Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

Jørgen Ringvold Ljønes

Multi-Depot Periodic Vehicle Routing Problem under Cleaning Constraints at the Oslo Metro

Master's thesis in Industrial Economics and Technology Management
Supervisor: Magnus Stålhane
June 2020

Line 3 towards Mortensrud (foto: Kjetil Ree, licensed under CC BY 2.0)

NTNU
Norwegian University of
Science and Technology

Jørgen Ringvold Ljønes

Multi-Depot Periodic Vehicle Routing Problem under Cleaning Constraints at the Oslo Metro

Master's thesis in Industrial Economics and Technology Management
Supervisor: Magnus Stålhane
June 2020

Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

NTNU
Norwegian University of Science and Technology

Problem Description

The goal of this master's thesis is to develop a mixed integer program formulation for a large matching problem routing rolling stock at the Oslo Metro between depots and the start and end of service routes. The object is to minimize deadheading while fulfilling periodic cleaning requirements over realistic instances with long planning horizons.

Main contents:

1. Description of the problem.
2. Literature review relevant to the application area and problem description.
3. Mathematical model for the problem.
4. Exact and heuristic solution methods.
5. Implementation and comparison of the proposed solution methods using a commercial solver.
6. Presentation of results and discussion of the possible implications and further research.

Preface

This master's thesis concludes my Master of Science degree in Industrial Engineering and Technology Management with specialization in Optimization and applied Economics at the Norwegian University of Science and Technology. The thesis builds upon my project report in the subject TIØ4500 - Managerial Economics and Operations Research, Specialization Project, researched and written during the fall semester 2018. Note that because of personal circumstances, with no relation to the content of this work, I did not finish this thesis in the spring of 2019 as initially planned. Throughout this thesis I assume that the reader is familiar with basic notation, terminology and concepts in operations research.

The idea for exploring these research questions was found in collaboration with Helge Holtebekk, Marius Sommerseth and Kjersti Moss at Sporveien AS in 2017. I am grateful for the opportunity to work with such an interesting, real world optimization problem with the Oslo Metro. In my experience, there are several unsolved problems in the field of Urban Rapid Transit Systems and observing this potential of societal benefits at the Oslo Metro has been especially motivating. I would like to thank my supervisor, Magnus Stålhane, Professor at the Department of Industrial Economics and Technology Management, NTNU, and Helge Holtebekk, Head of Analyses at Sporveien AS for their help with this report. I also thank my lovely wife, Anna, for being totally awesome.

Jørgen Ringvold Ljønes
Oslo, June 2020

Abstract

Mass rapid transit systems like the Oslo Metro are an important part of a city's transport service in a world of increased urbanization and demand for environmentally friendly transport solutions. Increasing size and complexity of transport systems also provides an opportunity for optimizing scheduling and planning to reduce costs. A significant cost saving measure is to reduce the distance empty trains drive between depots and terminal stations. Driving empty trains between depots and the first and last station of a day is called *deadheading*. Because of the integrated network design the problem of reducing deadheading is particularly relevant at the Oslo Metro.

A plan that allocate trains to depots and passenger routes must also satisfy cleaning constraints as all trains must be cleaned within certain time intervals, and only some depots have cleaning equipment. Such a plan must also adhere to the maximum storing capacity of all depots.

This thesis aims to find allocation plans that minimize deadheading in a realistic situation at the Oslo metro over a long planning horizon while satisfying all practical side constraints. The main contributions of this thesis is defining the Multi-Depot Periodic Vehicle Routing Problem, formulating a mathematical model for the problem, developing an exact and heuristic solution method and testing the heuristic solution method for realistic instances with long planning horizons. No previous research literature on metro systems are found that study this problem, however, related research on the similar bus rapid transit systems exists and are discussed in the thesis. A half-year allocation plan for the Oslo Metro is found using the proposed heuristic solution method. This solution is estimated to pose an 18.6 percent improvement, equivalent to a cost saving of about NOK 3.4 million, over the current plan used by Sporveien AS, the operator of Oslo Metro. The solution is found using a commercial solver on a standard desktop computer within 24 hours of computing time.

