-model, it is necessary to connect the pickup/delivery quantity at a station to a visit number, because we need to know the time of service. Both l_{imv} and q_{imv} are integer variables.

We introduce four sets of variables for accumulated violations. v_{im}^F and v_{im}^E represent the accumulated violations when the stations are full and empty, respectively, at station i between station visit $(i, (m-1))$ and (i, m). A violation when the station is full occurs when someone tries to return a bike, but there are no open slots. When a station is empty, a violation occurs when someone tries to lend a bike. v_i^{EF} and v_i^{EE} are accumulated violations between station visit $(i, |\mathcal{M}_i|)$ and the end of the planning period. Because of the continuous demand, the violation variables are also continuous. Violations when a station is full are often considered worse than violations when a station is empty, and could therefore be given more weight. Additionally, there could be given different weight to violations at different stations based on their importance. Here, all violations are weighted equally.

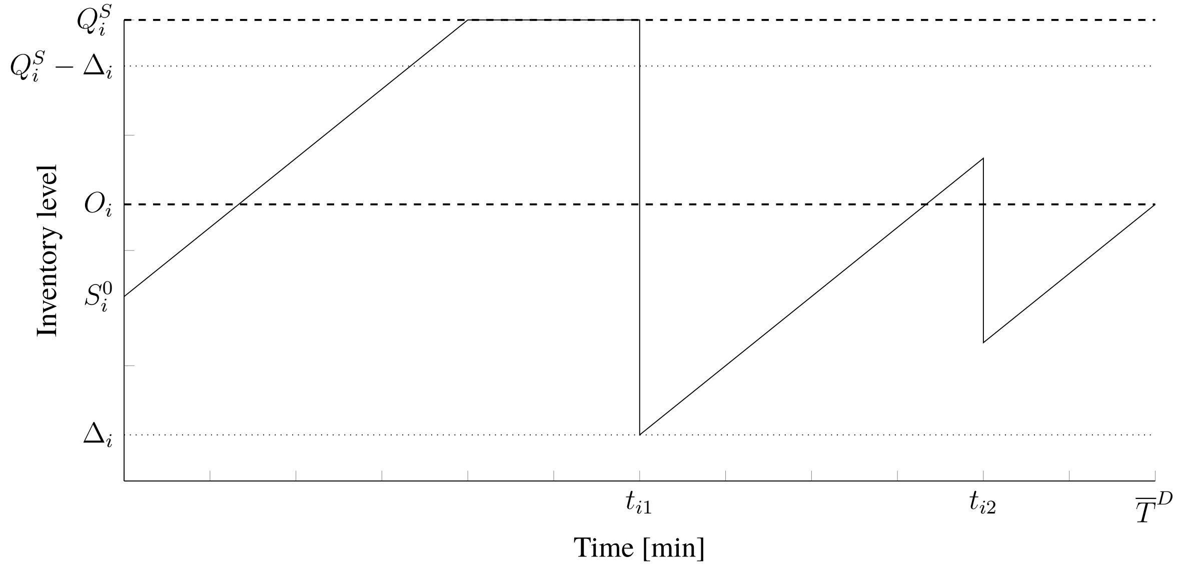


Figure 5.2: Illustration of demand and service at a pickup station, station i

Figure 5.2 illustrates the demand and service at a pickup station. A planning period of $\overline{T}^D$ minutes is used, and the initial inventory is S_i^0. The demand, D_i, increases the inventory level at the station until the station gets full, i.e. reaches Q_i^S. Then, violations might occur until the station is visited by a service vehicle after t_{i1} minutes. Bikes are picked up so that the safety margin, Δ_i, is reached. After t_{i2} minutes another vehicle visits the station and picks up bikes so that the final state, s_i^E, equals the optimal state, O_i, at the end of the planning period, $\overline{T}^D$.

5.2.3 Mathematical Formulation

$$\min \quad \beta \sum_{i \in \mathcal{N}} J_i \left(s_i^E - O_i \right) + (1 - \beta) \left(\sum_{i \in \mathcal{N}} \left(v_i^{EE} + v_i^{EF} \right) + \sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}_i} \left(v_{im}^E + v_{im}^F \right) \right) \tag{5.33}$$

subject to:

$$\sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} x_{o(v)1jnv} = 1 \qquad v \in \mathcal{V} \tag{5.34}$$

$$\sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}_i} x_{imd(v)nv} = 1 \qquad v \in \mathcal{V} \tag{5.35}$$

$$\sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} x_{jnimv} - \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} x_{imjnv} = 0 \qquad i \in \mathcal{N} \setminus \{o(v), d(v)\}, m \in \mathcal{M}_i, v \in \mathcal{V} \tag{5.36}$$

$$\sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \sum_{v \in \mathcal{V}} x_{imjnv} \leq 1 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i \tag{5.37}$$

$$\sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \sum_{v \in \mathcal{V}} x_{jnimv} \leq 1 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i \tag{5.38}$$

$$l_{o(v)1v} - J_i q_{o(v)1v} = L_v^0 \qquad v \in \mathcal{V} \tag{5.39}$$

$$l_{imv} + J_j q_{jnv} - l_{jnv} - Q_v^V (1 - x_{imjnv}) \leq 0$$
$$i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i, j \in \mathcal{N} \setminus \{d(v)\}, n \in \mathcal{M}_j, v \in \mathcal{V} \tag{5.40}$$

$$l_{imv} + J_j q_{jnv} - l_{jnv} + Q_v^V (1 - x_{imjnv}) \geq 0$$
$$i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i, j \in \mathcal{N} \setminus \{d(v)\}, n \in \mathcal{M}_j, v \in \mathcal{V} \tag{5.41}$$

$$q_{imv} \leq l_{imv} \leq \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} Q_v^V x_{imjnv} \qquad i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i, v \in \mathcal{V} \Big| J_i = 1 \tag{5.42}$$

$$0 \leq l_{imv} \leq \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} Q_v^V x_{imjnv} - q_{imv} \qquad i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i, v \in \mathcal{V} \Big| J_i = -1 \tag{5.43}$$

$$t_{im} + T^H q_{imv} + (T_{ij}^D + T^P) x_{imjnv} - t_{jn} - \overline{T}^D (1 - x_{imjnv}) \leq 0$$
$$i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, v \in \mathcal{V} \tag{5.44}$$

$$t_{im} - t_{i(m-1)} - \sum_{v \in \mathcal{V}} T^H q_{i(m-1)v} - \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \sum_{v \in \mathcal{V}} T^P x_{i(m-1)jnv} \geq 0 \qquad i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i \setminus \{1\} \quad (5.45)$$

$$t_{o(v)1} \geq T_v^{o(v)} \qquad v \in \mathcal{V} \quad (5.46)$$

$$t_{d(v)v} \leq \overline{T}^D \qquad v \in \mathcal{V} \quad (5.47)$$

$$s_{i1} + D_i t_{i1} - v_{im}^E + v_{im}^F = S_i^0 \qquad i \in \mathcal{N} \setminus \{d(v)\} \quad (5.48)$$

$$s_{i(m-1)} - \sum_{v \in \mathcal{V}} J_i q_{i(m-1)v} - D_i \left(t_{im} - t_{i(m-1)} \right) - s_{im} - v_{im}^F + v_{im}^E = 0 \qquad i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i \setminus \{1\} \quad (5.49)$$

$$s_{i|\mathcal{M}_i|} - \sum_{v \in \mathcal{V}} J_i q_{i|\mathcal{M}_i|v} - D_i \left(\overline{T}^D - t_{i|\mathcal{M}_i|} \right) - s_i^E - v_i^{EF} + v_i^{EE} = 0 \qquad i \in \mathcal{N} \setminus \{d(v)\} \quad (5.50)$$

$$\Delta_i \leq s_{im} - \sum_{v \in \mathcal{V}} J_i q_{imv} \leq Q_i^S - \Delta_i \qquad i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i \quad (5.51)$$

$$J_i s_i^E \geq J_i O_i \qquad i \in \mathcal{N} \setminus \{d(v)\} \quad (5.52)$$

$$s_{im} v_{im}^E = 0 \qquad i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i \quad (5.53)$$

$$s_i^E v_i^{EE} = 0 \qquad i \in \mathcal{N} \setminus \{d(v)\} \quad (5.54)$$

$$(Q_i^S - s_{im}) v_{im}^F = 0 \qquad i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i \quad (5.55)$$

$$(Q_i^S - s_i^E) v_i^{EF} = 0 \qquad i \in \mathcal{N} \setminus \{d(v)\} \quad (5.56)$$

$$x_{imjnv} \in \{0, 1\} \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, j \in \mathcal{N}, n \in \mathcal{M}_j, v \in \mathcal{V} \quad (5.57)$$

$$q_{imv} \geq 0, \text{integer} \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, v \in \mathcal{V} \quad (5.58)$$

$$0 \leq s_i^E \leq Q_i^S \qquad i \in \mathcal{N} \quad (5.59)$$

$$0 \leq s_{im} \leq Q_i^S \qquad i \in \mathcal{N}, m \in \mathcal{M}_i \quad (5.60)$$

$$t_{im} \geq 0 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i \quad (5.61)$$

$$l_{imv} \geq 0, \text{integer} \qquad i \in \mathcal{N}, m \in \mathcal{M}_i, v \in \mathcal{V} \quad (5.62)$$

$$v_{im}^{F},\ v_{im}^{E} \geq 0 \qquad i \in \mathcal{N}, m \in \mathcal{M}_i \quad (5.63)$$

$$v_{i}^{EF},\ v_{i}^{EE} \geq 0 \qquad i \in \mathcal{N} \quad (5.64)$$

The objective function of the DBRP-model, (5.33), minimizes the total deviation at the end of the planning period and the sum of violations for when the stations are both full and empty during the period. β indicates the relative importance of deviations to violations. Setting $\beta = 0$ would give the minimum number of violations during the planning period, but would make the system little prepared for the following period. Note that by setting $\beta = 1$, the DBRP-model would be quite similar to the SBRP-model. This formulation of the objective function does not consider the time when violations happen, which is discussed in Section 7.2.10.

The routing constraints (5.34-5.38), correspond to constraints (5.2-5.5) for the SBRP-model. The only difference is that $d(v)$ and $o(v)$ are used instead of d, because the vehicles may start and end their routes at any station. Note that constraints (5.34) presume that all vehicles start at different stations. The load constraints (5.39-5.43), handle the load quantities for the vehicles as well as the net pickup at each station visit. The load at each vehicle after leaving the origin station should equal the initial load plus the bikes picked up or delivered at that station. This is assigned by constraints (5.39). Using a big-M-formulation, constraints (5.40-5.41) connect load quantities at the vehicles to net pickup at the stations. Constraints (5.42-5.43) connect the load and pickup/delivery quantities to station visits and assign a bound to the load quantity, for pickup and delivery stations, respectively.

The time constraints (5.44-5.47), take care of the time usage. Constraints (5.44) force the time for a station visit to at least equal the time for the previous station visit plus the handling time, the parking time and the driving time between the two stations. To make sure that two vehicles do not visit a station simultaneously, constraints (5.45) are introduced to separate two station visits with at least the sum of the parking time and handling time for the station visit. Constraints (5.46) assign values to the time variables for the first visit for each vehicle's origin station. Since all vehicles start at an arbitrary location, the time for the first visit to the origin stations must equal at least the driving time from the vehicle's initial locations. All station visits must happen before the dynamic time limit, which is handled by constraints (5.47).

The inventory constraints (5.48-5.51), handle the inventory levels at the stations. Constraints (5.48) assign initial inventory levels, and accumulate violations that occur before the first station visit. The inventory level at the first station visit equals the initial inventory plus the demand until the time for the first station visit. Inventory levels at the other station visits, except the last one, are set by constraints (5.49). The inventory level at these station visits equal the inventory level at the previous station visit plus the number of bikes picked up or delivered by the service vehicle and the net demand since the previous station visit. These constraints also assign the right values to the accumulated violations variables. Constraints (5.50) correspond to constraints (5.49), but apply to the last station visits. They are included to take care of the time between the last station visit and the dynamic time limit, and assign the right values to the inventory levels and violations at the end of the planning period. Note that both the inventory and violation variables can take non-integer values.

To make sure that the quantity loaded at a station is within the safety limits set by Δ_i, constraints (5.51) are introduced. Without enforcing constraints (5.52), the objective function would reward re-positioning of too many bikes. Note however that these constraints may reduce the size of the solution space, but we consider this reduction to be of negligible importance. Compared with the optimal state, there must be more bikes in the final state for pickup stations and fewer bikes in the final state for delivery stations.

Constraints (5.53-5.56) are included to make sure that violations only occur when a stations is actually empty or full. They are clearly necessary when $\beta > 0.5$, else, the violations could replace deviations to balance constraints (5.53-5.56). Also, because the loading quantity is integer and the demand is continuous, these constraints make sure that the violation variables do not take a small value in order to allow loading of one additional bike. These constraints are non-linear, but can be linearized as illustrated below. Non-negativity and integer constraints are given by (5.57-5.64). Constraints (5.59-5.60) also ensure that the inventory levels are below the capacity for all station visits. For the DBRP-model, subtour eliminating constraints are not needed, because the formulation uses a strictly increasing time variable for station visits, t_{im}.

To linearize constraints (5.53-5.56), the binary variables presented in Table 5.3 and constraints (5.65-5.72) are used. The variables indicate whether the stations are full or empty at the station visits and at the end of the planning period. Constraints (5.65-5.72) assign the right values to the variables δ_{im}. They also connect the inventory and violation variables, so violations only occur when the stations are either full or empty. Big-M formulations are used for the linearization. Note that the upper capacity limit in constraints (5.59-5.60) may be omitted when constraints (5.65-5.72) are used, as they constrain the inventory levels.

Table 5.3: Variables for the linearization of constraints (5.53-5.56)

δ_{im}^{F}	1 if station i is full before service at station visit (i, m), 0 otherwise
δ_{im}^{E}	1 if station i is empty before service at station visit (i, m), 0 otherwise
δ_{i}^{EF}	1 if station i is full at the end of planning period, 0 otherwise
δ_{i}^{EE}	1 if station i is empty at the end of planning period, 0 otherwise

$$s_{im} + Q_i^S \delta_{im}^E \leq Q_i^S \qquad i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i \tag{5.65}$$

$$v_{im}^E - D_i \overline{T}^D \delta_{im}^E \leq 0 \qquad i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i \tag{5.66}$$

$$s_{im} - Q_i^S \delta_{im}^F \geq 0 \qquad i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i \tag{5.67}$$

$$v_{im}^F + D_i \overline{T}^D \delta_{im}^F \leq 0 \qquad i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i \tag{5.68}$$

$$s_i^E + Q_i^S \delta_i^{EE} \leq Q_i^S \qquad i \in \mathcal{N} \setminus \{d(v)\} \tag{5.69}$$

$$v_i^{EE} - D_i \overline{T}^D \delta_i^{EE} \leq 0 \qquad i \in \mathcal{N} \setminus \{d(v)\} \tag{5.70}$$

$$s_i^E - Q_i^S \delta_i^{EF} \geq 0 \qquad i \in \mathcal{N} \setminus \{d(v)\} \tag{5.71}$$

$$v_i^{EF} + D_i \overline{T}^D \delta_i^{EF} \leq 0 \qquad i \in \mathcal{N} \setminus \{d(v)\} \tag{5.72}$$

5.2.4 Symmetry Breaking Constraints

Like for the SBRP-model, station visit numbers cause symmetry for the DBRP-model. Hence, constraints (5.26) may be used directly to make sure that the station visits appear in the right sequence. In the DBRP-model, the vehicles start at different locations. This makes each vehicle unique, even though the fleet is homogeneous. Consequently, the symmetry breaking constraints (5.27-5.28) should not be used for the DBRP-model.

5.2.5 Valid Inequalities

The valid inequalities (5.29-5.31) that were introduced for the SBRP-model cannot be used for the DBRP-model. Neither constraints (5.29) nor (5.31) work in a dynamic setting, because they do not take demand during the planning period into account. Constraints (5.30) assume perfect re-balancing, i.e. the optimal state equals the final state $\left(O_i = s_i^E\right)$ for all stations i. Because of the demand it impossible to guarantee perfect re-balancing in the DBRP-model, even with an infinite time horizon is. Consequently, constraints (5.30) cannot be used.

A new set of valid inequalities, (5.73-5.74), is proposed, inspired by constraints (5.29). These constraints connect the q_{imv}-variables to the x_{imjnv}-variables and force some of the x_{imjnv}-variables to take on values closer to one in the linear relaxation. Constraints (5.75-5.76) take inventory level, optimal state and demand into account, and make sure that the vehicles do not move more bikes than necessary. Since demand is continuous and loading quantities integer, the demand has to be rounded down for pickup stations and up for delivery stations. This results in one set of constraints for pickup stations and another set for delivery stations. While constraints (5.73-5.74) handle the sum of all station visits, constraints (5.75-5.76) apply for every visit. Constraints (5.73-5.74) can be seen as an extension of constraints (5.52), which may be relaxed when enforcing one of these.

$$\sum_{v \in \mathcal{V}} \sum_{m \in \mathcal{M}_i} q_{imv} - \left(S_i^0 - O_i - \left\lfloor D_i \overline{T}^D \right\rfloor \right) \sum_{m \in \mathcal{M}_i} \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \sum_{v \in \mathcal{V}} x_{imjnv} \leq 0 \qquad i \in \mathcal{N} \setminus \{d(v)\} \; \Big| \; J_i = 1 \tag{5.73}$$

$$\sum_{v \in \mathcal{V}} \sum_{m \in \mathcal{M}_i} q_{imv} + \left(S_i^0 - O_i - \left\lceil D_i \overline{T}^D \right\rceil \right) \sum_{m \in \mathcal{M}_i} \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \sum_{v \in \mathcal{V}} x_{imjnv} \leq 0 \qquad i \in \mathcal{N} \setminus \{d(v)\} \; \Big| \; J_i = -1 \tag{5.74}$$

$$\sum_{v \in \mathcal{V}} q_{imv} - \left(s_{im} - O_i - \left\lfloor D_i (\overline{T}^D - t_{im}) \right\rfloor \right) \leq 0 \qquad i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i \; \Big| \; J_i = 1 \tag{5.75}$$

$$\sum_{v \in \mathcal{V}} q_{imv} + \left(s_{im} - O_i - \left\lceil D_i (\overline{T}^D - t_{im}) \right\rceil \right) \leq 0 \qquad i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i \; \Big| \; J_i = -1 \tag{5.76}$$

The set of valid inequalities, (5.77), forces the quantity picked up or delivered to be greater or equal to one if a station visit is used.

$$q_{imv} - \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} x_{imjnv} \geq 0 \qquad i \in \mathcal{N} \setminus \{d(v)\}, m \in \mathcal{M}_i, v \in \mathcal{V} \tag{5.77}$$

CHAPTER 6

IMPLEMENTATION

The mathematical models presented in Chapter 5 are implemented in Xpress using the Mosel programming language. This chapter presents the instances used to test the models and the key input data.

6.1 Test Instances

Based on the BSS in Oslo, Norway, five test areas (geographical regions) are generated. Figure 6.1 shows a map of the whole BSS with the five areas highlighted. Details about each area are found in Table 6.1. The areas are simplifications of the actual system, but aim to describe reality accurately. They cover the areas around *the National Theatre*, *Bislett*, *Grünerløkka*, *Grønland*, and *Frogner*, and these names are used to distinct the areas. The areas include between six and 14 stations, in addition to the depot. Some other, larger instances are used to evaluate the maximal solvable problem size.

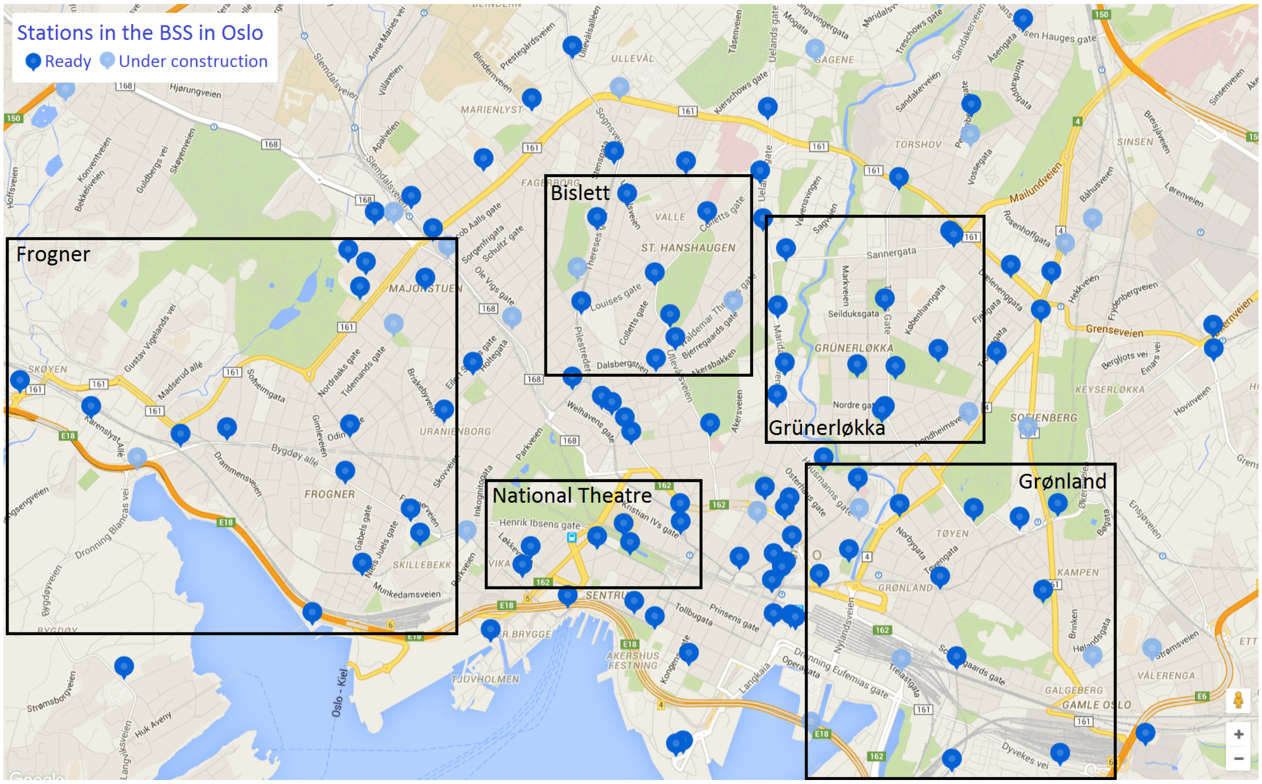


Figure 6.1: The BSS in Oslo, Norway, with the five test areas highlighted

Table 6.1: Comparison of the test areas

Area	# st.	Av. driving time	Time limit	# veh.	Cap. $v = 1$	Cap. $v = 2$
National Theatre	6	2 min	16 min	2	10	10
Bislett	8	6 min	30 min	2	10	15
Grünerløkka	10	6 min	40 min	2	12	12
Grønland	12	5 min	30 min	2	10	10
Frogner	14	7 min	45 min	2	12	12

For each area, three instances are created by varying the initial states, while all other parameters are unchanged. Note that we assume perfect re-balancing for the third instance in each area, making the instances easier to solve because of a simpler structure. Table 6.2 presents some key information about the instances. The maximum number of possible visits is calculated in accordance with the lower bound method. For the DBRP-model, safety margins of two bikes are added for all the stations.

Preliminary testing indicates that the DBRP-model is difficult to solve within reasonable computational time. Therefore, some additional instances are created from the two smallest areas, National Theatre and Bislett. The time limit for re-balancing is increased to 20 minutes for the National

Table 6.2: Comparison of the instances

Instance	**Initial deviation** $\sum_{i \in \mathcal{N}} \lvert S_i^0 - O_i \rvert$	**Max # possible visits** $\sum_{i \in \mathcal{N} \setminus \{d\}} \lvert \mathcal{M}_i \rvert$	**Zero deviation**
National Theatre 1	24	6	No
National Theatre 2	38	8	No
National Theatre 3	48	8	Yes
Bislett 1	32	8	No
Bislett 2	46	9	No
Bislett 3	48	10	Yes
Grünerløkka 1	56	10	No
Grünerløkka 2	64	12	No
Grünerløkka 3	58	12	Yes
Grønland 1	52	12	No
Grønland 2	74	15	No
Grønland 3	62	12	Yes
Frogner 1	70	14	No
Frogner 2	86	16	No
Frogner 3	106	16	Yes

Theatre to allow more possibilities. 15 instances are created for each of the two areas by varying the demand at each station, while all other parameters are unchanged, illustrating 15 different days. Table 6.3 shows a comparison of the total demand in the different instances.

Table 6.3: Comparison of the additional DBRP-model instances. There are 15 instances for each of the two areas. The instances only differ with regards to demand, and are therefore compared by the sum of the absolute demand at all stations. Demand is rounded to nearest integer.

Instance number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
National Th., $\sum_{i \in \mathcal{N}} \lvert D_i \overline{T}^D \rvert$	11	20	35	41	51	45	57	63	56	73	81	72	64	45	16
Bislett, $\sum_{i \in \mathcal{N}} \lvert D_i \overline{T}^D \rvert$	69	74	65	81	52	69	56	38	54	48	38	56	17	75	75

When studying the operational level, the number of service vehicles is fixed. The mathematical formulation can handle any number of service vehicles and a heterogeneous fleet. In Oslo, a city with a little more than 100 stations, there are five service vehicles. In our test instances, two service vehicles are used to get some complexity into the problem and verify that the formulation returns feasible solutions when using more than one vehicle. To illustrate the potency of the model, service vehicles with different capacities for the different instances are used, see Table 6.2. The weighting parameters α and β, from the objective functions, are set to 0.999 and 0.3, respectively.

6.2 Time

All areas include a time matrix for the driving time, T_{ij}^{D}, between all pair of stations. First, matrices with the absolute distances in kilometres between the stations were computed based on the coordinates of the stations. The fact that the relationship between the actual driving distance and the absolute distance between two stations varies from arc to arc is not taken into account. Then, time matrices were created by multiplying the distance matrices with a constant representing the driving time per kilometer. This constant is set to 6 min/km, which implies a driving speed of 10 km/h. The value is verified based on the actual driving times between some pair of stations, given by Google Maps. Differences in speed limits, traffic signals etc. are not considered. An important, but reasonable assumption, is that the driving time from station i to station k is always shorter than the sum of the driving times from i to k, via any station j. This is sometimes called the triangle inequality. The driving time from each vehicle's starting point to their origin station is set to $T_{v}^{o(v)}$.

The maximum times available for re-balancing the system, $\overline{T}^{S}$ and $\overline{T}^{D}$, must also be decided. The limit is set to 16, 30, 40, 60 and 80 minutes for the SBRP-model for the the National Theatre, Bislett, Grünerløkka, Grønland and Frogner, respectively, as shown in Table 6.1. These values correspond to the relative size of the instances. The dynamic time limit, $\overline{T}^{D}$, corresponds to the length of the planning horizon for the DBRP-model, in which the demand is considered constant. By increasing the dynamic time limit, total demand during the planning period increases, which again increases the problem complexity. To get instances of varying complexity, $\overline{T}^{D}$ is set to 20, 40, 50, 60 and 100, respectively. The parking time, T^{P}, includes parking the vehicle, planning the route etc., and is set to one minute. For each bike that is either loaded or unloaded from the service vehicle, a handling time, T^{H}, of 30 seconds is added.

6.3 Demand and States

The initial state, S_{i}^{0}, and the optimal state, O_{i}, are determined for all stations. The optimal states are calculated based on the expected demand for bikes and locks at the stations. By studying historical data, combined with information about the day of week, the hour, the current weather etc., a demand forecast can be made. This is elaborated in Chapter 3.2. For the DBRP-model instances, the demand at the stations is found by multiplying a random number for each station with a fixed number for each instance. Different fixed numbers are chosen in order to simulate days with different demand levels. The initial states are set to values that could represent the actual number of bikes at the stations at a given time.

An instance with one station that is initially balanced, appears like an instance with one station less with regard to problem complexity. To get a correct impression of the problem size, none of the stations in the instances are initially balanced.

For the SBRP-model, all vehicles start and finish empty at the depot. The vehicles in the DBRP-model, on the other hand, can start and finish at any station and with all possible load of bikes. In the DBRP-model, the origin station and initial load for each vehicle are set to fabricated values, but should in practice be the set to the vehicle's actual position and load. All vehicles must finish at an artificial destination station, which has a driving time of zero from all other stations. This is a theoretical station, that must be visited once and only once by all the service vehicles, resulting in an easier model formulation. In practice, the vehicles finish at the penultimate station.

6.4 Maximum Number of Possible Visits

In the mathematical formulation, a set $\mathcal{M}_i$ is defined for each station $i \in \mathcal{N}$. This is the set of possible visits to each station, and hence $|\mathcal{M}_i|$ is the maximum number of possible visits to station i. First, possible ways to compute $|\mathcal{M}_i|$ for the SBRP-model is presented, then differences between $|\mathcal{M}_i|$ for the SBRP-model and the DBRP-model are pointed out.

For the SBRP-model, to be entirely sure that the optimal solution is not cut away due to the size of the set $\mathcal{M}_i$, the maximum number of possible visits for each station must equal the difference between the initial and optimal number of bikes; $|\mathcal{M}_i| = |S_i^0 - O_i|$. This implies that a service vehicle can pick up (or deliver) only one bike at each visit and yet manage to perfectly re-balance the station. The calculation is illustrated using a small example. If a station, say $i = 4$, has a deviation between the initial and optimal state equal to 7 bikes ($|I_4 - O_4| = 7$), it gives $|\mathcal{M}_4| = 7$. This method is called the upper bound method.

If one on the other hand makes the set $\mathcal{M}_i$ as small as possible, being willing to risk cutting away the optimal solution, the maximum number of possible visits can be set to: $|\mathcal{M}_i| = \left\lceil \frac{|S_i^0 - O_i|}{\min_{v \in \mathcal{V}}\{Q_v^V\}} \right\rceil$. This formulation makes use of the capacity of the smallest service vehicle, only allowing to visit a station the minimum number of times that is needed in order to perfectly re-balance the station. Using the example above, with the smallest service vehicle having a capacity of 10 ($\min_{v \in \mathcal{V}}\{Q_v^V\} = 10$) we get $|\mathcal{M}_4| = \left\lceil \frac{7}{10} \right\rceil = 1$, and $|\mathcal{M}_8| = \left\lceil \frac{12}{10} \right\rceil = 2$.

Note that the largest vehicle, $\max_{v \in \mathcal{V}}\{Q_v^V\}$ should be used to calculate $|\mathcal{M}_i|$ in order to find the smallest possible size of the set, while maintaining the possibility to obtain a total deviation equal to zero. However, the formulation using the smallest vehicle is a more realistic choice and is used here. This is in the following referred to as the lower bound method.

The two different ways of calculating the maximum number of possible visits shown above, form reasonable upper and lower bounds for $|\mathcal{M}_i|$. Any value between these two is possible. A value smaller than the lower bound, however, cuts away the possibility of re-balancing the stations perfectly. In addition to testing the lower and the upper bound method for $|\mathcal{M}_i|$, we test a method called lower bound +1 method. This method allows one more visit to each station than the lower bound method.

For the DBRP-model, the calculation of $|\mathcal{M}_i|$ must be slightly modified compared with the SBRP-model. In addition to the difference between the optimal and initial state, the total demand for the period is be considered. This is because the demand impacts the total number of bikes it is desirable to reposition during the planning period. When considering the DBRP-model, the upper bound for $|\mathcal{M}_i|$ can be calculated as follows: $|\mathcal{M}_i| = |S_i^0 - O_i - D_i|$. For the lower bound method, the station capacity must be included in the computation of $|\mathcal{M}_i|$, and the minimum value of the smallest vehicle capacity and the station capacity should be used. This results in the following expression for the lower bound method for the maximum number of possible visits: $|\mathcal{M}_i| = \left\lceil \frac{|S_i^0 - O_i - D_i \overline{T}^D|}{\min\{Q_i^S\ ,\ \min_{v \in \mathcal{V}}\{Q_v^V\}\}} \right\rceil$. Compared with the SBRP-model, we see that the size of the set $\mathcal{M}_i$ is affected by both the dynamic time limit and the demand. Therefore, $|\mathcal{M}_i|$ is typically larger for the DBRP-model than for the SBRP-model.

CHAPTER 7

COMPUTATIONAL STUDY

Using the instances and input data presented in Chapter 6, different formulations, constraints and extensions of the mathematical models from Chapter 5 are thoroughly tested. Both the quality of the solution and the computational time are evaluated and discussed. We further elaborate on the size of solvable instances and illustrate how the models can be used as decision support for the system operators.

7.1 Computational Analysis of the SBRP-model

Solving the SBRP-model yields routes for the overnight re-balancing. Even though the problem is static, the service vehicles cannot be expected to wait several hours for their routes to be determined. Consequently, the computational time is an important issue, and in this section different formulations are tested and compared aiming at shorten it. Further the usefulness of the model is discussed and some of its possibilities illustrated.

7.1.1 Technicalities

Specifications of the computer and software used to solve the SBRP-model are listed in Table 7.1. To test the different formulations, a base case is defined, consisting of the formulations yielding the best results in the preliminary testing. The base case is shown in Table 7.2. In each section, it is explicitly defined how the formulation that is tested deviates from the base case. Of practical reasons, all executions are terminated after 3,000 seconds.

Table 7.1: Details of computer and solver used for the SBRP-model

Processor	Intel Core i7-3770 CPU @ 3.40GHz
RAM	16 GB
Operating system	Windows 7 Enterprise 64-bit
Xpress-IVE Version	1.24.06 64-bit
Xpress Optimizer Version	27.01.02
Mosel Version	3.8.0

Table 7.2: Base case for the testing of different constraints and formulation for the SBRP-model

Reduction of variables	Yes
Objective function	Weighted objective function (5.1)
Subtour elimination	MTZ with strengthening (5.20)
Symmetry breaking constraints	Visit numbers (5.26) and length of route (5.27)
Valid inequalities	(5.29-5.31)
Maximum number of possible visits	Lower bound method

7.1.2 Variable Reduction

The mathematical formulation contains many binary and integer variables. By only creating those variables that are actually needed, the number of variables can be reduced significantly. As a consequence, the computational time is expected to be reduced. Identifying the variables that are predetermined to take the value 0 before solving the model, makes it possible to omit creating them. Some of the variable reductions are listed below. Note that the triangle inequality is assumed to be valid.

- The arc-variables between i and j, where $i = j$ are not created, because it is never beneficial to depart from a station and then return directly to the very same station.

- Since the vehicles start out empty, they should always go directly to a pickup station from the depot. Thus, variables for the arcs from the depot directly to delivery stations are not created. Of similar reasons, the arcs from pickup stations directly to the depot are not generated.

- In accordance with constraints (5.2-5.3), the arcs from the depot, d, to all station visits for the last $|\mathcal{V}|$ visit numbers in $\mathcal{M}_d$ are not created. In the same manner, the variables that end up in the depot for the first $|\mathcal{V}|$ visit numbers are omitted.

- One of the inequalities for the 1-PDTSP from Hernández-Pérez and Salazar-González (2007), states that the arcs between a pair of stations will never be traversed if the sum of the quantity to be picked up or delivered exceeds the service vehicle's capacity. For this inequality to be valid, a zero-deviation solution must be obtained, the two stations must be of the same type, and both stations must be visited maximum once.

The model is run both with and without the variable reduction. In our test instances, the number of variables is reduced 20.8-40.4 %, and the computational times decreases between 68.9 and 99.5 %. On average, the computational time is reduced by as much as 98.5 %. Fixating the vehicle sequence at the depot, probably contribute to a great part of the reduction in computational time, as it removes symmetry. Even a small reduction in the number of binary variables may result in a significantly smaller B-&-B-tree, which in turn contributes to a large decrease in the computational time.

7.1.3 Subtour Eliminating Constraints

In Section 5.1.3, three different sets of subtour eliminating constraints are proposed. First, the MTZ-constraints, originally presented by Miller et al. (1960), secondly a strengthened variant of the MTZ-constraints by Desrochers and Laporte (1991), and finally a recent formulation proposed by Chemla et al. (2013).

Table 7.3 shows a comparison between the three subtour eliminating formulations, and also a formulation without subtour elimination. With only a few exceptions, the problem is easier to solve without subtour eliminating constraints. This was anticipated. We see that the solution becomes better without subtour elimination in terms of total deviation between optimal and final state, and that the number of subtours increases with the problem size. These solutions are however infeasible. Note that the shortest computational times are associated with uncertainty as they may be affected by other tasks run on the computer. Conclusions based on the shortest computational times must therefore be drawn carefully.

The formulation by Chemla et al. (2013) generally performs worse than the MTZ-formulations, and is outperformed by the strengthened MTZ-formulation for all test instances. Thus this formulation

Table 7.3: Comparison of computational times for different subtour eliminating constraints. The numbers in the headline refer to the constraint numbers in the mathematical formulation. Dev. is the total number of deviations in the solution. Zero deviation implies perfect re-balancing
*Miller et al. (1960), **Desrochers and Laporte (1991), ***Chemla et al. (2013)

Instance	**Without subtour elimination**			**With subtour elimination**			
	Dev.	#subtours	Comp.time	Dev.	(5.19)*	(5.20)**	(5.23-5.25)***
National Th. 1	4	1	0.14 s	6	0.17 s	0.19 s	0.34 s
National Th. 2	14	0	0.22 s	14	0.31 s	0.34 s	0.70 s
National Th. 3	0	0	0.22 s	0	0.23 s	0.20 s	0.45 s
Bislett 1	12	1	0.28 s	14	0.45 s	0.64 s	1.28 s
Bislett 2	10	1	0.06 s	12	0.24 s	0.44 s	0.84 s
Bislett 3	0	0	0.50 s	0	0.60 s	0.62 s	1.37 s
Grünerløkka 1	8	1	1.67 s	8	1.35 s	1.25 s	3.77 s
Grünerløkka 2	16	1	3.86 s	18	7.30 s	7.00 s	29.08 s
Grünerløkka 3	0	2	0.52 s	0	2.00 s	1.25 s	4.74 s
Grønland 1	14	2	3.08 s	18	28.08 s	8.40 s	69.27 s
Grønland 2	30	2	3.49 s	32	20.80 s	17.00 s	41.65 s
Grønland 3	0	2	0.19 s	0	0.72 s	0.40 s	0.66 s
Frogner 1	16	3	273.00 s	18	88.00 s	69.00 s	384.09 s
Frogner 2	22	3	3.60 s	28	75.50 s	15.30 s	22.47 s
Frogner 3	0	2	0.78 s	0	1.40 s	1.07 s	2.93 s
Average	9.7	n/a	19.51 s	11.2	15.15 s	8.21 s	37.58 s

can be rejected. When looking at the two MTZ-formulations, the original MTZ-formulation seems to perform well at the small instances. But, since the strengthened MTZ-formulation performs better for all the nine largest instances, we conclude that this variant of the MTZ-formulation is the best. Using this formulation results in a tighter bound on the visit sequence number variables, as shown in constraints (5.21-5.22), which evidently is beneficial. Hence, we use the strengthened MTZ-formulation by Desrochers and Laporte (1991) throughout the thesis.

In the MTZ-formulations implemented in Xpress, the big-M (called $\overline{A}$ in the constraints) equals the sum of the maximum number of possible visits over all stations. Ways to reduce the size of the big-M is not pursued, although it is expected to give some improvement. Reducing the size of the big-M would require use of problem specific information, and is thus not easily implemented in a general model.

7.1.4 Symmetry Breaking Constraints

The problem at hand has symmetrical solutions, which is common for routing problems with more than one vehicle. To remove some of that symmetry, constraints (5.26-5.28) are introduced. All symmetry breaking constraints may be used separately, but not necessarily together. The visit sequence constraints (5.26) may be used with any of the two others, because they handle a different type of symmetry. Constraints (5.27) and (5.28), on the other hand, remove the same type of symmetry in two different ways, and should therefore not be used together. All the symmetry breaking constraints are tested separately, and in addition, constraints (5.26) are tested in combination with both constraints (5.27) and (5.28). The average computational times are presented in Table 7.4.

Table 7.4: Comparison of computational times for the different symmetry breaking constraints. Only average values are shown. For the first instance in each area, $|M_i| = 1$ for all the stations, and the visit sequence constraints (5.26) are inactive. Since the fleet is homogeneous for the Bislett area, constraints (5.27-5.28) are inactive for those instances. When the constrains are inactive, the corresponding test run is denoted with ”n/a”. There are three rows with average values in the table in order to make the values comparable.

* Average computational time, all instances

** Average computational time, excluded the instances where $|M_i| = 1, \forall i \in \mathcal{N}$

*** Average computational time, excluded the Bislett instances

Instance	**Without con.**	**Visit seq.** (5.26)	**# st. visited** (5.27)	**Time used** (5.28)	**(5.26) & (5.27)**	**(5.26) & (5.28)**
Average*	15.42 s	n/a	n/a	n/a	8.21 s	6.89 s
Average**	11.37 s	5.45 s	n/a	n/a	4.80 s	7.37 s
Average***	19.16 s	n/a	14.77 s	10.75 s	10.12 s	8.46 s

Evidently, the symmetry breaking constraints for the visit sequence, (5.26), are very effective. The computational time is reduced for 9 out of 10 instances, and is on average reduced by 52 %. The value of using these constraints is expected to increase with the size of the sets $\mathcal{M}_i$. A small test on the National Theatre 3 instance illustrates their importance. When the lower bound + 1 method is used, the computational time for this instance is reduced from > 3000 s. to 0.25 s., by including constraints (5.26).