Sammendrag

Kollektivtransportsystemer som T-banen i Oslo dekker en essensiell del av storbyers behov for persontransport. Verden opplever en økende urbanisering og etterspørsel etter klimavennlige transportløsninger. Større og mer komplekse transportsystemer gir samtidig en mulighet for å for å redusere kostnader gjennom operasjonsanalyse. En betydelig kostnadsbesparing er å redusere tomkjøring av tog mellom depot og første og siste stasjon på en T-banerute. På grunn av det integrerte designet er dette problemet spesielt presserende for T-banenettverket i Oslo.

En allokeringsplan av tog til depot og ruter må også tilfredstille togenes vaskekrav innen visse tidsintervall, og bare noen depot er utstyrt med vaskeutstyr. En slik plan må også overholde depotenes makskapasitet.

Denne oppgaven forsøker å finne allokeringsplaner som minimerer tomkjøring i realistiske scenarier ved T-banen i Oslo over lange planleggingshorisonter mens vaskekrav er tilfredstilt. Oppgavens hovedbidrag er å definere problemet, formulere en tilhørende matematisk modell, utvikle en eksakt og en heuristisk løsningsmetode og teste den heuristiske løsningsmetoden for realistiske instanser over lange planleggingshorisonter. Det er ikke identifisert tidligere forskningslitteratur på T-banesystemer som dekker dette problemet, men relatert forskning på metrobussystemer finnes og er diskutert i oppgaven. En halvårlig allokeringsplan for T-banen i Oslo er funnet i oppgaven ved bruk av den foreslåtte heuristiske løsningsmetoden. Denne løsningen er estimert til å utgjøre en 18,6 prosent forbedring, tilsvarende en kostnadsbesparelse på 3,4 millioner kroner, sammenlignet med dagens løsning fra Sporveien AS, operatøren av T-banen i Oslo. Løsningen er funnet ved bruk av kommersiell programvare på en vanlig stasjonær datamaskin innen en løsningstid på 24 timer.

Our existence in time
is determined for us,
but we are largely free
to select our location.

— August Lösch (1954, p.3)

Table of Contents

List of Figures

List of Tables

Acronyms

AS	limited company [Norwegian: aksjeselskap]
B&B	Branch-and-Bound
BRT	bus rapid transit
CP	central period
CT	cost threshold
FoP	forecasting period
FzP	frozen period
hr	hour
IP	integer programming
km	kilometer
LB	lower bound
LP	linear program(ing)
MDPVRP	Multi-Depot Periodic Vehicle Routing Problem
MDVRP	Multi-Depot Vehicle Routing Problem
MIP	mixed-integer programming
NOK	Norwegian Krone
NP-hard	non-deterministic polynomial-time hard
PVRP	Periodic Vehicle Routing Problem
RHH	Rolling Horizon Heuristic
SSR	solution space reduction
TW	time window
UB	upper bound
VRP	Vehicle Routing Problem
VSP	Vehicle Scheduling Problem

1 Introduction

More than half the world's population now live in cities and the urbanization trend is only increasing. Since 1980, the number of people living in cities with more than 500,000 inhabitants has tripled. Larger and denser cities bring both challenges and opportunities. With more people living and working close together, traffic congestion and pollution become major problems. Relying on cars for personal transportation brings immense social costs, including using high value real estate for roads and parking. On the other hand, increased population density makes mass rapid transit systems like a metro more cost effective and eco-friendly. The world's total rail length built for metro systems has also tripled since 1980 from about 5,000 km to over 15,000 km (Institute for Transportation & Development Policy, 2017). Oslo, the capital of Norway, is also growing, recently passing 1 million inhabitants in the greater Oslo area with an estimated total of 1.2 million by 2040 (Statistics Norway, 2018). The Oslo City Council has decided that cars for personal transport shall be banned from the city center, further strengthening the need for alternative transport systems (Business Insider, 2018).