Looking at the symmetry breaking constraints handling the homogeneous fleet, there is a significant improvement when using constraints (5.27) or (5.28). Constraints (5.28) are faster for the larger instances and also seem to perform a little better than (5.27) in combination with constraints (5.26), and are therefore recommended. Note that this result implies a change from the base case.

7.1.5 Valid Inequalities

Three valid inequalities are presented in Section 5.1.5, namely constraints (5.29-5.31). These have been tested for the different instances, and the results for the Frogner instances and average results for all the instances are shown in Table 7.5.

Table 7.5: Computational times when using different valid inequalities for the Frogner instances and average results for all instances. The table shows the results without valid inequalities, each inequality tested separately and all three inequalities tested together. The right column shows reduction in comp. time when using all valid inequalities compared to using none. Not all valid inequalities are active for all instances, resulting in three rows in the table to make the values in each row comparable. ”n/a” indicates that the constraints in the corresponding test run are inactive.
$^{+}$Hernández-Pérez and Salazar-González (2007), $^{++}$Chemla et al. (2013)
* Average computational time when using all the valid inequalities
** Average computational time when a zero-deviation solution can be obtained
*** Average computational time, excluded the instances where $|M_i| = 1, \forall i \in \mathcal{N}$

Instance	**Without**	**Con.(5.29)**	**Con.(5.30)$^{+}$**	**Con.(5.31)$^{++}$**	**All**	**Red. using all**
Frogner 1	1736.00 s	33.00 s	n/a	n/a	69.00 s	96.0 %
Frogner 2	154.00 s	27.0 s	n/a	156.00 s	15.30 s	90.1 %
Frogner 3	232.00 s	18.80 s	215.00 s	135.00 s	1.07 s	99.5 %
Average*	154.43 s	8.17 s	n/a	n/a	8.21 s	94.7 %
Average**	69.11 s	n/a	47.00 s	n/a	n/a	n/a
Average***	56.19 s	n/a	n/a	49.83 s	n/a	n/a

Studying the computational times with and without constraints (5.29), it is clear that these constraints have a positive effect. The computational times are reduced for 13 out of 15 instances, and the average reduction is 94.7 %.

Constraints (5.30), introduced by Hernández-Pérez and Salazar-González (2007), are only active when there is enough time available to obtain perfect re-balancing. The results prove that these constraints are useful, as the computational times are reduced for all instances, on average by 32 %.

The third set of constraints, (5.31), presented by Chemla et al. (2013) requires more than one possible visit ($|\mathcal{M}_i| >= 2$) for at least two stations to have effect. It is hard to conclude whether these constraints give a positive contribution to the model or not, as the results point in different directions. However, for the Frogner 3 instance, these constraints give a considerable decrease in computational times, from 232.00 s to 135.00 s. The average result also indicates a slightly positive effect.

All the valid inequalities combined reduce the computational times significantly for most of the instances, on average by as much as 95 %. The effect is negative for three of the instances, but the positive contributions on the others compensate for this. When constraints are included, but inactive, they increase the computational time. This is clearly visible for Frogner 1, where including constraints (5.30-5.31) double the computational time, compared to only including constraints (5.29). For the Frogner 3 instance, all the constraints are active, and including them seems very useful.

We conclude that constraints (5.29) should always be included, and constraints (5.30-5.31) should be included when we know a priori that they will be active. The value of using valid inequalities is expected to increase with the problem size.

7.1.6 Maximum Number of Possible Visits

In Section 6.4, different alternatives for setting the maximum number of possible visits to each station, $|\mathcal{M}_i|$, are discussed. In Appendix B, Section 4.2, three alternatives are tested to see how they influence the computational times and the quality of the solution. All the computational times and the differences in the solutions between the lower bound method and the two other methods are stated in Table 5 in Appendix B.

On average, the computational times increase about 100 times when the lower bound +1 method is used, compared to the lower bound method. Using the upper bound method, the computational time is more than 3,000 seconds for many instances, even some of the smallest. For all instances, the number of deviations in the optimal solution is the same with all methods. However, when the lower bound +1 and the upper bound method are used, there is sometimes found solutions where the vehicles use less time than with the lower bound method. Because the deviations are considered more important than the driving time, the solutions from the lower bound +1 and the upper bound method are considered to be only marginally better. For larger instances than those used here, it is however likely that the quality of the solution would improve more when using the lower bound +1 method compared to the the lower bound method.

The importance of the size of the sets $\mathcal{M}_i$ can also be explained theoretically. The arc-flow variables used in the mathematical formulation use station visits, (i,m), making it possible to consider each possible station visit a distinct node in the graph. Hence, adding one element to the set $\mathcal{M}_i$ for one station i, is equivalent to adding a node to the graph. Each node added to the graph results in many new binary variables. Looking only at the variable x_{imjnv}, adding one new node gives $\left(2|\mathcal{V}|\sum_{j\in\mathcal{N}\setminus\{i\}}|\mathcal{M}_j|\right)$ new binary variables; one new variable for each vehicle both arriving and departing the node to all other nodes. Considering that one new binary variable in the worst case may double the size of the B-&-B tree, it is necessary to limit the number of nodes in the graph, and hence limiting the maximum number of possible visits to each station.

Both the theory and the computational results presented in this section imply that the lower bound method should be used in the practical implementation for the SBRP-model, when evaluating both the solution quality and the computational time.

7.1.7 Time Limit for Re-balancing

By changing the time limit for re-balancing, the optimal solution could change, as the size of the solution space either increases or decreases. Changing the time limit could also affect the computational time. In this section we take a deeper look at the time limit for re-balancing, $\overline{T}^S$, and the time constraints (5.12).

Figure 3 in Appendix B shows how the computational time varies with different time limits, $\overline{T}^S$, for the Grønland 1 instance. It is evident that the time limit has radical impact on the computational time, and the computational time is shown to peak when the number of deviations is slightly above zero.

When $\overline{T}^S$ is small, the problem is severely restricted and easy to solve. By setting a short time limit, many stations are out of reach for the vehicles. For the Grønland 1 instance, with a time limit of 12 minutes (very short), only 5 out of 12 stations may be part of the solution because of the driving, handling, and parking times that are needed. When the time limit increases, the deviation gets closer to zero, the solution space gets bigger, and the problem gets harder to solve. With a number of deviations close to zero, it is hard to find good lower bounds. As the time limit gets even higher, the time constraints are no longer binding and it becomes easier to find good lower bounds, because there exists many feasible routes resulting in zero deviation. Finding good lower bounds makes it possible to cut away great pars of the B-&-B tree and is therefore essential to the efficiency of the B-&-B algorithm.

The time constraints, (5.12), give the time limit for re-balancing the system. Studying these constraints, it is possible to say something about the value of additional time. In practice, this information may be used as decision support for the system operator. The operator must then decide if she will expand or restrict the time limit, by evaluating the operating costs versus the improvement in the solution. It is natural to think that this information is obtainable using the dual variables (or *shadow prices*) of the time constraints. They would give the change in the objective value for a unit increase on the right hand side of the constraints, i.e. increasing the time limit by one unit. As this is a MIP problem, the dual variables cannot be used, and the value of additional time is found by solving the problem multiple times for different time limits and record the objective values.

The results from the analysis of the Grünerløkka 2 instance are presented in Figure 7.1. Note that a low objective value is desirable, which means few deviations and little time usage. The objective value as a function of the time limit is shown to be partly flat and partly linearly decreasing. On average, a unit change in the time limit gives a change of 1.3 in the objective value in the region that is examined here. Note that the ”dual value” may be interpreted as the slope of the curve. Hence, the ”dual value” would be 0 when the curve is flat and negative when the curve is decreasing.

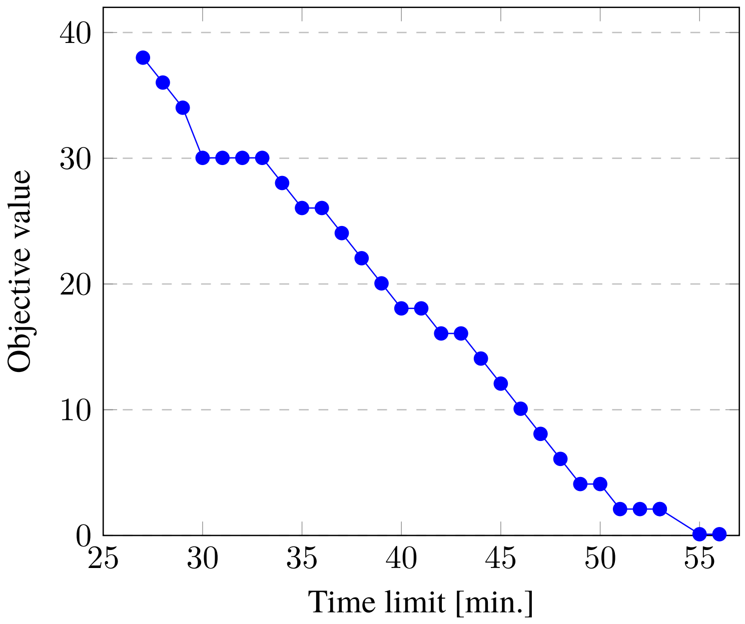


Figure 7.1: The objective value as a function of time limit, $\overline{T}^S$, for the SBRP-model, for the Grünerløkka 2 instance

Referring to the Grünerløkka 2 instance in Figure 7.1, the slope is approximately -2.0 in the interval between $\overline{T}^S = 36$ and $\overline{T}^S = 40$. In this interval, each unit increase in the time limit results in the objective value improving by approximately 2, meaning that two additional bikes are re-positioned. With $\overline{T}^S = 55$, the objective value is close to zero, implying that the system is perfectly re-balanced. Then the solution only negligible improves by a further increase in the time limit. This analysis is only valid for the Grünerløkka 2 instance, but still the results give insight to how the time limit may affect the objective value.

7.1.8 Size of Solvable Instances

The number of stations in an instance obviously affects the computational time. Instances with varying number of stations are tested to determine the capability of the model. As the computational time is strongly affected by the time limit that is used (see Section 7.1.7), the results in this section only indicate a general pattern. The lower bound method is used for setting the maximum number of possible visits, $|\mathcal{M}_i|$.

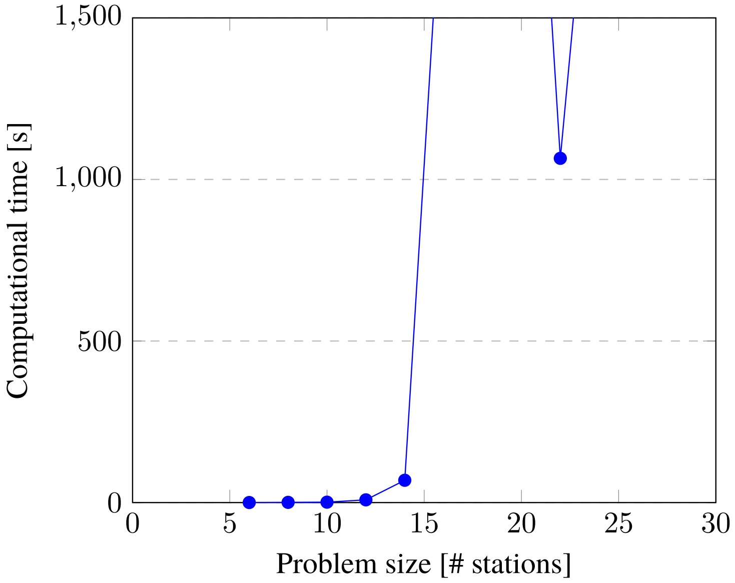


Figure 7.2: Computational time for different number of stations, SBRP-model

Computational times for different instance sizes are shown in Figure 7.2. The computational time is less than one second for the smallest instances, and significantly grows with an increasing number of stations. The computational time sky rockets between 14 and 17 stations. Solving the instance with 14 stations to optimality took 69 seconds, while the instance with 17 stations was not solved to optimality in 3,000 seconds. None of the larger instances were solved to optimality within 3,000 seconds, except the instance with 22 stations, that was solved in 1,065 seconds. The instance with 22 stations probably has an advantageous problem structure, making it easier to solve. However, it is clear that the model in general is able so solve instances with about 15 stations in reasonable time. Solving larger instances would require other solutions methods, e.g. decomposition, extended formulation, or heuristics.

7.1.9 Service Vehicles

Increasing the number of vehicles results in new flow variables, new pickup/delivery variables, and new sets of constraints. Hence the model complexity escalates. Having multiple identical vehicles also introduces symmetry, necessitating effective symmetry breaking constraints. On the other hand, utilizing many vehicles may improve the solution. The same is true for expanding the vehicle capacity. These possible changes are tested and discussed in this section.

Table 7.6 illustrates how the computational time is affected by increasing the number of vehicles for the Grønland 1 instance. The instance is run with a big enough time limit for the system to be perfectly re-balanced by only one vehicle. This results in a fair comparison of the different runs, as the number of station visits stay about the same. We see from the table that the computational time grows rapidly with increasing fleet size. When increasing the number of vehicles above three vehicles, it seems like the growth in computational time declines. This is assumed to be due to too many vehicles compared to the number of stations. Imagine an instance with ten stations and five vehicles, where all vehicles are determined to drive directly from the depot to a pickup node and all stations can only be visited once. This problem is quite easy to solve and the solution makes little sense. Note that when the system is required to be perfectly re-balanced, there exists no feasible solution for this instance using only one vehicle and the lower bound method for the maximum number of possible visits. To conclude, increased number of service vehicles results in a significant growth in computational time, which can be explained by an increased amount of variables and constraints.

Table 7.6: Computational time for different fleet sizes, Grønland 1 instance, assuming perfect re-balancing

Number of vehicles	**Computational time**
1	n/a
2	1.6 seconds
3	23.7 seconds
4	37.4 seconds

For the system operator it may be difficult to decide whether to acquire an extra service vehicle or not. To support this decision making process, the model can be used to quantify the effect of adding or removing a service vehicle, in terms of number of deviations. The National Theatre 2, Bislett 2 and Grünerløkka 1 instances are run with different number of service vehicles and the results are presented in Table 7.7. We see that the increase in the objective value is higher when changing from one to two vehicles, than from two to three.

Table 7.7: Objective value for different fleet sizes

Instance	1 vehicle	2 vehicles	3 vehicles
National Theatre 2	23.99	14.02	4.04
Bislett 2	26.00	12.05	6.08
Grünerløkka 1	26.01	8.07	2.11

To show how the objective value is affected by the vehicle capacity, Q_v^V, the model is solved multiple times for different values. This is depicted in Figure 7.3 for the National Theatre 2 instance. As expected, increased vehicle capacity leads to reduced objective value; reduced number of deviations and/or reduced driving time. For the National Theatre 2 instance, the objective value reaches its lowest level when the vehicle capacity is six. At that point, the time limit restricts the objective value from decreasing further. Between $Q_v^V = 1$ and $Q_v^V = 6$, the objective value is reduced from 30.00 to 14.02, implying an average reduction of 3.2 per unit capacity increase. Figure 7.3 will look different for all instances, but the overall pattern is the same; the objective value decreases with increased capacity. The system operator is referred to cost–benefit analysis, for determining if the reduction in total deviation and time usage is worth the cost of an additional service vehicle or increased vehicle capacity.

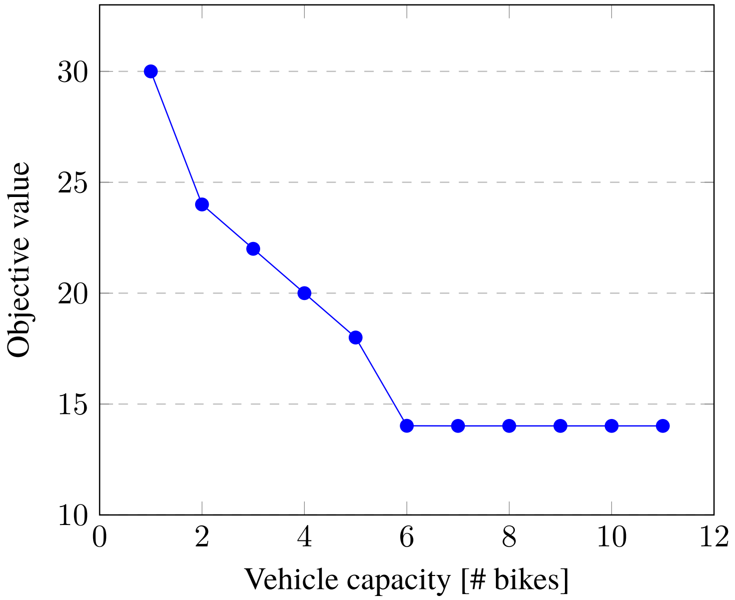


Figure 7.3: The objective value as a function of vehicle capacity, $Q_v^{\mathcal{V}}$, for the National Theatre 2 instance

7.1.10 Alternative Formulations of the Objective Function

In this section we shortly describe possible ways of re-formulating the objective function. Changing the objective function could yield different solutions as input to the system operator.

An alternative way of handling time usage in the objective function is to minimize the time used by the vehicle driving the longest route. This is called a minimum makespan formulation (MSF), and could be favourable if a fixed number of drivers were to work a given amount of hours. Using the current objective function, (5.1), may result in solutions where one vehicle drives a much longer route than the others. This will especially occur when the time constraints (5.12) are relatively loose.

The deviations in the objective function could be handled differently. One alternative is to utilize intervals, rather than a fixed number, to describe the optimal state. This provides more flexibility to the model, presumably making it harder to solve, but it may be more realistic. Another alternative is to punish large deviations relatively more than small, for instance by squaring the deviation. Whether one of these alternatives should be used rather than the current formulation, depends on the information available to the system operator and her utility function.

A third option is to assume perfect re-balancing, i.e. not consider deviations at all, only minimizing time usage or costs. Making this assumption, multiple adjustments could be made to the original formulation. The resulting problem would be equivalent to the 1-PDTSP with multiple vehicles (Hernández-Pérez and Salazar-González, 2003) (if the lower bound method for $|M_i|$ is used). This problem is simpler than the original, but also less realistic. When there is limited time available, the model needs to prioritize which stations to re-balance. We argue that this is in accordance with the real world problem, where time is a scarce resource.

7.2 Computational Analysis of the DBRP-model

The DBRP-model bears many similarities with the SBRP-model, as pointed out in Chapter 4. Thus, many of the results from the SBRP-model are applicable for the DBRP-model as well, and are not replicated. This section evaluates the computational results from the SBRP-model in a dynamic context. Computational testing of the DBRP-model is presented when the results from the SBRP-model are inapplicable. This is especially true for the valid inequalities and when determining the maximum size of solvable instances. Preliminary testing indicated that the DBRP-model is difficult to solve within reasonable time, even for really small instances, thus are the size of solvable instances important in the analysis of the DBRP-model.

7.2.1 Technicalities

Specifications of the computer and software used to solve the DBRP-model are listed in Table 7.8, and the base case is listed in Table 7.9. When necessary, it is explicitly defined how the formulation that is tested deviates from the base case. Because the DBRP-model should be used in a dynamic setting and regularly re-solved, long computational times are undesirable. All computations are therefore stopped after 1,000 seconds.

Table 7.8: Details of computer and solver used for the DBRP-model

Processor	Intel Core i5-4300 CPU @ 2.50GHz
RAM	4 GB
Operating system	Windows 10 Pro 64-bit
Xpress-IVE Version	1.24.06 64-bit
Xpress Optimizer Version	27.01.02
Mosel Version	3.8.0

Table 7.9: Base case for the testing of different constraints and formulation for the DBRP-model

Reduction of variables	Yes
Beta, β	0.3
Valid inequalities	(5.73-5.76)
Symmetry breaking constraints	Visit numbers (5.26)
Maximum number of possible visits	Lower bound method

7.2.2 Variable Reduction

In the SBRP-model, extensive variable reduction is possible, partly because all vehicles must start and finish empty at the depot. Such variable reduction is however not possible for the DBRP-model, as there is no depot. For the DBRP-model, as for the SBRP-model, self-loops do not make sense, and these arc-variables are therefore not created.

If a station is defined to be a pickup station, violations because the station is empty should never occur. Correspondingly, a delivery station should never get violations because the station is full. Consequently, these violation variables do not need to be created. When inequality (7.1) is fulfilled for a station, the violation variables for full stations do not need to be created for this station. Similar for violation variables for empty stations when inequalities (7.2) are fulfilled.

$$S_i^O - D_i\overline{T}^D \leq Q_i^S \implies v_{im}^F \ \& \ v_i^{EF} \text{ are not created} \qquad i \in \mathcal{N} \qquad (7.1)$$

$$S_i^O - D_i\overline{T}^D \geq 0 \implies v_{im}^E \ \& \ v_i^{EE} \text{ are not created} \qquad i \in \mathcal{N} \qquad (7.2)$$

In Section 7.1.2, variable reduction is proved to be very effective. Thus, variable reduction is implemented for the DBRP-model, and no analysis of the effect is performed.

7.2.3 Subtour Eliminating Constraints

Unlike for the SBRP-model, subtours are no issue for the DBRP-model because the model uses a strictly increasing time variable ensuring that it is impossible to make one station visit twice. Subtour eliminating constraints may however work as valid inequalities and introducing them may possibly strengthen the mathematical formulation. Introduction of the MTZ-constraints with strengthening (5.20) is tested in Section 7.2.5. This is the set of subtour eliminating constraints that yields the best results for the SBRP-model.

7.2.4 Symmetry Breaking Constraints

Because of the visit numbers, there are symmetrical solutions when solving the DBRP-model, similar to the SBRP-model. This symmetry is handled by constraints (5.26). These constraints are tested in Section 7.1.4 for the SBRP-model, and prove to considerably reduce the computational time. The effect of introducing these symmetry breaking constraints is assumed to be similar for the DBRP-model, and this is verified in the computational testing. Even though the fleet of vehicles may be homogeneous in our DBRP-model, the vehicles are considered unequal because they start at different locations. Constraints removing symmetry in a homogeneous fleet should therefore not be introduced in the DBRP-model.

7.2.5 Valid Inequalities

The valid inequalities proposed for the DBRP-model, constraints (5.73-5.77), are different from those introduced in the SBRP-model and are thus not comparable with them. Constraints (5.73-5.77) are tested and the results are shown in Table 7.10. The MTZ-formulation by Desrochers and Laporte (1991), introduced in Section 5.1.3 is also tested, to see if it may strengthen the formulation and have effect as a set of valid inequalities.

It is difficult to see a clear pattern in the results, as the computational times differ substantially. The valid inequalities used in the base case, (5.73-5.76), yield on average the lowest computational times, but this is mostly due to the low computational time in instance number 14. In Table 7.11 the inequalities are compared by the number of times they are fastest and slowest. From this table

Table 7.10: Comparison of computational times using different valid inequalities for the DBRP-model. Tested on 15 instances from the National Theatre area. The instances are distinguished only by the demand.

Inst.	**None**	**Base case+(5.77)** (5.73-5.77)	**Base case** (5.73-5.76)	**Imp. LP-rel.** (5.73-5.74)	**Bound on q_{imv}** (5.75-5.76)	**Base case+MTZ** (5.73-5.76)+(5.20)
1	0.09 s	0.07 s	0.10 s	0.05 s	0.02 s	0.07 s
2	0.50 s	0.74 s	1.00 s	0.80 s	0.43 s	1.16 s
3	1.28 s	1.26 s	1.90 s	1.61 s	1.32 s	2.84 s
4	4.44 s	2.81 s	2.80 s	3.21 s	3.16 s	3.18 s
5	9.18 s	20.94 s	11.20 s	48.03 s	5.87 s	17.69 s
6	0.61 s	0.53 s	0.50 s	0.58 s	0.97 s	1.11 s
7	3.69 s	6.56 s	4.80 s	6.75 s	4.58 s	5.45 s
8	11.35 s	5.71 s	3.90 s	2.89 s	14.71 s	5.31 s
9	22.70 s	30.77 s	37.20 s	38.59 s	26.77 s	36.82 s
10	1.42 s	1.03 s	1.20 s	1.64 s	1.34 s	1.54 s
11	2.69 s	2.39 s	5.00 s	6.47 s	3.69 s	4.74 s
12	20.90 s	18.57 s	29.20 s	28.09 s	30.69 s	50.41 s
13	26.24 s	26.43 s	24.30 s	43.72 s	28.99 s	42.73 s
14	75.64 s	36.26 s	10.00 s	75.21 s	53.93 s	147.90 s
15	0.04 s	0.02 s	0.10 s	0.05 s	0.04 s	0.36 s
Avg.	12.05 s	10.27 s	8.88 s	17.18 s	11.77 s	12.54 s

it becomes evident that it is beneficial to include some of the valid inequalities for 13 out of the 15 instances. We would recommend using what we call the base case + constraints (5.77), i.e. (5.73-5.77), as this results in the lowest computational time for one third of the instances and newer performs worst. Similar testing is done for 15 instances from the Bislett area and these results point in the same direction. The MTZ-formulation, (5.20), results in general in higher computational times, and is therefore rejected.

Table 7.11: Comparison of the set of valid inequalities by the number of times they are fastest and slowest for the 15 test instances. The MTZ-constraints are excluded from this comparison as the results from 7.10 indicate that they clearly are disadvantageous.

Set of inequalities	**# fastest**	**# slowest**
None	2	2
Base case + (5.77)	5	0
Base case	4	4
Imprv. LP-relax.	1	6
Bound on q_{imv}	3	3

7.2.6 Maximum Number of Possible Visits

Our DBRP-model uses a set of possible visits to each station, $\mathcal{M}_i$, similar to the SBRP-model. The size of this set depends on the method chosen to calculate the maximum number of possible visits to each station, as explained in Section 6.4. Additionally, for the DBRP-model, the size of $\mathcal{M}_i$ depends on the demand at station i. In Section 7.1.6 different methods for calculating the maximum number of possible visits to each station are tested on the SBRP-model, and we conclude that the lower bound method should be used. By allowing more visits to each station, the computational time sky rockets and the quality of the solution only gets marginally better. Because demand is taken into account in the DBRP-model, $|\mathcal{M}_i|$ is normally even bigger than for the SBRP-model. Consequently, the lower bound method should also be used for the DBRP-model as well.

7.2.7 Time Limit for Re-balancing

Section 7.1.7 describes how the computational time depends on the time limit for re-balancing for the SBRP-model. There is a similar time limit in the DBRP-model; $\overline{T}^D$. With a short time limit, the solution space is small and little computational time is needed. When the time limit increases, the number of possibilities increases and the computational time rises. As the time limit increases even further, the computational time falls for the SBRP-model. This is however not the case for the DBRP-model because of the constant demand for bikes and locks. Because of the demand, will an increasing time limit not result in slack in the time restriction, but bigger sets $|\mathcal{M}_i|$ and an increasing problem complexity. The time limit for the DBRP-model should not however be needlessly long, as the model is regularly re-solved, and the time limit should only stretch through periods with similar demand. The peak in computational time seen for the SBRP-model for a certain time limit is not present for the DBRP-model.

7.2.8 Size of Solvable Instances

The SBRP-model manages to solve problems with up to 15 stations within reasonable time, when using the lower bound method for calculating $|\mathcal{M}_i|$. In this section, we determine the maximum size of solvable instances for the DBRP-model.

The most important characteristic for describing the size of the problem is the total maximum number of possible visits, $\sum_{i \in \mathcal{N}} |\mathcal{M}_i|$. As explained in Section 6.4 does this value depend on the number of stations, the size of the demand, and the time limit, as well as the capacity of the service vehicles and the stations. 15 instances from the areas National Theatre and Bislett are created to illustrate the size of solvable instances for the DBRP-model, as described in Section 6.1. The lower bound method is used for calculating $|\mathcal{M}_i|$. The results are shown in Figure 7.4. Although there are many outliers, there is clearly an exponential relationship between the computational time and the total maximum number of possible visits. The computational time sky rockets when the maximum number of possible visits is about twelve. All the instances tested with $\sum_{i \in \mathcal{N}} |\mathcal{M}_i| \leq 7$ have a computational time of less than one second, while none of the instances with $\sum_{i \in \mathcal{N}} |\mathcal{M}_i| \geq 13$ are solved within 1,000 seconds. As the figure illustrates, the computational times are very high, even for quite small instances. Thus, we suggest that heuristics should be developed, in order to yield meaningful results for realistically sized instances.

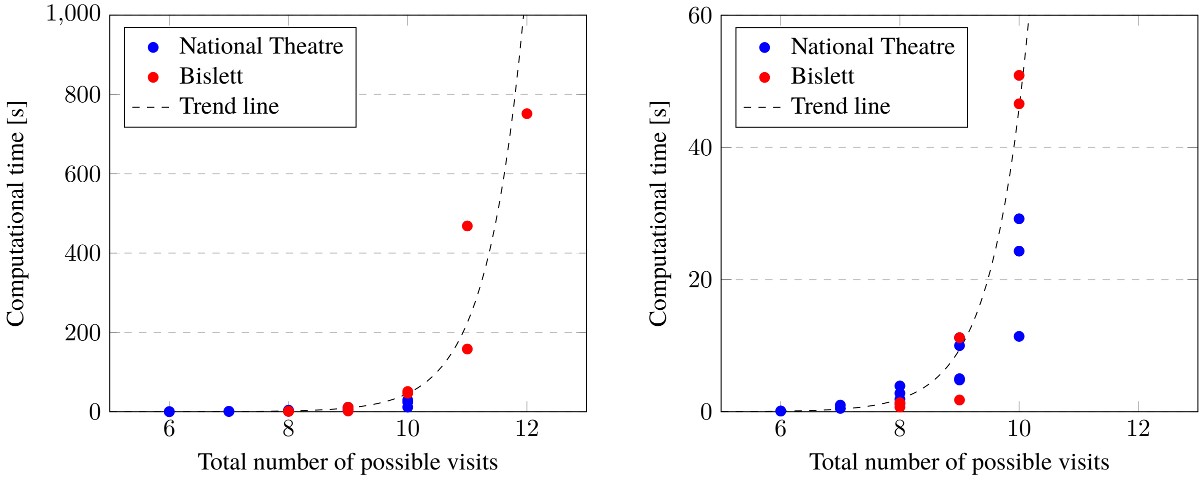


Figure 7.4: Computational time varies with the total maximum number of possible visits, $\sum_{i \in \mathcal{N}} |\mathcal{M}_i|$. Instances in the National Theatre area are marked with blue and the ones from Bislett are marked with red. The dotted line is an exponential regression line. The figure to the right is more detailed, as the y-axis is cut on 60 seconds.

Considering the number of service vehicles, the discussion from Section 7.1.9 for the SBRP-model is considered to be applicable for the DBRP-model as well. An increased fleet size results in more variables and constraints, raising the problem complexity and computational time.

7.2.9 Solution Stability

Since we consider a dynamic problem, the model should be regularly re-solved to get updated decisions based on new information, e.g. every time a service vehicle arrives at a station. Therefore, the most important output from the DBRP-model is the first loading decision and the subsequent choice of station visit. This section studies how the decisions vary for different time limits, $\overline{T}^D$, and if the solutions stabilize when the time limit increases. Tests are performed on two instances from the National Theatre area, and the results are depicted in Table 7.12.

Table 7.12: The first loading decision and the subsequent routing decision for different values for $\overline{T}^D$ for two instances from the National Theatre area. With a capacity of ten and an initial load of five, pickup and delivery quantities are limited to five bikes for both service vehicles. The time limit is restricted to 45 and 60 minutes for the two instances, respectively, because the computational times get too high with a higher time limit.

Instance 1	$v = 1$		$v = 2$	
Time Limit, $\overline{T}^D$	**Loading at origin**	**Next station**	**Loading at origin**	**Next station**
10	-3	3	+1	1
15	-5	3	+1	1
20	-5	3	0	1
25	-2	3	+1	1
30	-4	3	+1	1
35	-4	3	+1	1
40	-5	3	+1	1
45	0	1	+5	2
Instance 2	$v = 1$		$v = 2$	
Time Limit, $\overline{T}^D$	**Loading at origin**	**Next station**	**Loading at origin**	**Next station**
10	+5	3	-1	5
15	+5	3	-2	1
20	+5	3	-3	6
25	+5	3	-4	2
30	+5	3	-5	1
35	+5	3	-5	1
40	+5	5	-5	6
45	+5	3	-5	6
50	+5	3	-5	2
55	+5	3	-1	6
60	+5	4	-2	6

From Table 7.12, we see that the solution is dependent on the chosen time limit for re-balancing, and that it is difficult to find a clear pattern. The solutions are sometimes quite stable, e.g. the next station decision for both vehicles in Instance 1. However, we see for vehicle 2 in Instance 2 that the solutions vary a lot. One may expect the results to become more stable when $\overline{T}^D$ increases, but that does not seem to be the case. In Instance 1, the routing decision is stable until $\overline{T}^D = 45$. The solution changes because the maximum number of possible visits to station 1 increases, as a result of an increased time limit. With more possible visits, the vehicle could find a more effective route. This is tested by increasing the maximum number of possible visits for all time limits for Instance 1. Doing this, we observe that vehicle 1 should visit station 1 also for $\overline{T}^D = 40$ and $\overline{T}^D = 35$. We conclude that the maximum number of possible visits constrains the problem. With $\overline{T}^D \leq 30$ is it still optimal to visit station 3. To find the optimal solution for Instance 1, the DBRP-model should either be solved with $\overline{T}^D \geq 45$ or with $\overline{T}^D \geq 35$ and more possible visits to each station.

For vehicle 1 in Instance 2, the loading quantity at the origin station is stable for all time limits. The initial load of bikes at the vehicle is five and all these bikes are needed at the origin station. For vehicle 2 on the other hand, the loading quantity varies. It is worth noticing that when the time limit changes, the total demand changes as well. Consider a station with initial inventory of ten bikes, a negative demand of 0.1 bike per minute, an optimal state of ten bikes, and a planning period of 30 minutes. When considering a planning period of ten minutes instead, we still want to end up with 10 bikes after 30 minutes, but we expect delivery of two bikes during the time between our new planning period and 30 minutes. Thus, the optimal state should be adjusted to compensate for the length of the planning period, and set to eight bikes. Such adjustment is not however done for the instances in Table 7.12, which to some degree may explain why the loading quantities are varying.

An other reason for the solutions to be unstable with regards to the time limit is thresholds caused by the violations. If the demand is higher than what the service vehicles can manage, an increasing demand as a result of a longer time limit results in more deviations. When demand increases further, the stations will at some point experience violations. Since violations are given a higher weight in the objective function than deviations, this could result in a new prioritization of station visits.

To sum up, is it clear that the solutions are dependent on the chosen time limit for re-balancing. For some instances, the decision for loading at the origin station and the next station visit stay the same for almost all time limits, while for other instances the solutions change a lot. It is therefore hard to conclude which time limit should be used when solving the DBRP-model.

7.2.10 Objective Function

A weakness with the DBRP-model is that the objective function (5.33) does not consider the time when violations happen. Section 3.2 shows how the demand could be estimated based on historical data and how these estimates are associated with uncertainty. Estimates regarding the near future are more secure than estimates reaching further into the future. Consequently, violations expected to happen in the near future are more likely to actually happen than violations expected to happen later. The most probable violations are indisputably the most severe, and one could argue that violations should be discounted based on the time they are expected to happen. Considering the time of the violations, the objective function would become non-linear. Since we assume relatively short planning periods and frequent re-solving of the model, discounting of violations are not implemented, as the effect would be small. If we were considering longer periods however, violations should be discounted.

The parameter β is introduced in Section 5.2 and represents the weighting of deviations relative to violations in the objective function. Deviations are calculated based on gap from the optimal state and reflect violations happening after the planning period. Zero deviations represents the situation where the probability of future violations is minimized. Thus, the relative weighting of deviations in the objective function can be considered a discounting of violations happening after the planning period. Beta is set by the operator of the BSS, and has great impact on the solution.

Figure 7.5 shows how the number of violations and deviations change with different values for β for two Bislett instances. Other instances that are tested show the same pattern. Only β-values between 0 and 0.5 are tested. A β above 0.5 is never beneficial, as more weight is given to deviations than to violations. This is unfavourable, because violations are more severe and violations are closer in time than deviations, and hence should be given more weight. To clearly distinguish between the relative importance of violations and deviations would we recommend always setting β below 0.4.

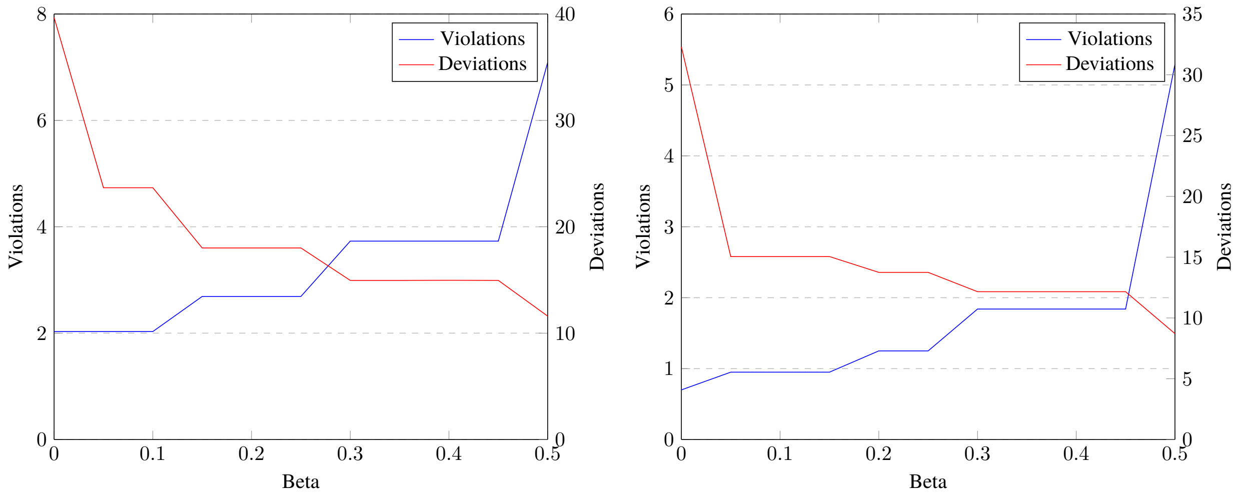


Figure 7.5: Number of violations and deviations for different values for β are shown in blue and red, respectively. Note that violations are shown on the left axis, and deviations on the right axis. The Bislett area is used, and results for instance 9 are shown to the left and instance 10 to the right.

From the figure, it is clear that a β above zero should be used. For β equal to zero, only violations are weighted. If there are symmetrical solutions, the model will not necessarily find the solution with the least number of deviations. However, with a β only marginally higher than zero, deviations are given some weight and better solutions are found. From the figure, it seems like a $\beta \geq 0.15$ is reasonable for finding good solutions. Hence, we recommend β-values in the interval from 0.15 to

0.4, as it seems to yield stable solutions with a quite low number of both violations and deviations.

The time limit, $\overline{T}^D$, impacts the importance of violations relative to deviations. With a low time limit, deviations are close in time, and hence more important. This could be regarded by setting a higher beta. With a higher time limit, deviations are however more uncertain, and violations should be given more weight by setting a lower beta.

CHAPTER 8

COMPARISON OF SOLUTION METHODS

The SBRP-model and the DBRP-model presented in Chapter 5 are useful for solving small instances and for describing the problems precisely. It is however shown in Sections 7.1.8 and 7.2.8 that they only can be solved for instances with up to about 15 and eight stations, respectively. Short computational time is essential when operating in a dynamic setting and the drivers need updated routes within few seconds. Consequently, the problem size is a pressing issue for the DBRP-model, and the DBRP is therefore the focal point of this chapter. Through this chapter, six simple evaluation techniques used to determine the next station visit in a dynamic setting are presented and evaluated. We call them rules of thumb (ROTs) to emphasize their simplicity and distinguish them from heuristics that are used to solve a specific mathematical model. The six ROTs and using the solution from the DBRP-model are different solutions methods for the DBRP and are tested and compared.

8.1 Comparing the SBRP-model with Rules of Thumb

In Appendix B, Section 4.3, two different greedy ROTs are created to imitate the operators' current behavior for overnight re-balancing. Solutions from these ROTs are compared with the solution from the SBRP-model presented in Section 5.1. The SBRP-model is shown to perform significantly better than the two ROTs. Further comparison of the SBRP-model with ROTs is not done here.

8.2 Rules of Thumb for Dynamic Bicycle Repositioning

In this section, the six different ROTs for re-balancing during the day are introduced. They use the number of violations, probability of violations, deviations from the optimal state and the driving time between the stations in various ways to prioritize which station to visit next. A new decision is taken every time a vehicle is finished with loading at a station. The different ROTs are developed in cooperation with Lars Felin, COO at Urban Infrastructure Partner who operates the BSS in Oslo (Felin, 2016). The ROTs are intentionally made rather simple to better differentiate them.