Sporveien AS is the operator of the Oslo Metro and the technical collaborator of this thesis. The Oslo Metro serviced 119 million travelers in 2019, a 25 percent increase since 2015 (Sporveien AS, 2019). From 2014 to 2017, the average cost-per-trip went down from NOK 11.5 to NOK 8.5, a decrease of 35 percent, following a period with large cost saving measures and increased production. The latest numbers from 2019 show a rebound to about NOK 9.5 average cost-per-trip (Sporveien AS, 2019). A significant operational cost is driving empty trains, so-called *deadheading* or *dead mileage*, between the depots where the trains are parked overnight to the starting stations of the metro lines. Similarly, at the end of the day, the trains deadhead back to a depot from the last station of the line they were plying, which means serving a line. Driving trains without passengers is a costly, non-revenue generating activity and thus desirable to reduce. Every kilometer of deadheading costs Sporveien about NOK 43 in power consumption, wear and personnel costs, totaling NOK 36 million over the course of a year. Sporveien wants to continue their cost saving measures and reduce the total deadheading length by finding a better routing plan between depots and the terminal stations.

Within a given time interval all trains must visit certain depots for cleaning, however the depots have limited capacity. For a given planning horizon, a schedule must alternate which trains are assigned a cleaning depot to fulfill this requirement. The goal is to minimize deadheading, while not exceeding the maximum depot capacities and adhering to the periodic cleaning requirements as well as fulfilling the timetable. Trains are assigned a metro line and a timetable block to ply for the day. Blocks are a set of routes and technical movements trains perform over the course of a day in traffic. Aggregating all blocks generates the complete timetable. The timetable, and the individual blocks with it, differ between weekdays, Saturdays and Sundays, with reduced service on Saturdays and Sundays. Public holidays run on a Sunday timetable.

The purpose of this thesis is constructing a model and solution method that find an improved routing of trains between depots and the terminal stations while fulfilling multi-period cleaning constraints. This has — to my knowledge — not been studied or solved before in published academic literature in English. Although the literature on deadheading minimization in metro systems is non-existing, literature found on problems in city bus systems share many of the core properties of the problem studied in this thesis.

In this thesis, I develop an integer programming (IP) model for the Multi-Depot Periodic Vehicle Routing Problem (MDPVRP) which is outlined above. Moreover, I develop an heuristic approach based on the Rolling Horizon Heuristic (RHH) and solution space reduction (SSR) techniques and compare this with an exact Branch-and-Bound (B&B) algorithm using a commercial solver. The purpose is to identify solution methods suitable for solving instances representing Sporveien's current situation over long planning horizons. Long deadheading planning schedules makes personnel and long-term maintenance planning easier.

The largest instance I manage to solve in this thesis is the half year instance. The proposed solution suggests that Sporveien may improve its block-vehicle-depot matching by 18.6 percent in terms of reduced deadheading, which amounts to a cost reduction of about NOK 3.4 million over a six month period.

This thesis is a continuation of my work on a project report on the same theme and problem. The focus of the report was to formulate the problem and implement an exact solution method. However, most of this work has since been developed further and transformed such that only the general idea and preparation of instance input data remains to be found in this thesis. The chapters that still inherit a significant part from the report are Chapters 2, 4 and 5. I will not proceed to reference the report throughout the thesis.

The remainder of this thesis is structured as follows: Chapter 2 expands on Sporveien and the Oslo Metro to provide a technical background. A literature study is presented in Chapter 3 to discuss the problem properties, application area and solution methods in previous research. Chapter 4 defines the problem, and Chapter 5 presents its mathematical formulation. In Chapter 6 the exact and heuristic solution methods are described. Sporveien's current situation represented as problem instances of different lengths are presented in Chapter 7. A computational study of the solution methods proposed and other scenarios are presented in Chapter 8 and finally, Chapter 9 offers concluding remarks.