8.2.1 Number of Violations

In Section 3.2.3, we presented a procedure to compute the number of violations at a station within a time period, given a deterministic demand. Using this information, we have developed a ROT that finds the next station visit by evaluating the expression in equation (8.1), i.e. choosing the station $j \in \mathcal{N}$ with the highest number of violations divided by the driving time from the current location, i. We call this method the Num.viol. method. $\mathcal{N}$ is the set of stations that the service vehicle is able to serve with the initial load on board. A service vehicle is able to serve a station if the load is more than three bikes before visiting a delivery station or if there is room for more than three additional bikes on the vehicle before visiting a pickup station. Note that if all stations have zero violations, the vehicle is waiting at its current location.

$$\arg\max_{j \in N} \left\{ \frac{\text{Number of violations at station } j}{\mathrm{T}_{ij}} \right\} \tag{8.1}$$

An advantage with this ROT is that it considers the actual number of violations and therefore may be effective at prioritizing the stations where the effect of a visit is largest. The number of violations is divided by the driving time to maximize the effect per time. A vehicle should for instance choose the nearest of two stations with the same number of violations. Of similar reasons are the measure for all the ROTs divided by the driving time to the relevant station.

It may be favourable to give more weight to the most critical stations, i.e. the stations with the highest number of violations. To do this, the number of violations could for instance be squared. We introduce a ROT that is similar to the Num.viol. method, except that the number of violations is squared. This method is mathematically described in equation (8.2) and is called the Num.viol.sq. method. Squaring leads to a higher relative weighting of stations with a high number of violations that are far away from the current location of the service vehicle. The vehicle therefore uses more time driving than without the squaring.

$$\underset{j \in N}{\arg\max} \left\{ \frac{(\text{Number of violations at station } j)^2}{\mathrm{T}_{ij}} \right\} \tag{8.2}$$

8.2.2 Probability of Violations

Instead of using the number of violations, this ROT makes the vehicles go to the station with the highest probability of violations that the vehicle is able to serve with the initial load, as described in equation (8.3). The procedure of computing the probability of violations at a station is presented in Section 3.2.3. This ROT is called the Prob.viol. method and may be favourable because it considers uncertainty in demand better than the other ROTs. It does for instance take into account the probability of violations because of a full station, even though bikes are expected to be picked up by the users at this station.

$$\underset{j \in N}{\arg\max} \left\{ \frac{\text{Probability of violations at station } j}{\mathrm{T}_{ij}} \right\} \tag{8.3}$$

Additionally, we introduce a ROT squaring the probability of violations, and name it the Prob.viol.sq. method. This method, described in equation (8.4), prioritizes high probability of violations relatively more than the Prob.viol. method, but they are otherwise equal. Imagine two stations A and B, with 100 % and 50 % probability of violations, respectively, and a driving time of one minute. The Prob.viol. method would give that station A is twice as important at station B, while the Prob.viol.sq. method would give that A is four times as important.

$$\underset{j \in N}{\arg\max} \left\{ \frac{(\text{Probability of violations at station } j)^2}{\mathrm{T}_{ij}} \right\} \tag{8.4}$$

8.2.3 Nearest Station

The ROT called the Nearest st. method tells the vehicle to go to the nearest station the vehicle is able to serve. We also require the number of violations at the station to be larger than zero for the vehicle to be allowed to go there, implying that if no station has violations, the vehicle is waiting at its current location. This ROT is mathematically described in equation (8.5), and is very similar to the STR proposed by Brinkmann et al. (2015b). A favourable characteristics with this ROT is that it prioritizes to make many station visits. It may however end up with only re-balancing a small part of the system.

$$\underset{j \in N}{\arg\min} \ \{\mathrm{T}_{ij}\} \tag{8.5}$$

8.2.4 Number of Deviations from the Optimal State

Another option is to prioritize the stations based on the deviations from the optimal state. According to this ROT, the vehicles should visit the stations with the highest ratio for the number of deviations squared divided by the driving time, in accordance with equation (8.6). We call it the Dev.sq. method. This ROT may not be the best to avoid violations for the current planning period, but may be good at preparing the system for future periods. If the optimal state somewhat reflects demand in the current period, this ROT could perform well also in the current period. The Dev.sq. method is quite similar to the current practice for re-balancing in the BSS in Oslo (Felin, 2016). When using this method, there is less need for a proper demand forecast or calculations of number/probability of violations. The method prioritizes large stations close to the current location of the service vehicle.

$$\underset{j \in N}{\arg\max} \left\{ \frac{(\text{Deviation from optimal state at station } j)^2}{\mathrm{T}_{ij}} \right\} \tag{8.6}$$

8.2.5 Loading decision

All the ROTs determine the loading quantity in the same way. The service vehicle loads or unloads bikes until the optimal state at the station is reached or until the vehicle has reached its capacity limit. It may sometimes be suboptimal to load bikes until the optimal state is reached, because the vehicle rather should drive to a more critical station. This is not however considered here.

8.3 Implementation

Both the DBRP-model and the ROTs are tested in simulations performed in Excel. In this section, details of the simulation are elaborated and the test instances presented.

8.3.1 Simulation in Excel

To test the different ROTs, we have developed a simulation model. The model uses fixed-time incrementing, where each time increment results in a new row in the spreadsheet. For each time increment, all stations are updated with pickup and delivery events, and the current inventory and accumulated violations are adjusted. The location of and load on the service vehicle is also updated.

The service vehicle will at some point arrive at a station. Bikes are then loaded or unloaded until the optimal state at the station is reached or until the vehicle has reached its capacity limit. As long as the vehicle is located at a station, a new loading decision is taken every time increment. When the loading decision is to do nothing, the vehicle is routed to a new station based on the chosen ROT. Note that the loading decisions are equal for all ROTs, it is only the routing decisions that differ. Driving times between stations are found in driving-time matrices similar to those introduced in Section 6.2.

The time increment in the simulations is set to 0.5 minutes, as this corresponds to the lowest time unit in the problem; the time used to load or unload one bike. All simulations consider the system over a two-hour period. The ROTs that use number of violations and probability of violations both compute these values based on the next 30 minutes.

The simulations are implemented and executed in Microsoft Excel 2016, and the ROTs are created using Microsoft Visual Basic for Applications (VBA) version 7.1.

8.3.2 Random Numbers and Exponential Distributed Events

The exponential distribution is commonly used for modeling queues, and Seo et al. (2015), Raviv and Kolka (2013), and Schuijbroek et al. (2013) use it to model demand in BSSs. Time between events cannot be negative, which is one of several reasons why the exponential distribution is suited.

In the simulations, time between pickup and deliveries of bikes at the stations are assumed to be exponentially distributed with a pickup rate and a delivery rate at station j equal to λ_j and μ_j, respectively. Then, expected time between pickup events equals $1/\lambda_j$ and the expected time between delivery events equals $1/\mu_j$. These parameters could be derived from some forecast based on historical data, see Section 3.2.1. Assuming a Poisson distribution where the arrival times are independent and identically distributed, and the number of arrivals in the time interval $[0, t]$ has a mean of λt, the arrivals can be represented using an exponential distribution with a mean of $1/\lambda$.

To check if a pickup and/or a delivery event has occurred at a station during the previous time increment, the acceptance-rejection method is used. Random numbers are generated using the $=\texttt{RAND}()$-function in Excel, and these numbers are compared to the cumulative distribution functions for pickup and delivery events. To simplify, the probability distributions are constant for the entire simulation period.

The exponential distribution assumes all events to be independent and identically distributed. This assumption may be inaccurate, as users often arrive in small groups, e.g. in connection with public transportation. When using fixed-time incrementing, it is not possible to have more than one pickup and one delivery event at each station in a single time increment. In the simulations, the length of the time increments are 0.5 minutes, which is small enough to make this simplification insignificant.

8.3.3 Instances

Two different areas are used in the simulations, a small area with six stations and a larger area with 20 stations. Both areas are based on parts of the BSS in Oslo. The smaller area is needed to simulate the system when using the DBRP-model to determine the next station visit, as the model is hard to solve for the larger instances. Driving times are estimated as described in Section 6.2, and vary between one and 20 minutes. The stations have capacities between 13 and 20 slots, and all of them have a safety margin of two slots.

Three different instances are created for the area with six stations, and five for the area with 20 stations. The instances are created by setting mean pickup and mean delivery to random numbers between zero and 0.3 bikes per minute for each station, and the initial inventory to a random number within the station's capacity. Instances are manually verified to have a total demand close to zero (pickup $-$ delivery ≈ 0) and total initial inventory close to half of the total station capacity. The demand at the stations are in the same order of magnitude as typical stations in the Oslo BSS during a busy period. The optimal state at each station is calculated based on the demand, using a simplification of the procedure presented in Section 3.2.4.

8.4 Comparison of the DBRP-model with the Rules of Thumb

The DBRP-model is tested in a simulation on the instances from the area with six stations. While the system behaviour is simulated, routing and loading decisions are found by solving the DBRP-model with Xpress. The performance of the DBRP-model is compared with the ROTs from Section 8.2.

8.4.1 Simulation Details

As presented in Section 7.2.8, the DBRP-model is incapable of solving large instances. The test instances are therefore made so that $\sum_{i \in \mathcal{N}} |\mathcal{M}_i|$ varies between eight and eleven. For simplicity there is only one vehicle, with a capacity of 15 bikes. The system is simulated over a period of two hours. This system is clearly too small to be a realistic imitation of an actual BSS, but the results may still give valuable information about the performance of the DBRP-model and the ROTs.

Three instances with different expected demand and different initial and optimal states are created. These instances are simulated three times with different realizations of the random variables, which for instance may represent three different days. The DBRP-model and all the ROTs are tested using the same realization of the random variables, making the results from each instance comparable. Each solution method is tested nine times in total. Note that since the instances only have six stations, the decision of which station to visit next is sometimes trivial.

In the DBRP-model, the time limit is set to 60 minutes and the weighting between deviations and violations in the objective function, β, is set to 0.3. The ROTs are programmed in VBA and called directly from the simulation procedure. As the DBRP-model is programmed in Xpress, information must be sent between the two programs and the simulation takes additional time. Solving the DBRP-model once in Xpress may take several minutes, and one simulation run requires multiple solutions of the model.

8.4.2 Results

Table 8.1 shows the results from the test runs described in Section 8.4.1. The solution methods are tested nine times, which is insufficient to draw any conclusions, but the results may still indicate which solution methods that are favourable and some characteristics of the different solutions. Both average total violations, loading quantity and number of station visits are presented for all the solution methods.

The main objective when repositioning bikes in a BSS during the day is to minimize the number of violations. Based on this measure, the DBRP-model and the Prob.viol.sq. method perform best. They both obtain average total violations of 9.5 for the tested instances. Even though this is better than the other solution methods, it is hard to conclude whether these methods actually are better based on only nine runs. The DBRP-model is however assumed to generally perform well, which is discussed in Section 8.4.3.

Table 8.1: Comparison of the performance of the DBRP-model and six different ROTs. All methods are tested on nine instances, based on three different demand parameter sets. The values in the table represent the averages over these nine simulation runs. The instances have six stations, one vehicle and are simulated over a period of two hours.

Solution method	**Avg. tot. violations**	**Avg. tot. loading quantity**	**Avg. # of station visits**
DBRP-model	9.5	88.0	11.4
Num.viol. method	14.0	79.9	6.8
Prob.viol. method	15.9	74.8	15.2
Num.viol.sq. method	13.0	78.6	6.6
Prob.viol.sq.	9.5	91.2	10.7
Nearest st. method	13.1	78.5	6.7
Dev.sq. method	11.2	89.2	10.6

When looking at the average total loading quantity, it is evident that the ROTs leading to the largest loading quantities also perform well when looking at total violations. The DBRP-model, the Prob.viol.sq. method and the Dev.sq. method have the largest average total loading quantities and the lowest average total violations.

Another characteristic is that the ROTs that on average visit the most stations during the period, also have the lowest number of violations. The exception from this pattern is the Prob.viol. method, which has the highest number of station visits, but also the lowest loading quantity and the highest number of violations. This behavior is explained by the fact that a station may be perfectly balanced, but still have a probability of both full and empty violations. Because the probability is divided by the driving time, a station with 10 % probability of both full and empty violations and a driving time of one minute is prioritized over a station with 90 % probability of violations and ten minutes driving time. With two stations being close to each other, the Prob.viol. method may result in solutions where the vehicle drives back and forth between these two stations, doing quite few loading operations. Because of this behavior, the Prob.viol. method is not recommended. The Prob.viol.sq. method does not however have this problem and seems to perform very well.

Table 8.1 indicates that the results for the Num.viol. method, the Num.viol.sq. method and the Nearest st. method are quite similar. For an instance with only six stations, there is often only one station of the right type with more than zero violations. When this is the case, these three ROTs chose the same station. For instances with more stations or higher demand, these rules would probably give more distinct results and be easier to distinguish from each other.

To sum up, the DBRP-model and the Prob.viol.sq. method seems to be the two best solution methods for the tested instances. Visiting many stations and loading many bikes are shown to be favourable characteristics of a ROT.

8.4.3 Characteristics of the DBRP-model in the Simulation

The DBRP-model performed well in the simulations, but the number of violations could probably be reduced by tailoring the DBRP-model to the simulation setting.

In the simulation of the DBRP-model, the loading quantity is computed together with the routing decision when the service vehicle leaves a station. The loading may therefore start up to 20 minutes after the decision is taken. During this time, there can be many pickup and delivery events, and the loading decision is not necessarily optimal when the loading actually happens. If the loading decision is updated when the service vehicle arrives at a station, the DBRP-model probably performs better, especially when the demand is high. For the ROTs, a new decision is made before every single loading operation, making these solution methods more flexible.

To improve the simulation results from the DBRP-model, deviations could be omitted for the entire or for the last part of the simulation period. Deviations are considered for an optimal state one hour into the future. The DBRP-model therefore makes decisions that probably are beneficial to get a well-functioning system in the longer run, but not when considering violations in the simulation results.

While there might be only few minutes left of the simulation, the DBRP-model plans a total route for the next 60 minutes. This results in solutions that are good with a long perspective, but sometimes not when looking at only the last move. Changing the time limit in the DBRP-model in accordance with the remaining time of the simulation would improve its simulated performance.

To conclude, the number of violations for the DBRP-model in Table 8.1 could be reduced if the model is customized to the simulation. The value of using such a model in practise is evident when comparing to simple ROTs, but modifications of the model are needed for solving larger instances.

8.5 Comparison of the Rules of Thumb

The testing in Section 8.4 was made on quite small instances, and only a few simulations was conducted. To make a better comparison of the ROTs, we have created larger instances and conducted numerous simulations.

8.5.1 Simulation Details

The ROTs are tested on five different instances from an area with 20 stations. For simplicity, the area is serviced by only one vehicle, which has a capacity of 23 bikes, making it identical to the actual service vehicles operating in Oslo. The instances differ with regard to demand and initial and optimal state, and the system is simulated over a period of two hours. In addition to simulating the system when using the six ROTs from Section 8.2, the system is simulated when the vehicle is doing nothing and when the vehicle chooses its next destination at random, as a basis of comparison. Note that the Random method determines the loading and unloading quantities in the same way as the ROTs, see Section 8.2.5.

All solution methods are simulated 1,000 times for every instance and the total number of violations in the system is recorded for each simulation. This is done using DataTable from the What-if analysis tool in Excel, resulting in 5,000 simulations of the system for every ROT. Unlike the simulation in Section 8.4, the realization of the random variables is different for every simulation, necessitating many simulations to make the results for the different ROTs comparable. With this approach is it impossible to compare results from single simulations, only average values for whole sets of simulations.

8.5.2 Results

Results from the simulations are showed in Table 8.2. Each instance is simulated 1,000 times, resulting in 1,000 violation values for each ROT, and the average values shown in the table. We note that doing nothing and choosing station visits at random are the worst and second-worst methods for every instance. The Num.viol. method has the lowest total average number of violations, but it is not much better than the Num.viol.sq., the Prob.viol., and the Nearest st. method. Out of the six ROTs the Prob.viol. method performs worst, and the Dev.sq. method results in a few more violations than the top four.

T-tests and F-tests are performed to see if the results from the six ROTs are significantly different. F-

Table 8.2: Comparison of six different ROTs for determining the next station visit. The results are compared to doing nothing and to choose station visits at random. Each ROT is simulated 1,000 times for five different instances. The instances are based on an area with 20 stations and one vehicle, and are simulated over a two hour period.

Solution method	Average total violations					Avg.tot.viol.
	Instance 1	Instance 2	Instance 3	Instance 4	Instance 5	
Num.viol. method	39.8	76.5	43.5	66.5	87.6	62.8
Prob.viol. method	53.4	105.2	89.3	101.9	105.6	91.1
Num.viol.sq. method	40.4	77.0	45.8	68.8	90.8	64.6
Prob.viol.sq. method	41.5	78.6	42.7	67.4	88.8	63.8
Nearest st. method	40.1	77.5	44.0	65.8	90.9	63.7
Dev.sq. method	44.1	84.0	44.3	67.7	94.9	67.0
Random	68.6	123.7	81.7	116.7	132.7	104.7
Do nothing	94.1	148.0	108.7	148.8	165.8	133.1

tests are used to test the null hypothesis that the variances of two populations are equal. Results from F-tests executed in Excel indicate that the variances are unequal, hence two-sample t-tests assuming unequal variances are performed. A t-test of two samples yields a p-value, which is the probability of observing the two samples given that they represent the same distribution. We choose a significance level of $\alpha = 1\%$, i.e. two methods are to considered to yield different number of violations if it is less than one percent probability of getting the observed results given that the methods yield the same number of violations. Then, there is no instance where one ROT is significantly better than all the others. Table 8.3 shows the ROT with the lowest average total violation for each instance, as well as the ROTs with an average value insignificantly worse than the best. With $\alpha = 1\%$, it is impossible to distinguish between the ROTs listed for each instance.

Table 8.3: For every instance, the table shows which ROT that yields the lowest average number of violations for 1,000 simulations. The table also shows which ROTs that according to a t-test have more than 1% probability of yielding the same number of violations as the best.

Instance	Best ROT	ROTs with p-value $\geq 1\%$	
1	Num.viol. method	Nearest st. method	Num.viol.sq. method
2	Num.viol. method	Num.viol.sq. method	Nearest st. method
3	Prob.viol.sq. method	Num.viol. method	Nearest st. method
4	Nearest st. method	Num.viol. method	Prob.viol.sq. method
5	Num.viol. method	Prob.viol.sq. method	

From Table 8.3 we see that the Num.viol. method has the lowest average violation for three of the instances and is among the best ROTs for all five instances. Further we see that the Nearest st. method is among the best for four instances, the Prob.viol.sq. method is among the best for three instances, and that the Num.viol.sq. method is among the best for two instances. When using a simple ROT for deciding the next station visit, one should therefore choose one of the four methods highlighted here. Our results indicate further that the Num.viol. method. is the preferred ROT.

It is possible that the driving matrix (the topography of the area) favours some of the solution methods to others. When considering an even larger system we are expecting the Nearest st. method to perform worse, as the vehicle is likely to get stuck in one area. Of similar reasons we are expecting the Num.viol.sq. method and the Prob.viol.sq. method to improve relative to the Num.viol. method and the Prob.viol. method when the problem size increases.

A weakness with the simulations is that all instances are taken from the same area. To better illustrate the differences between the ROTs, it would be interesting to do simulations from different areas. Another possible weakness is that the system is simulated over a two hour period. This may be insufficient to shed light on the differences between the ROTs. By increasing the planning horizon it is likely that differences between them would be easier to spot.

The best and worst ROTs, namely the Num.viol. method. and the Prob.viol. method, differ with 28,3 in average violations. The simulated instances have 20 stations and are served by one service vehicle over a two hour period. By comparison, the BSS in Oslo is open 18 hours a day, has a total of 100 stations, and is served by five service vehicles. This implies that using a clever ROT instead of a poor one, could reduce the total violations in the BSS in Oslo by as much as 250 violations per day, or more than 90,000 violations per year.

8.6 Final Remarks

In this chapter, the DBRP-model was tested in a simulation and compared to six different ROTs. Further, the ROTs have been exhaustively tested through a large number of simulations. Even though the number of simulations is too small to draw a conclusion, the DBRP-model seems to perform well compared to the ROTs. It is important to note that the DBRP-model is expected to yield better results in the simulations if $\beta = 0$, i.e. only considering violations in the objective function. Re-optimizing the DBRP-model when the service vehicle arrives at a station in order to determine the loading quantity, should further improve the model's performance in the simulations.

When simulating a larger system to compare the ROTs, we see that four of the methods result in approximately the same number of violations in the system. It may however be argued that the Num.viol. method. is slightly preferable to the others. Further testing, for instance by simulating different areas and different planning horizons could unveil greater differences between the ROTs.

Using the results from this chapter, it is possible to combine the desired elements from the different ROTs and create a smarter and more detailed ROT. In addition, one could implement more intelligent strategies for only considering the stations that are not balanced, and test other alternatives than to square or not square the different measures. Creating a smarter ROT would require additional simulations to adjust the parameters and verify its performance.

CHAPTER 9

CONCLUSION

There are currently 1,029 BSSs around the world, and about 320 more are under planning or construction. Using optimization for the operation of these systems is getting increased attention. This thesis thoroughly gathers the research on the operational level, i.e. both the SBRP and the DBRP, in an exhaustive literature survey, where the studies are systematized and compared. As far as we know, a survey of this scope on the operational level-literature is not conducted before. Most of the studies introduce a mathematical model to solve the problems, but often is realism not prioritized or the models are incomplete. We therefore present new detailed mathematical formulations for both the SBRP and the DBRP.

Trough a comparison with the literature (Table 1 in Appendix B), it is evident that the SBRP-model presented here makes fewer assumptions and allows more possibilities than most existing models. Our SBRP-model allows a heterogeneous fleet, multiple visits to each station, and non-perfect rebalancing. The DBRP-model also takes more real-life aspects into account than most of the models in the literature (Table 9.1 at the end of this chapter). In addition to the aspects mentioned for the SBRP-model, it stands out by giving a complete driving route, specific loading instructions, and by taking the timing of events into account. However, it does not consider the stochasticity in demand, but uses a deterministic continuous demand.

The demand for bikes and locks are fluctuating, and a thorough understanding of the demand is essential for rational re-balancing. When forecasting the demand, it is important to consider the geographical location of the station, the day of week, the time of day, and the weather, as our analyses show that the system usage is greatly dependent on these parameters. Using historical data from representative days, we could get an expectation and a variance for future demand. With this information, we show how to find the number of violations in the system, and the probability of violations at each station. The optimal state is an important input in both the SBRP- and the DBRP-model, and we suggest that the optimal state is set to the point where the probability of violations caused the station being full equals the probability of violations caused by the station being empty.

Various formulations and constraints are examined for the SBRP-model. For subtour elimination, the MTZ-formulation (Miller et al., 1960) with strengthening proposed by Desrochers and Laporte (1991), yields the best results. Three valid inequalities and three symmetry breaking constraints are introduced and evaluated. All the valid inequalities reduce the computational time, and the combination of all of them is very effective. The symmetry breaking constraints handling visit sequence, are shown to work very well in combination with the constraints for differentiating the vehicles by time usage. There are various ways for reducing the number of binary and integer variables in the problem. Enforcing variable reduction reduces the average computational time as much as 98.5 %. We show that the computational time depends greatly on the time limit that is used, and peaks when the time limit is set so that the total deviation is slightly above zero. When the total deviation is close to zero, the solution space is large and it is difficult to find good lower bounds.

Most of the results from the computational study for the SBRP-model apply for the DBRP-model as well. Examining different valid inequalities for the DBRP-model, we conclude that constraints (5.73-5.77) should be enforced. As the system is dynamic, the DBRP-model should be re-solved after every station visit. Hence, the most important part of the DBRP is to decide which station to visit first. Studying the stability of the solution with different time limits, it is clear that the first station visit in the solution is dependent on the chosen time limit. For some instances, the solutions are quite stable, while for others every new time limit for re-balancing results in a new solution. It is therefore difficult to conclude which time limit should be used when solving the DBRP-model. The weighting between deviations and violations in the objective function is examined, and we argue that it should depend on the planning horizon. Our analysis shows that a weight-parameter, β, between

0.15 and 0.40 yields the best results, but the weight on the deviations should be reduced as the time limit increases.

Both the number of stations and the maximum number of possible visits are crucial to the computational time, as they make the problem size increase significantly. Adding an extra possible visit at a station gives almost the same increase in problem complexity as adding a new station. The SBRP-model and the DBRP-model can only be solved to optimality for about 15 and eight stations, respectively. This is natural, as realism is prioritized, and heuristics and decomposition are not used. As the models presented here are only able to solve quite small instances, simple ROTs are very useful in practice. We introduce six different ROTs and perform numerous simulations in Excel to compare them with the DBRP-model and with each other. The DBRP-model is shown to give great results and could probably be customized to perform even better in the simulation setting. Several of the ROTs perform well, but the Exp.viol. method seems to yield the best results overall.

The main contributions of this thesis are an extensive literature survey on optimization of BSS operation and new mathematical models for the SBRP and the DBRP. The models are more detailed and captures more real-life aspects than most existing models. In addition, ROTs for re-balancing of BSSs are developed, evaluated, and compared.

Table 9.1: Characteristics of the DBRP-model. Could be seen in relation to the corresponding Table 2.5.

Parameter	**DBRP-model**
Preprocessed / Online	Preprocessed / Online
Modeling demand	Expected value
Modeling time	Continuous
Re-balancing	Continuous time
Objective function	Min. weight of violations and deviations
Complete routing	Yes
Multiple vehicles	Yes
Multiple visits to a station	Yes
Modeling handling	Yes
Selective pickup/delivery	Yes
Solution algorithm	Exact
Size of solvable instances	8 stations, 2 vehicles
Contribution	Realistic, complete model

CHAPTER 10

FURTHER WORK

The SBRP- and DBRP-models presented in this thesis are able to find optimal solutions within reasonable time for instances with up to about 15 and eight stations, respectively. Knowing that the BSS in Oslo has 100 stations (300 stations from 2017), it is obvious that the utility of these models is limited. In this chapter we present possible extensions for further work.

10.1 Solve Larger Instances

Further development of the SBRP- and DBRP-models could focus on solution methods making it possible to solve larger instances. This may be challenging, as every added station visit makes the problem bigger and more complex, as discussed in Section 6.4.

To solve larger instances using the mathematical models, there are three main alternatives. Firstly, the models could be simplified. This can be done for the SBRP-model by relaxing the time restriction and assume perfect re-balancing. Both models could be simplified by assuming only one vehicle and/or that the vehicle(s) can only visit each station once. Also, the vehicles could be restricted to always drive to one of the n stations nearest to its current location, which could result in a huge reduction in the number of arc-variables.

Instead of, or in addition to simplifying the problem, heuristics could be developed to find primal and dual bounds. Heuristic solution methods do not guarantee that the optimal solution is found, but a good heuristic could yield satisfactory solutions with little computational effort. From the literature, it seems like both tabu search (Chemla et al. (2013) and Ho and Szeto (2014)) and LNS/VNS (Rainer-Harbach et al. (2013b) and Gaspero et al. (2015)) would work well for the SBRP-model. Some form of neighborhood search could also be implemented for the DBRP-model, as illustrated by Vogel et al. (2014), Kloimüllner et al. (2014), and Brinkmann et al. (2015a). An alternative heuristic approach for both the SBRP- and the DBRP-model is to use a clustering algorithm. By grouping stations using a clustering heuristic, the routing problem for each cluster may be allocated to one vehicle and solved to optimality with the current models. Each cluster must then have less than 15 stations for overnight re-balancing and less than eight stations for through-the-day re-balancing. This method is called cluster first route second, and is implemented for the SBRP by Schuijbroek et al. (2013) and Forma et al. (2015).

Using decomposition is a third method to solve larger instances. Decomposition may be an effective way of finding the optimal solution. Looking at the literature on SBRP-models, Benders' decomposition have been used with convincing results (Erdoğan et al. (2014) and (2015)). The DBRP-model is commonly decomposed into problem specific sub-problems, e.g. one problem for determining routes and another for determining loading quantities (Angeloudis et al., 2014) or one problem for prioritizing the need for re-balancing and another for determining the routes (Regue and Recker, 2014).

10.2 Improve the Rules of Thumb

In Section 8.2 different ROTs for the DBRP are tested and compared. There is a need for such ROTs when solutions to large problems are wanted fast. Using the results from Section 8.5 is it possible to create a smarter ROT that combines elements from the other ROTs and in a better way make use of both system data and demand forecasts. Creating a smarter ROT also requires additional simulations to adjust the parameters and verify its performance.

10.3 Determine the Optimal State

A weakness of the SBRP- and the DBRP-models, is that the optimal state is considered known for all stations. This is a simplification, as the optimal state is a function of many stochastic parameters and is normally unknown. Determining the optimal number of bikes at each station at a given time is a problem at the tactical level. This problem may either be solved separately or together with the repositioning problem. One option is to further develop the ideas presented in Section 3.2.4 to create a mathematical model that calculates the optimal state at each station using the probability of violations. As the decisions on the tactical and the operational levels are closely related, it could be meaningful to study them together. By doing this, the optimal state becomes a variable in the model, determined by, among other things, the season, day of week, demand profile and the time of day.

10.4 Create a Model for Demand Forecasting

Understanding the demand is essential to be able to make good decisions during system operation. A reliable demand forecast for all stations is needed to determine the optimal state, as an direct input in the DBRP-model, and to calculate the next station visits using ROTs. Section 3.1 highlights many of the aspects that must be considered in a demand forecast, but a complete forecasting model is yet to be developed. A forecasting model should be based on historical data and adjust for the day, the time and the weather. During peak hours many stations are either full or empty. A main challenge using historical data is therefore to make good estimates for demand in periods when the stations are either full or empty.

APPENDIX A

ACRONYMS

1-PDTSP One-Commodity Pickup and Delivery Traveling Salesman Problem

B-&-B Branch-and-Bound

B-&-C Branch-and-Cut

BRSP Bike Request Scheduling Problem

BSS Bike Sharing System

DBRP Dynamic Bicycle Repositioning Problem

GIS Geographical Information Systems

GRASP Greedy Randomized Adaptive Search Procedure

IP Integer Program

LNS Large Neighborhood Search

LP Linear Program

LTR Long-Term Relocation

MIP Mixed Integer Program

MSF MakeSpan Formulation

MTZ Miller-Tucker-Zemlin (Set of constraints used to eliminate subtours)

OR Operational Research

PDP Pickup and Delivery Problem

PILOT Preferred Iterative LOok ahead Technique

ROT Rule of Thumb

SBRP Static Bicycle Repositioning Problem

STR Short-Term Relocation

TSP Travelling Salesman Problem

UDF User Dissatisfaction Function

VBA Visual Basic for Applications

VND Variable Neighborhood Decent

VNS Variable Neighborhood Search

APPENDIX B

ARTICLE SUBMITTED TO THE 7TH INTERNATIONAL CONFERENCE ON COMPUTATIONAL LOGISTICS (ICCL)

The Static Bicycle Repositioning Problem - Literature Survey and New Formulation

Hans Martin Espegren, Johannes Kristianslund, Henrik Andersson, and Kjetil Fagerholt

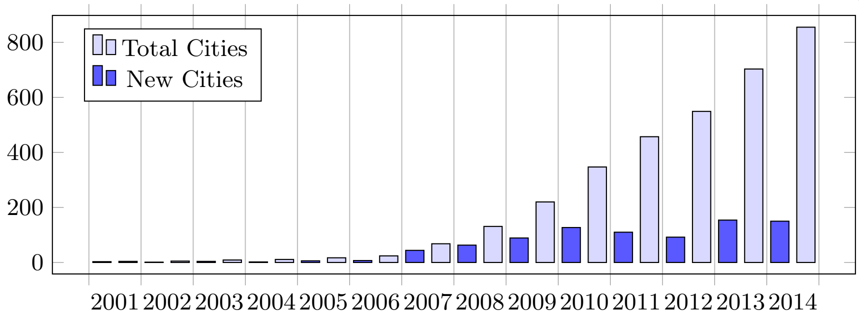
Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Trondheim, Norway

Abstract. This paper considers the static bicycle repositioning problem (SBRP), which deals with optimally re-balancing bike sharing systems (BSS) overnight, i.e. using service vehicles to move bikes from (nearly) full stations to (nearly) empty stations. An exhaustive literature survey comparing existing models is presented, and a new and improved mathematical formulation for the SBRP is proposed. The model is tested on a number of instances generated based on data from a real BSS.

1 Introduction

As urbanization proceeds throughout the world, public decision makers are looking for effective, affordable, and environmentally friendly means of transportation. Bike sharing fulfills these criteria for short distance traveling within city centres, and consequently bike sharing is getting increased attention from both governments and the public. Currently there are 948 cities with an active Bike Sharing System (BSS) and 273 with a system under planning or construction [10]. Figure 1 shows the expansion of bike sharing the recent years, expressed as number of cities in the world with a public BSS. For an extensive review of the historical development of BSSs, the reader is referred to [9], [34], and [23].

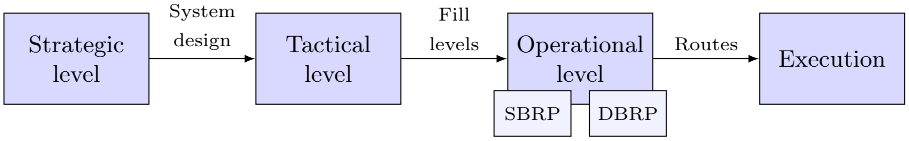
Fig. 1. Worldwide development in number of cities with a public BSS, 2000-2014 [10]



Bike sharing is a public system for automatic or semi-automatic lending of bicycles for use within a restricted time period and area. A bike can be lent at one station and delivered at another. Note that during the night most systems are either closed or in limited use. For the system to function well, it is crucial that there are bikes available at a station when someone wants to pick up a bike and that there are free slots available when someone wants to return one. To achieve this, most BSSs use service vehicles to re-balance the system, i.e. to move bikes from (nearly) full stations to (nearly) empty stations. This paper studies one important aspect of the operation of BSSs, namely the logistics of the service vehicles used to re-balance the system overnight.

The planning problems arising from BSSs are divided into three levels in accordance with [36]; a strategic, a tactical, and an operational level, as illustrated in Figure 2. The strategic level contains problems that arise when designing the system, e.g. determining the optimal number of bikes and locations of stations. On the tactical level the objective is to find an optimal distribution of bikes between the stations at a specific time, while finding optimal routes for the service vehicles to re-balance the system is the objective at the operational level.

Fig. 2. Planning levels of BSS optimization



It is common to divide the operational level in two: static and dynamic problems. In line with [30], the problems are named *static bicycle repositioning problem* (SBRP) and *dynamic bicycle repositioning problem* (DBRP). The SBRP is typically used for overnight balancing, when the demand forecast for the operating period is not considered; the problem is static and deterministic. To describe the SBRP we introduce the concept of states, i.e. a distribution of bikes throughout the system, expressed as a specific number of bikes at each station. The *optimal state* is the desired distribution of bikes at the end of the planning period, i.e. early in the morning, while the *initial state* is the distribution at the beginning of the planning period, i.e. late in the evening. After solving the model, we get the *final state*. The difference between the final state and optimal state is called deviation. All stations and vehicles have restricted capacities, and the fleet of service vehicles may be either homogeneous or heterogeneous. For every vehicle, a complete route and the number of bikes to pick up or deliver at each station must be decided. Hence the SBRP can be classified as a *static many-to-many one-commodity pickup and delivery problem with selective pickups and selective deliveries*, in accordance with [3]. The DBRP is on the other hand used for intraday re-balancing, as the demand during the operating time is taken into account. Hence, the DBRP is both dynamic and stochastic.

In this paper we focus on the SBRP. Our contributions are 1) to present an exhaustive literature survey on the SBRP, including a comparison of the existing models, and 2) to propose a new mathematical model of the problem that captures more real-life aspects. We also propose symmetry-breaking constraints and valid inequalities to tighten the formulation. The model is tested on a number of test instances based on data from a real BSS.

Section 2 provides the literature survey on the SBRP, while a new mathematical model for the SBRP is introduced in Section 3. A computational study is presented in Section 4 and concluding remarks are given in Section 5.

2 Literature Survey

In this literature survey we focus on the static bicycle repositioning problem (SBRP). For studies on the strategic level, we refer to [16], [22], and [32] that determine the number of stations and their locations, and to [15] that finds the optimal number of bikes in the system and the number of slots at each station. At the tactical level we can refer to [29], [33], and [37] for analyses of the placement of bikes, while [20] studies the detection of broken bikes in the system. There are also a number of studies regarding the DBRP, see for example [1], [4], [5], [7], [21], [25], [26], [31], and [38].

The SBRP was first studied in [2]. They describe the system using graph-theory. The objective is to move bikes along the arcs so each station is perfectly re-balanced at minimal cost. One of the main findings is that the SBRP is NP-hard. In [6], the work from [2] is continued. An optimization model is presented, but shows to be hard to solve, so they relax the problem by removing the sequential dimension and solve it using a branch-and-cut (B&C) algorithm.

In [30], two different mixed integer programming formulations are introduced; an arc-indexed and a time-indexed. The objective is to minimize a weighted sum of the stations' penalty costs for deviations and the operating cost. The authors conclude that the arc-indexed model provides the best results for most instances, but the time-indexed formulation is easier to adapt to the DBRP. Valid inequalities and dominance rules are proposed to strengthen the formulations.

The arc-indexed formulation from [30] is enhanced in [19] and [14], both proposing methods for solving larger instances. In [19], the formulation is simplified by allowing only one vehicle, stating that a station is either a pickup or delivery station and assuming that each station only can be visited once. The objective is to minimize a penalty function depending on the number of bikes at each station. The authors present a construction heuristic used to generate an initial solution followed by a tabu search. On the other hand, the model is expanded in [14] by using a three-step algorithm. In the first step, stations are clustered using a saving heuristic. In the second step, vehicles are assigned to clusters, while the routes for each vehicle are determined in the third step.

The SBRP is represented using a complete directed graph in [27] and [28]. Further, several metaheuristics are presented and tested. The authors conclude that Variable Neighborhood Search (VNS) yields the best results on instances of moderate size, while a PILOT/GRASP hybrid turns out to be superior on large instances. A neighborhood search is also used in [17]. Two formulations for the SBRP are also developed; a routing model and a step model, both incorporated in a Large Neighborhood Search (LNS). The routing model uses an arc-indexed formulation, while the step model allocates all station visits to routes.

In [33], the SBRP is solved in combination with the tactical level problem of finding the optimal states. The routes from the SBRP must satisfy the service level requirements from an inventory problem. The objective is to minimize the maximal route length, hence it is formulated as a makespan problem. To solve the model the authors propose a cluster first route second heuristic.

Four possible formulations of the SBRP are tested and discussed in [8]. To handle the exponential number of subtour eliminating constraints, a B&C algorithm is proposed in addition to both valid inequalities and separation procedures. The authors conclude that the subtour elimination and separation techniques proposed by [18] for the 1-PDTSP give the best computational results.

A decomposition method is introduced in [35], consisting of a request generation algorithm and a *bike request scheduling problem* (BRSP). The request generation algorithm uses various data to generate repositioning requests. A request includes the location and number of bikes to be picked up or delivered, a time window and an importance weight. The BRSP determines which requests to execute and assigns them to vehicles. The objective is to minimize the total weight of rejected requests.

The objective of the SBRP-model in [26] is to maximize the number of rebalanced stations, only allowing pickup and delivery of full truckloads of bikes. The authors use a heuristic that solves the one-vehicle problem for each vehicle.

In [13], the SBRP is decomposed using a Benders decomposition scheme. The subproblem determines the pickup and delivery quantities along a fixed route of station visits, while the master problem finds new routes visiting all stations with too few or too many bikes. In a later study, [12], the authors use insights from [13] to solve the SBRP formulation from [6]. Whilst [6] could only find heuristic solutions for realistically sized instances, the method from [12] yield optimal solutions.

Table 1 shows a comparison of the main characteristics of the SBRP models in the studies surveyed above, as well as some key information about the solution methods. Note that the mathematical model proposed in Section 3 is also included in the table. The numbers in the top row correspond to the numbers in Table 2.

From the table it becomes evident that half of the studies solve the problem with only one service vehicle, even though most problems of realistic size use several. Note that many articles use clustering algorithms. By assigning each cluster to a vehicle, the SBRP could be solved once for each vehicle. Among the studies allowing multiple vehicles, two assume the fleet to be homogeneous. Half of the studies allow multiple visits to a station, while the other half does not. When the deviation between the optimal and initial state is larger than the vehicle capacity, allowing multiple visits to each station seems most reasonable.

Table 1. Comparison of SBRP Articles. The numbers in the table header are explained in Table 2.

Parameter	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Multiple vehicles	No	No	Yes	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Yes	Yes
Heterogeneous fleet	n/a	n/a	Yes	Yes	Yes	n/a	n/a	No	Yes	n/a	n/a	No	Yes	Yes
One vehicle can visit a station multiple times	Yes	Yes	No/Yes two mod.	Yes	Yes	No	No	No	Yes	No	Yes	No	No	Yes
Several vehicles can visit the same station	n/a	n/a	Yes	Yes	Yes	n/a	n/a	No	Yes	n/a	n/a	No	No	Yes
Modeling loading/unloading	No cost for l/unl.	No cost for l/unl.	Depends on quantity	Average time for station	No cost for l/unl.	Depends on quantity	No cost for l/unl.	Time for each request	Average time for station	Depends on quantity	No cost for l/unl.	Average time for station	Depends on quantity	Depends on quantity
Selective pickup/delivery	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Stations used as temporal inventories	No	Yes	Yes	No	Yes	No	No	No	Yes	No	Yes	No	Yes	No
Allows non-perfect rebalancing	No	No	Yes	Yes	No	Yes	No	Yes	Yes	No	No	Yes	Yes	Yes
Objective function	Min. cost	Min. cost	Min. dev. and cost	Min. dev & operations	Min. longest route	Min. penalty func.	Min. cost	Min. rejected requests	Min. dev. and cost	Min. time and cost	Min. cost	Max. rebal. nodes	Min. dev. and cost	Min. dev. and time
Subtour elimination	Many new var. & cons.	Many new var. & cons.	MTZ / Time	Heuristics	Time	MTZ	MTZ / Sep. & cut	Time	Time / Route	Separation & cut	Separation & cut	A variant of MTZ	MTZ	Strengthened MTZ
Solution method	n/a	B&C & Tabu search	Heuristics & exact	PILOT, GRASP, VNS	Cluster first route second	Tabu search	Exact: B&C	n/a	LNS	Exact: Benders decomp.	Exact: Benders cut	Heuristics & exact	Cluster first route second	Exact
Size of solvable instances	n/a	1v., 100st.	1v., 60st.	21v., 700st.	5v., 135st.	1v., 400st.	1v., 50st.	n/a	6v., 240st.	1v., 50st.	1v., 60st.	5v., 100st.	3v., 200st.	2v., 15st.
Based on article (table header as reference)	n/a	1	n/a	n/a	n/a	3	n/a	n/a	4	n/a	2 & 10	n/a	3	n/a

Table 2. Articles overview for Table 1

1	Benchimol et al. [2]	**8**	Sörensen and Dilip [35]
2	Chemla et al. [6]	**9**	Gaspero et al. [17]
3	Raviv et al. [30]	**10**	Erdoğan et al. [13]
4	Rainer-Harbach et al. [27] & [28]	**11**	Erdoğan et al. [12]
5	Schuijbroek et al. [33]	**12**	O'Mahony and Shmoys [26]
6	Ho and Szeto [19]	**13**	Forma et al. [14]
7	Dell'Amico et al. [8]	**14**	Espegren et al. (this study)

Five studies assume that there is no time usage or cost associated with the loading and unloading operations at the stations, three use an average time and five studies let the time usage depend on the number of bikes handled. Note that none of the studies take traffic congestion into account, but presume the driving time between two stations to be constant. Just one study, [26], allows only full truckloads.

The studies by [2], [6], [33], [8], [13], and [12] minimize the time and/or cost associated with repositioning the bikes. In these studies, the solutions are only valid if the number of deviations is zero, i.e. the system is perfectly re-balanced. The remaining studies use objective functions that in various ways minimize the number of deviations.

All but two studies ([2] and [35]) include computational experiments on either theoretical or real instances. The majority use some kind of heuristics to solve the instances. All studies that use exact methods fail to find the optimal solution when the problem size increases and only yield upper and lower bounds. Since the problems include binary and/or integer variables, a common approach is to use B&C algorithms. The cuts can be generated using inequalities from [18] or using Benders decomposition [12]. Popular heuristics are tabu search and VNS/LNS. In [14] the problem is decomposed, and one part is solved by a heuristic and another part using exact methods.

The studies using a time-variable do not need subtour eliminating constraints. Among the remaining articles, the MTZ-formulation [24] is widely used to avoid subtours, while three studies, [8], [13], and [12], eliminate subtours using separation algorithms and cuts.

3 Mathematical Formulation

In this section, we propose a new mathematical model for the SBRP. The objective of the model is to minimize a weighted combination of total deviation and the time used. We assume a heterogeneous fleet of service vehicles that start and finish their routes empty at the depot. Several vehicles can visit the same station and a single vehicle can visit the same station several times. We presume the driving time between stations to be constant and independent of the hour. In addition to the driving time, each vehicle uses a fixed parking time at each station visit. Time used to load and unload bikes at a station is proportional to the number of bikes handled plus a given parking time. All stations are defined as either *pickup stations* or *delivery stations* depending on their initial state relative to their optimal state. It is not possible to pick up bicycles at a delivery station or deliver them at pickup station.

Each station $i \in \mathcal{N}$ has a set of possible visits $\mathcal{M}_i$. Note that the depot is included in this set. Our formulation uses arc flow variables x_{imjnv}, $i \in \mathcal{N}$, $m \in \mathcal{M}_i$, $j \in \mathcal{N}$, $n \in \mathcal{M}_j$, $v \in \mathcal{V}$ indicating whether vehicle v drives from station visit (i,m) to station visit (j,n) or not, where m and n are the station visit numbers. The entire notation is presented in Table 3.

Table 3. Notation used in the mathematical formulation

Sets	
$\mathcal{N}$	Set of stations, indexed by i, j
$\mathcal{V}$	Set of vehicles, indexed by v
$\mathcal{M}_i$	Set of possible visits at station i, indexed by m, n
Parameters	
T^D_{ij}	Driving time between stations i and j
T^P	Time used for parking a vehicle
T^H	Handling time used for loading or unloading a bike
$\overline{T}$	Time limit for operation of service vehicles
Q_v	Capacity of vehicle v
J_i	1 if station i is a pickup station, and -1 if it is a delivery station
α	Weight on deviations in the objective function relative to time usage
$\overline{A}$	Maximum number of station visits for a vehicle
I_i	Initial state, number of bikes at station i
O_i	Optimal state, number of bikes at station i
Variables	
x_{imjnv}	1 if vehicle v is driving directly from station visit $(i,\ m)$ to station visit $(j,\ n)$, 0 otherwise
f_{ijv}	Total number of bikes carried by vehicle v between stations i and j
q_{iv}	Number of bikes either picked up or delivered at station i by vehicle v
y_i	Final state, number of bikes at station i
u_{imv}	The sequence number in which station visit (i, m) is made by vehicle v

$$\begin{aligned} \min \quad & \alpha \sum_{i \in \mathcal{N}} J_i (y_i - O_i) \\ & +(1-\alpha) \left[\sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}_i} \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \sum_{v \in \mathcal{V}} \left(T^D_{ij} + T^P \right) x_{imjnv} + \sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{V}} T^H q_{iv} \right] \end{aligned} \tag{1}$$

subject to:

$$\sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} x_{dvjnv} = 1 \qquad v \in \mathcal{V} \tag{2}$$

$$\sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}_i} x_{imd(v+|\mathcal{V}|)v} = 1 \qquad v\in\mathcal{V} \tag{3}$$

$$\sum_{j\in\mathcal{N}}\sum_{n\in\mathcal{M}_j} x_{jnimv} - \sum_{j\in\mathcal{N}}\sum_{n\in\mathcal{M}_j} x_{imjnv} = 0 \qquad i\in\mathcal{N}\setminus\{d\}, m\in\mathcal{M}_i, v\in\mathcal{V} \tag{4}$$

$$\sum_{j\in\mathcal{N}}\sum_{n\in\mathcal{M}_j}\sum_{v\in\mathcal{V}} x_{imjnv} \leq 1 \qquad i\in\mathcal{N}, m\in\mathcal{M}_i \tag{5}$$

$$\sum_{j\in\mathcal{N}} f_{jiv} + J_i q_{iv} - \sum_{j\in\mathcal{N}} f_{ijv} = 0 \qquad i\in\mathcal{N}, v\in\mathcal{V} \tag{6}$$

$$y_i + \sum_{v\in\mathcal{V}} J_i q_{iv} = I_i \qquad i\in\mathcal{N} \tag{7}$$

$$\sum_{v\in\mathcal{V}} q_{iv} - J_i(I_i - O_i) \leq 0 \qquad i\in\mathcal{N} \tag{8}$$

$$f_{ijv} - \sum_{m\in\mathcal{M}_i}\sum_{n\in\mathcal{M}_j} Q_v x_{imjnv} \leq 0 \qquad i,j\in\mathcal{N}, v\in\mathcal{V} \tag{9}$$

$$\sum_{j\in\mathcal{N}} f_{djv} = 0 \qquad v\in\mathcal{V} \tag{10}$$

$$\sum_{i\in\mathcal{N}} f_{idv} = 0 \qquad v\in\mathcal{V} \tag{11}$$

$$\sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}_i}\sum_{j\in\mathcal{N}}\sum_{n\in\mathcal{M}_j} (T^D_{ij} + T^P) x_{imjnv} + \sum_{i\in\mathcal{N}} T^H q_{iv} \leq \overline{T} \qquad v\in\mathcal{V} \tag{12}$$

$$u_{imv} - u_{jnv} + (\overline{A}-1)x_{imjnv} + (\overline{A}-3)x_{jnimv} \leq \overline{A} - 2 \qquad i,j\in\mathcal{N}, m\in\mathcal{M}_i, n\in\mathcal{M}_j, v\in\mathcal{V} \tag{13}$$

$$x_{imjnv} \in \{0,1\} \qquad i,j\in\mathcal{N}, m\in\mathcal{M}_i, n\in\mathcal{M}_j, v\in\mathcal{V} \tag{14}$$

$$f_{ijv} \geq 0, \text{integer} \qquad i,j\in\mathcal{N}, v\in\mathcal{V} \tag{15}$$

$$q_{iv} \geq 0, \text{integer} \qquad i\in\mathcal{N}, v\in\mathcal{V} \tag{16}$$

$$y_i \geq 0, \text{integer} \qquad i\in\mathcal{N} \tag{17}$$

$$u_{imv} \geq 0, \text{integer} \qquad i\in\mathcal{N}, m\in\mathcal{M}_i, v\in\mathcal{V} \tag{18}$$

The objective function (1) consists of two terms that are to be minimized. The first term is the deviation in number of bikes between the final state, y_i, and the optimal state, O_i, for all stations. Having too many and too few bikes are equally penalized. The second term is the total time used to obtain the final state. Total time corresponds to the sum of driving time, T^D_{ij}, parking time, T^P, and handling time, T^H. By setting α slightly below one, the most effective routes minimizing the deviation are found.

Constraints (2) and (3) force the vehicles to start and end at the depot, d. Symmetry at the depot is handled by stating that vehicle v uses visit numbers v and $v + |\mathcal{V}|$ when leaving and arriving at the depot, respectively. Constraints (4) ensure that a vehicle that enters a station visit, leaves the same station visit, while constraints (5) make sure all station visits happen at most once.

The loading and unloading constraints (6) ensure that the flow of bikes into station i, f_{jiv}, equals the flow out of the station, f_{ijv}, plus the net pickup, q_{iv}. Since the problem is static, only the total net pickup is considered. Constraints (7) and (8) assign values to the final state, y_i. In addition, constraints (8) give an upper bound on the net pickup at station i by vehicle v, q_{iv}.

The vehicle capacity constraints (9) make sure that a vehicle never carries more bikes along an arc than the vehicle's capacity multiplied by the number of times the arc is traversed. Constraints (10) and (11) state that the service vehicles must be empty when leaving and returning to the depot. Capacity constraints for the stations are handled implicitly. The total time spent for each vehicle is limited to $\overline{T}$ by constraints (12).

Subtours are handled in constraints (13), similar to the Miller-Tucker-Zemlin (MTZ) constraints [24], but with a strengthening proposed in [11]. Various methods for eliminating subtours have been tested, and these constraints showed to perform best.

Symmetry breaking constraints remove solutions that are mathematically different, but practically identical, while adding valid inequalities is a way of improving the solution of the linear relaxation. Various symmetry breaking constraints and valid inequalities have been tested, and the ones presented here are those found most effective.

$$\sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \sum_{v \in \mathcal{V}} \left(x_{imjnv} - x_{i(m-1)jnv} \right) \leq 0 \qquad i \in \mathcal{N} \setminus \{d\}, m \in \mathcal{M}_i \setminus \{1\} \tag{19}$$

$$\begin{aligned} &\sum_{i \in \mathcal{N}} \sum_{m \in \mathcal{M}_i} \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \left(T^D_{ij} + T^P \right) \left(x_{imjnv} - x_{imjn(v+1)} \right) \\ &\quad + \sum_{i \in \mathcal{N}} T^H \left(q_{iv} - q_{i(v+1)v} \right) \geq 0 \qquad v \in \mathcal{V} \setminus \{|\mathcal{V}|\} \;\Big|\; Q_v = Q_{(v+1)} \end{aligned} \tag{20}$$

Constraints (19) reduce symmetry by handling the station visits, so that they appear in the right sequence. By introducing constraints (20), symmetry that occurs when using a homogeneous fleet of service vehicles is reduced.

$$\sum_{v \in \mathcal{V}} q_{iv} - |\,(I_i - O_i)\,| \sum_{m \in \mathcal{M}_i} \sum_{j \in \mathcal{N}} \sum_{n \in \mathcal{M}_j} \sum_{v \in \mathcal{V}} x_{imjnv} \leq 0 \qquad i \in \mathcal{N} \tag{21}$$

$$\sum_{v \in \mathcal{V}} \sum_{m \in \mathcal{M}_i} \sum_{n \in \mathcal{M}_j} x_{imjnv} + \sum_{v \in \mathcal{V}} \sum_{m \in \mathcal{M}_i} \sum_{n \in \mathcal{M}_j} x_{jnimv} \leq 1 \quad i, j \in \mathcal{N} \;\Big|\; J_i = J_j \tag{22}$$

Constraints (21) force the x_{imjnv}-variables to take values closer to one or zero in the linear relaxation. For instance, for a station to be perfectly rebalanced, the sum over the x_{imjnv}-variables associated with that station must equal one. In [6] it is shown that the arcs between two stations of similar type need not be traversed more than once, resulting in constraints (22).

Table 1 includes a comparison of this mathematical model with the models in previous studies.

4 Computational Study

The mathematical model presented in Section 3 has been implemented in Xpress-IVE 1.24.06 using the Mosel programming language. The computational experiments have been executed on a computer with Intel Core i7-3770 3.40 GHz processor, 16 GB of RAM and running Windows 7.

4.1 Test Instances

Based on the BSS in Oslo, Norway, six test areas (geographical regions) have been identified. Details about the areas can be found in Table 4. The areas have an estimated optimal state for each station and a driving time matrix, T^D_{ij}. A parking time, T^P, set to one minute, is added for each station, while the handling time for each bike, T^H, is set to 30 seconds. All areas have two service vehicles. For each area, three instances are created by varying the initial states, while all other parameters are unchanged. Note that we assume perfect re-balancing for the third instance in each area, making the instances easier to solve because of a simpler structure.

Table 4. Test areas

Area	$\lvert\mathcal{N}\rvert$	**Avg. driving time**	$\overline{T}$	$\lvert\mathcal{V}\rvert$	**Cap.** $v = 1$	**Cap.** $v = 2$
1	6	2 min	16 min	2	10	10
2	8	6 min	30 min	2	10	15
3	10	6 min	40 min	2	12	12
4	12	5 min	30 min	2	10	10
5	14	7 min	45 min	2	12	12

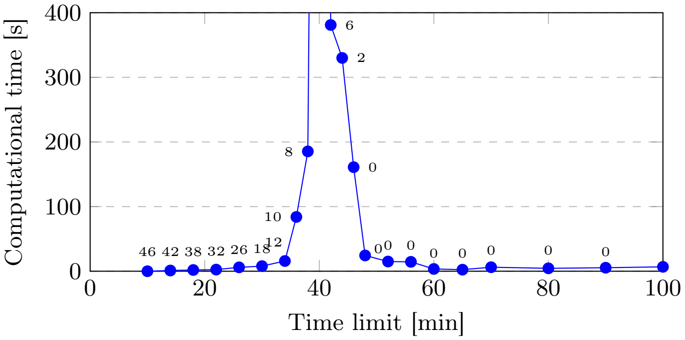
4.2 Computational Results

Various parameters in the model affect the computational time; the time limit, $\overline{T}$, the number of stations, $|\mathcal{N}|$, the maximum possible number of visits to each station, $|\mathcal{M}_i|$, and the number of service vehicles, $|\mathcal{V}|$. Among these, the time limit and the maximum possible number of visits are studied here.

Figure 3 shows that the computational time peaks when the time limit is set so that the total deviation is slightly above zero. By only changing the time limit, the computational time varies from less than one second to more than 35 minutes. The same pattern is seen for all instances.

The use of station visit numbers, $m, n \in \mathcal{M}_i$, is a new approach for the SBRP, allowing multiple station visits without a time-index. Though this formulation has some advantages, both the solution and the computational time depends on the value of $|\mathcal{M}_i|$, i.e. the maximum possible number of visits to each station. Each possible station visit (i,m) could be considered a distinct node in the graph.

Fig. 3. The computational time depicted for different time limits, $\overline{T}$, for instance 4.1, i.e. the first instance from area 4. The numbers beside the markers indicate the total deviation between the initial and optimal state in the solution.



Hence, adding one element to the set $\mathcal{M}_i$ for one station i, is equivalent to adding a node to the graph.

Consequences of using different values for $|\mathcal{M}_i|$ is illustrated in Table 5. The *lower bound method* is the smallest number of visits to each station to allow perfect re-balancing, defined as: $|\mathcal{M}_i| = \left\lceil \frac{|I_i - O_i|}{\min_{v \in \mathcal{V}} C_v^{\mathcal{V}}} \right\rceil$. The *lower bound +1 method* allows one more visit to each station than the *lower bound method*. The *upper bound method* is defined as $|\mathcal{M}_i| = |I_i - O_i|$. For all our test instances the total number of deviations at the stations were the same for every method, independent of $|\mathcal{M}_i|$, hence only improvement in driving time is recorded in the table. Consequently, the *lower bound method* is recommended as it yields near optimal solutions with much less computational effort.

Depending on the input parameters, the mathematical model from Section 3 can be solved to optimality for instances of about 15 stations. Combined with some form of clustering, this could be enough to solve many realistically sized instances.

4.3 Comparison with Rules of Thumb

Today, in the Oslo BSS, the operators utilize their experience and common sense to decide the routes and the pickup and delivery quantities. Here, two greedy rules of thumb are created to imitate the operators behavior. The first rule of thumb states that the service vehicle should visit the nearest pickup and delivery stations in sequence, unless it is able to meet the demand at two subsequent stations of the same type. The vehicle should serve the entire demand of bikes at the stations, but is restricted by its capacity and the time limit for re-balancing. The second rule of thumb works quite similar, but the vehicle always goes to the station with the largest deviation.

A comparison is made between the results obtained with these rules of thumb and the ones obtained by solving the model from Section 3. The comparison is

Table 5. Comparison of number of nodes in the graph, computational times, and quality of solution for three different methods for setting the maximum possible number of visits, $\mathcal{M}_i$. The improvement in solution is relative to the *lower bound method*. Note that the deviation between the initial and optimal state is equal for all methods, hence improvement in solution only refers to driving time.

	Lower bound		**Lower bound +1**			**Upper bound**		
Instance	$\sum_{i\in\mathcal{N}} \lvert\mathcal{M}_i\rvert$	**Comp. time**	$\sum_{i\in\mathcal{N}} \lvert\mathcal{M}_i\rvert$	**Comp. time**	**Imprv. in sol.**	$\sum_{i\in\mathcal{N}} \lvert\mathcal{M}_i\rvert$	**Comp. time**	**Imprv. in sol.**
1.1	6	0.19s	12	5.60s	0.0 %	24	139.70s	0.0 %
1.2	8	0.34s	14	1.91s	0.0 %	38	47.40s	0.0 %
1.3	8	0.20s	14	0.23s	0.0 %	48	>3000s	≥0.0 %
2.1	8	0.64s	16	697.00s	0.0 %	32	>3000s	≥0.0 %
2.2	9	0.44s	17	1.51s	0.0 %	46	>3000s	≥0.0 %
2.3	10	0.62s	18	7.81s	3.6 %	48	462.00s	3.6 %
3.1	10	1.25s	20	279.50s	0.0 %	56	>3000s	≥0.0 %
3.2	12	7.00s	22	281.00s	0.0 %	64	>3000s	≥0.0 %
3.3	12	1.25s	22	74.70s	0.0 %	58	>3000s	≥0.0 %
4.1	12	8.40s	24	>3000s	≥0.0 %	52	>3000s	≥0.0 %
4.2	15	17.00s	27	2089.00s	0.0 %	74	>3000s	≥0.0 %
4.3	12	0.40s	24	20.30s	3.3 %	62	286.50s	3.3 %
5.1	14	69.00s	28	>3000s	≥0.0 %	70	>3000s	≥0.0 %
5.2	16	15.30s	30	2708.00s	0.0 %	86	>3000s	≥0.0 %
5.3	16	1.07s	30	57.70s	7.7 %	106	>3000s	≥7.7 %
Average	n/a	8.21s	n/a	>814.95 s	≥1.6 %	n/a	>2262.00 s	≥1.6 %

only made for instances 2.1 and 3.1, and to simplify only one vehicle is used. With regard to deviations, the SBRP-model finds solutions that are between 20.0 and 56.6% better than the two rules of thumb. A characteristic for the optimal solution is that it has less slack in the time restriction than the rules of thumb.

4.4 Practical Use of the Model

Six of the 13 articles listed in Table 1 minimize time usage or cost, given that the system will be perfectly re-balanced. By assuming zero deviation, several simplifications can be made, and the computational time will decrease significantly, as indicated in Figure 3.

It is possible to utilize intervals, rather than a fixed number, to describe the optimal state. This provides more flexibility to the model, presumably making it harder to solve, but it may be more realistic. An alternative to use intervals, is to punish large deviations relatively more than small, for example by punishing the square of the deviation.

In addition to serving as a tool for operational planning, the SBRP-model could be used to support both strategic and tactical decisions. Analyzing changes in parameter values can be done by re-solving the problem for different values. By increasing the time limit for re-balancing operations, the number of deviations could go down. The operator may use this information to decide whether to expand the time limit or not. To support the decision of whether to acquire or dispose a service vehicle, the SBRP-model may be used to quantify the effect. Increased vehicle capacity leads, as expected, to a reduced objective value. At a certain point, the objective value reaches its lowest point, where the total deviation is zero or the time limit restricts the objective value from decreasing further. To compare a change in the objective value with the cost of changing a parameter, the system operator is referred to a cost–benefit analysis.

5 Concluding Remarks

As the SBRP is a relatively novel problem, a review of the research made on the topic is missing in the literature. An extensive literature survey, consisting of the review and comparison of 13 studies, has therefore been conducted. As can be seen from Table 1, many studies make assumptions that are unrealistic for most practical problems. We have proposed a new mathematical model for the SBRP that makes fewer assumptions and allows more possibilities than many existing models. For instance does this model allow a heterogeneous fleet, multiple visits to each station, and non-perfect re-balancing.

Since we have focused on the modeling and not on solution algorithms in this study, we are only able to solve relatively small instances. The model should however provide a good starting point for proposing more advanced solution methods, for instance as an important part of a clustering algorithm for solving realistically sized instances.

References

1. Angeloudis, P., Hu, J., Bell, M.G.: A strategic repositioning algorithm for bicycle-sharing schemes. Transportmetrica A: Transport Science 10(8), 759–774 (2014)
2. Benchimol, M., Benchimol, P., Chappert, B., De La Taille, A., Laroche, F., Meunier, F., Robinet, L.: Balancing the stations of a self service “bike hire” system. RAIRO-Operations Research 45(1), 37–61 (2011)
3. Berbeglia, G., Cordeau, J.F., Gribkovskaia, I., Laporte, G.: Static pickup and delivery problems: a classification scheme and survey. Top 15(1), 1–31 (2007)
4. Brinkmann, J., Ulmer, M.W., Mattfeld, D.C.: Short-term strategies for stochastic inventory routing in bike sharing systems. Transportation Research Procedia 10, 364–373 (2015)
5. Caggiani, L., Ottomanelli, M.: A modular soft computing based method for vehicles repositioning in bike-sharing systems. Procedia-Social and Behavioral Sciences 54, 675–684 (2012)
6. Chemla, D., Meunier, F., Calvo, R.W.: Bike sharing systems: Solving the static rebalancing problem. Discrete Optimization 10(2), 120–146 (2013)
7. Contardo, C., Morency, C., Rousseau, L.M.: Balancing a dynamic public bike-sharing system. Tech. Rep. CIRRELT-2012-09, Universitè de Montrèal, Montrèal,

Canada (2012), http://claudio.contardo.org/wp-content/uploads/2011/09/CIRRELT-2012-09.pdf

8. Dell'Amico, M., Hadjicostantinou, E., Iori, M., Novellani, S.: The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. Omega 45, 7–19 (2014)
9. DeMaio, P.: Bike-sharing: History, impacts, models of provision, and future. Journal of Public Transportation 12(4), 3 (2009)
10. DeMaio, P., Meddin, R.: The bike-sharing world map. Available at: www.bikesharingmap.com (2015), (Accessed: 2015-October-06)
11. Desrochers, M., Laporte, G.: Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. Operations Research Letters 10(1), 27–36 (1991)
12. Erdoğan, G., Battarra, M., Calvo, R.W.: An exact algorithm for the static rebalancing problem arising in bicycle sharing systems. European Journal of Operational Research 245(3), 667–679 (2015)
13. Erdoğan, G., Laporte, G., Calvo, R.W.: The static bicycle relocation problem with demand intervals. European Journal of Operational Research 238(2), 451–457 (2014)
14. Forma, I.A., Raviv, T., Tzur, M.: A 3-step math heuristic for the static repositioning problem in bike-sharing systems. Transportation Research Part B: Methodological 71, 230–247 (2015)
15. Fricker, C., Gast, N.: Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. EURO Journal on Transportation and Logistics 3, 1–31 (2014)
16. García-Palomares, J.C., Gutiérrez, J., Latorre, M.: Optimizing the location of stations in bike-sharing programs: a GIS approach. Applied Geography 35(1), 235–246 (2012)
17. Gaspero, L., Rendl, A., Urli, T.: Balancing bike sharing systems with constraint programming. Constraints 20, 1–31 (2015)
18. Hernández-Pérez, H., Salazar-González, J.J.: The one-commodity pickup-and-delivery traveling salesman problem: Inequalities and algorithms. Networks 50(4), 258–272 (2007)
19. Ho, S.C., Szeto, W.: Solving a static repositioning problem in bike-sharing systems using iterated tabu search. Transportation Research Part E: Logistics and Transportation Review 69, 180–198 (2014)
20. Kaspi, M., Raviv, T., Tzur, M.: Detection of unusable bicycles in bike-sharing systems. Available at: http://www.eng.tau.ac.il/~talraviv/Publications/Detection%20of%20Unusable%20Bicycles.pdf (2015), working paper. Tel-Aviv University. (Published: 2015-October, Accessed: 2015-November-08)
21. Kloimüllner, C., Papazek, P., Hu, B., Raidl, G.R.: Balancing bicycle sharing systems: an approach for the dynamic case. In: Evolutionary Computation in Combinatorial Optimisation, pp. 73–84. Springer (2014)
22. Lin, J.R., Yang, T.H.: Strategic design of public bicycle sharing systems with service level constraints. Transportation Research Part E: Logistics and Transportation Review 47(2), 284–294 (2011)
23. Midgley, P.: Bicycle-sharing schemes: enhancing sustainable mobility in urban areas. In: 19th session of the Commission on Sustainable Development. 2011-May-02. United Nations, Department of Economic and Social Affairs (05 2011), background Paper No. 8
24. Miller, C.E., Tucker, A.W., Zemlin, R.A.: Integer programming formulation of traveling salesman problems. Journal of the ACM (JACM) 7(4), 326–329 (1960)

25. Nair, R., Miller-Hooks, E., Hampshire, R.C., Bušić, A.: Large-scale vehicle sharing systems: analysis of Vélib'. International Journal of Sustainable Transportation 7(1), 85–106 (2013)
26. O'Mahony, E., Shmoys, D.B.: Data analysis and optimization for (citi) bike sharing. In: Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015-January-25. Association for the Advancement of Artificial Intelligence (01 2015)
27. Rainer-Harbach, M., Papazek, P., Hu, B., Raidl, G.: Balancing bicycle sharing systems: A variable neighborhood search approach. In: Middendorf, M., Blum, C. (eds.) Evolutionary Computation in Combinatorial Optimization, Lecture Notes in Computer Science, vol. 7832, pp. 121–132. Springer Berlin Heidelberg (2013)
28. Rainer-Harbach, M., Papazek, P., Raidl, G.R., Hu, B., Kloimüllner, C.: PILOT, GRASP, and VNS approaches for the static balancing of bicycle sharing systems. Journal of Global Optimization 63(3), 1–33 (2013)
29. Raviv, T., Kolka, O.: Optimal inventory management of a bike-sharing station. IIE Transactions 45(10), 1077–1093 (2013)
30. Raviv, T., Tzur, M., Forma, I.A.: Static repositioning in a bike-sharing system: models and solution approaches. EURO Journal on Transportation and Logistics 2(3), 187–229 (2013)
31. Regue, R., Recker, W.: Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem. Transportation Research Part E: Logistics and Transportation Review 72, 192–209 (2014)
32. Romero, J.P., Ibeas, A., Moura, J.L., Benavente, J., Alonso, B.: A simulation-optimization approach to design efficient systems of bike-sharing. Procedia-Social and Behavioral Sciences 54, 646–655 (2012)
33. Schuijbroek, J., Hampshire, R., van Hoeve, W.J.: Inventory rebalancing and vehicle routing in bike sharing systems. Available at: http://repository.cmu.edu/cgi/viewcontent.cgi?article=2490&context=tepper (2013), working paper. Tepper School of Business. (Published: 2013-February, Accessed: 2015-December-01)
34. Shaheen, S., Guzman, S., Zhang, H.: Bikesharing in Europe, the Americas, and Asia: past, present, and future. Transportation Research Record: Journal of the Transportation Research Board (2143), 159–167 (2010)
35. Sörensen, K., Dilip, D.: The (city) bike request scheduling problem–a novel approach to solve the city bike repositioning problem. In: Toklu, Y.C., Bekdas, G. (eds.) Metaheuristics and Engineering, Workshop of the EURO Working Group, vol. 15, pp. 157–161. Bilecik Şeyh Edebali University (2014)
36. Vogel, P., Ehmke, J.F., Mattfeld, D.C.: Service network design of bike sharing systems. Available at: https://www.tu-braunschweig.de/Medien-DB/winfo/publications/service_network_design_of_bike_sharing_systems.pdf (2015), working paper. Technische Unversität Braunschweig. (Published: 2015-March-24, Accessed: 2015-September-25)
37. Vogel, P., Greiser, T., Mattfeld, D.C.: Understanding bike-sharing systems using data mining: Exploring activity patterns. Procedia-Social and Behavioral Sciences 20, 514–523 (2011)
38. Vogel, P., Neumann Saavedra, B., Mattfeld, D.: A hybrid metaheuristic to solve the resource allocation problem in bike sharing systems. In: Blesa, M., Blum, C., Voß, S. (eds.) Hybrid Metaheuristics, Lecture Notes in Computer Science, vol. 8457, pp. 16–29. Springer International Publishing (2014)

REFERENCES

Andersen, I. (2015). Kjøp månedskort på bussen, få tilgang til elsykkel. Teknisk Ukeblad. Available at: http://www.tu.no/artikler/222982. (Published: 2015-February-01, Accessed: 2016-May-24).

Angeloudis, P., Hu, J., and Bell, M. G. (2014). A strategic repositioning algorithm for bicycle-sharing schemes. *Transportmetrica A: Transport Science*, 10(8):759–774.

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

Berbeglia, G., Cordeau, J.-F., Gribkovskaia, I., and Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *Top*, 15(1):1–31.

Berbeglia, G., Cordeau, J.-F., and Laporte, G. (2010). Dynamic pickup and delivery problems. *European journal of operational research*, 202(1):8–15.

Brinkmann, J., Ulmer, M. W., and Mattfeld, D. C. (2015a). Inventory routing for bike sharing systems. Available at: https://www.tu-braunschweig.de/Medien-DB/winfo/publications/wp_brinkmann_iventory_routing_bike_sharing.pdf. Working paper. Technische Universität Braunschweig. (Published: 2015-January-12, Accessed: 2015-October-15).

Brinkmann, J., Ulmer, M. W., and Mattfeld, D. C. (2015b). Short-term strategies for stochastic inventory routing in bike sharing systems. *Transportation Research Procedia*, 10:364–373.

Caggiani, L. and Ottomanelli, M. (2012). A modular soft computing based method for vehicles repositioning in bike-sharing systems. *Procedia-Social and Behavioral Sciences*, 54:675–684.

Caggiani, L. and Ottomanelli, M. (2013). A dynamic simulation based model for optimal fleet repositioning in bike-sharing systems. *Procedia-Social and Behavioral Sciences*, 87:203–210.

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

Christiansen, M. (1999). Decomposition of a combined inventory and time constrained ship routing problem. *Transportation Science*, 33(1):3–16.

Clear Channel Norge (2015). Du kan sykle med oss. Available at: http://clearchannel.no/produkter-priser/bysykkel. (Accessed: 2015-December-12).

Contardo, C., Morency, C., and Rousseau, L.-M. (2012). Balancing a dynamic public bike-sharing system. Technical Report CIRRELT-2012-09, Universitè de Montrèal, Montrèal, Canada.

Dell'Amico, M., Hadjicostantinou, E., Iori, M., and Novellani, S. (2014). The bike sharing rebalancing problem: Mathematical formulations and benchmark instances. *Omega*, 45:7–19.

Dell'Olio, L., Ibeas, A., and Moura, J. L. (2011). Implementing bike-sharing systems. *Proceedings of the ICE-Municipal Engineer*, 164(2):89–101.

DeMaio, P. (2009). Bike-sharing: History, impacts, models of provision, and future. *Journal of Public Transportation*, 12(4):3.

DeMaio, P. and Meddin, R. (2016). The bike-sharing world map. Available at: www.bikesharingmap.com. (Accessed: 2016-May-24).

Desrochers, M. and Laporte, G. (1991). Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10(1):27–36.

Erdoğan, G., Battarra, M., and Calvo, R. W. (2015). An exact algorithm for the static rebalancing problem arising in bicycle sharing systems. *European Journal of Operational Research*, 245(3):667–679.

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

Felin, L. (2016). Interview with COO at Urban Infrastructure Partner. Oslo, 13-may-2016.

Forma, I. A., Raviv, T., and Tzur, M. (2015). A 3-step math heuristic for the static repositioning problem in bike-sharing systems. *Transportation research part B: methodological*, 71:230–247.

Fricker, C. and Gast, N. (2014). Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. *EURO Journal on Transportation and Logistics*, 3:1–31.

García-Palomares, J. C., Gutiérrez, J., and Latorre, M. (2012). Optimizing the location of stations in bike-sharing programs: a GIS approach. *Applied Geography*, 35(1):235–246.

Gaspero, L., Rendl, A., and Urli, T. (2015). Balancing bike sharing systems with constraint programming. *Constraints*, 20:1–31.

Go Bike (2014). By- & pendlercyklen. Available at: http://bycyklen.dk/en/. (Accessed: 08-November-2015).

Hernández-Pérez, H. and Salazar-González, J.-J. (2007). The one-commodity pickup-and-delivery traveling salesman problem: Inequalities and algorithms. *Networks*, 50(4):258–272.

Hernández-Pérez, H. and Salazar-González, J.-J. (2003). The one-commodity pickup-and-delivery travelling salesman problem. In Jünger, M., Reinelt, G., and Rinaldi, G., editors, *Combinatorial Optimization — Eureka, You Shrink!*, volume 2570 of *Lecture Notes in Computer Science*, pages 89–104. Springer Berlin Heidelberg.

Ho, S. C. and Szeto, W. (2014). Solving a static repositioning problem in bike-sharing systems using iterated tabu search. *Transportation Research Part E: Logistics and Transportation Review*, 69:180–198.

Høgåsen-Hallesby, J. (2016). Interview with CTO/Marketing Director at Urban Infrastructure Partner. Oslo, 23-may-2016.

Kaspi, M., Raviv, T., and Tzur, M. (2015). Detection of unusable bicycles in bike-sharing systems. Available at: http://www.eng.tau.ac.il/~talraviv/Publications/Detection%20of%20Unusable%20Bicycles.pdf. Working paper. Tel-Aviv University. (Published: 2015-October, Accessed: 2015-November-08).

Kloimüllner, C., Papazek, P., Hu, B., and Raidl, G. R. (2014). Balancing bicycle sharing systems: an approach for the dynamic case. In *Evolutionary Computation in Combinatorial Optimisation*, pages 73–84. Springer.

Lin, J.-R. and 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.

Midgley, P. (2011). Bicycle-sharing schemes: enhancing sustainable mobility in urban areas. In *19th session of the Commission on Sustainable Development. 2011-May-02*. United Nations, Department of Economic and Social Affairs. Background Paper No. 8.

Miller, C. E., Tucker, A. W., and Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*, 7(4):326–329.

Morgan, J. (2014). A simple explanation of 'the Internet of Things'. Available at: http://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand. (Published: 2014-May-13, Accessed: 2015-December-08).

Nair, R. (2010). *Design and analysis of vehicle sharing programs: A systems approach*. PhD thesis, University of Maryland, College Park.

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

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

O'Mahony, E. and Shmoys, D. B. (2015). Data analysis and optimization for (citi) bike sharing. In *Twenty-Ninth AAAI Conference on Artificial Intelligence. 2015-January-25*. Association for the Advancement of Artificial Intelligence.

Oslo Bysykkel (2016). Hva er Oslo Bysykkel? Available at: https://oslobysykkel.no/om-bysykkel. (Accessed: 2016-May-18).

Parikh, P. and Ukkusuri, S. V. (2015). Estimation of optimal inventory levels at stations of a bicycle sharing system. In *TRB 94th Annual Meeting. 2015-January-11*. Transportation Research Board. Paper No. 15-5170.

Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11.

Psaraftis, H. N., Wen, M., and Kontovas, C. A. (2016). Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1):3–31.

Rainer-Harbach, M., Papazek, P., Hu, B., and Raidl, G. (2013a). Balancing bicycle sharing systems: A variable neighborhood search approach. In Middendorf, M. and Blum, C., editors, *Evolutionary Computation in Combinatorial Optimization*, volume 7832 of *Lecture Notes in Computer Science*, pages 121–132. Springer Berlin Heidelberg.

Rainer-Harbach, M., Papazek, P., Raidl, G. R., Hu, B., and Kloimüllner, C. (2013b). PILOT, GRASP, and VNS approaches for the static balancing of bicycle sharing systems. *Journal of Global Optimization*, 63(3):1–33.

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

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

Regue, R. and Recker, W. (2014). Proactive vehicle routing with inferred demand to solve the bikesharing rebalancing problem. *Transportation Research Part E: Logistics and Transportation Review*, 72:192–209.

Ritzinger, U., Puchinger, J., and Hartl, R. F. (2015). A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, pages 1–17.

Romero, J. P., Ibeas, A., Moura, J. L., Benavente, J., and Alonso, B. (2012). A simulation-optimization approach to design efficient systems of bike-sharing. *Procedia-Social and Behavioral Sciences*, 54:646–655.

Schuijbroek, J., Hampshire, R., and van Hoeve, W.-J. (2013). Inventory rebalancing and vehicle routing in bike sharing systems. Available at: `http://repository.cmu.edu/cgi/viewcontent.cgi?article=2490&context=tepper`. Working paper. Tepper School of Business. (Published: 2013-February, Accessed: 2015-December-01).

Seo, Y., Kim, D.-K., and Kho, S.-Y. (2015). A static positioning strategy for public bicycle systems considering penalty cost: Comparison of different demand patterns. In *EASTS 2015 Conference. 2015-September-11*. Eastern Asia Society for Transportation Studies.

Shaheen, S., Guzman, S., and Zhang, H. (2010). Bikesharing in Europe, the Americas, and Asia: past, present, and future. *Transportation Research Record: Journal of the Transportation Research Board*, 2143:159–167.

Sörensen, K. and Dilip, D. (2014). The (city) bike request scheduling problem–a novel approach to solve the city bike repositioning problem. In Toklu, Y. C. and Bekdas, G., editors, *Metaheuristics and Engineering*, volume 15 of *Workshop of the EURO Working Group*, pages 157–161. Bilecik Şeyh Edebali University.

Toth, P. and Vigo, D. e. (2014). *Vehicle routing: problems, methods, and applications*, volume 18. SIAM: Society for Industrial and Applied Mathematics. Philadelphia.

Vogel, P., Ehmke, J. F., and Mattfeld, D. C. (2015). Service network design of bike sharing systems. Available at: https://www.tu-braunschweig.de/Medien-DB/winfo/publications/service_network_design_of_bike_sharing_systems.pdf. Working paper. Technische Unversität Braunschweig. (Published: 2015-March-24, Accessed: 2015-September-25).

Vogel, P., Greiser, T., and Mattfeld, D. C. (2011). Understanding bike-sharing systems using data mining: Exploring activity patterns. *Procedia-Social and Behavioral Sciences*, 20:514–523.

Vogel, P., Neumann Saavedra, B., and Mattfeld, D. (2014). A hybrid metaheuristic to solve the resource allocation problem in bike sharing systems. In Blesa, M., Blum, C., and Voß, S., editors, *Hybrid Metaheuristics*, volume 8457 of *Lecture Notes in Computer Science*, pages 16–29. Springer International Publishing.

Wikipedia (2015). Hangzhou public bicycle. Available at: https://en.wikipedia.org/w/index.php?title=Hangzhou_Public_Bicycle&oldid=684974719. (Accessed: 27-October-2015).