2 Technical Background

This chapter provides an introduction to the company Sporveien AS in general and Oslo Metro in particular, as well as a background on the challenges of cleaning and storing metro trains. I present the publicly owned company Sporveien AS and its subsidiary Sporveien T-banen AS in Section 2.1. In Section 2.2 I describe Oslo Metro on a general level and in Section 2.3 I go into detail about how the trains are cleaned and stored. Section 2.4 discusses the cost drivers and the utility Sporveien is interested in achieving. All specific information provided in this chapter is either sourced from conversations with representatives at Sporveien or their latest annual report (Sporveien AS, 2019).

2.1 Sporveien AS

Sporveien AS (from now on just *Sporveien*) is a company owned by the Municipality of Oslo, the capital of Norway. It owns, operates and maintains the tracks and rolling stock — all trains and trams — of the Oslo Metro and Oslo Tramway. Sporveien also owns Unibus, a bus operating company and two other smaller operating subsidiaries as well as Sporveien Vognmatriell AS, formerly a separate company named Oslo Vognselskap. Sporveien Vognmatriell owns the metro and tramway rolling stock and lease it to Sporveien Metro and Sporveien Tramway which maintains the rolling stock in-house. See Figure 2.1 for a visual representation of Sporveien, their owners and subsidiaries. In 2019 Sporveien employed 3,351 people and transported a total of 269 million journeys, with NOK 4,785 million in total revenue. The public transport authority, Ruter, is the official buyer of transport services from Sporveien, which in turn offer transport services to the public. Ruter is owned 60 percent by the Municipality of Oslo and 40 percent by the County of Viken, the neighbouring county to Oslo.

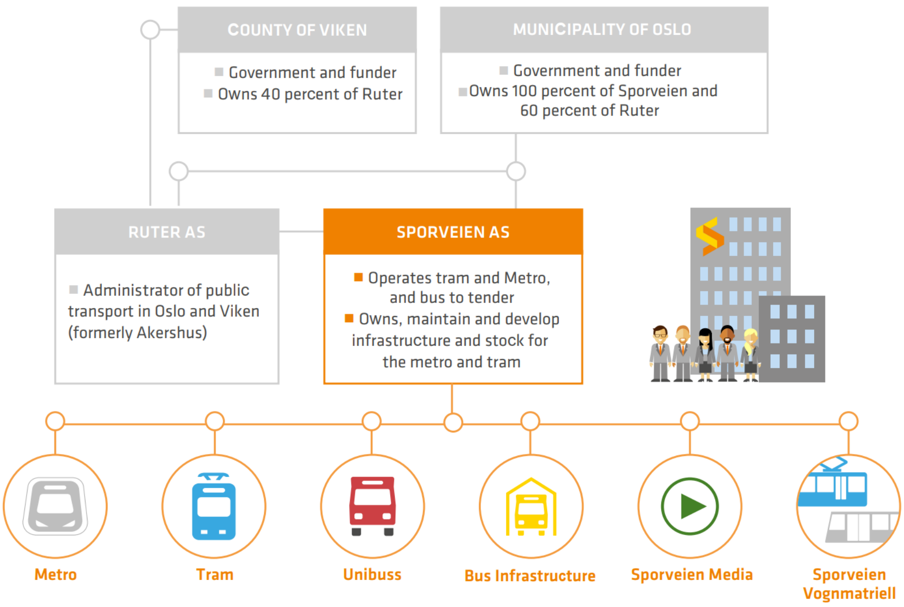

Figure 2.1 – Sporveien, its owners and subsidiaries. Translated from Sporveien AS (2019)

2.2 Oslo Metro

Oslo Metro (from here on "the Metro") is the partly underground urban train in Oslo, functioning as the city's rapid transit system. The Norwegian name "T-bane" is short for "Tunnelbane" (literally *tunnel rail*) and corresponds to the more colloquial English terms *Underground* or *Tube* (British), *Subway* (North American), *U-Bahn* in German speaking countries or *Metro*, which is more used internationally (Wikipedia, 2020b).

The rail network has a total length of 85 km and consists of five lines that all run through the city centre, see Figure 2.2. The network was historically built as separate networks on each side of the city, but was later merged in the city center and the Ring (Line 5) was completed with the northern stations Nydalen, Storo and Sinsen in 2006. There are 17 underground or indoor stations, and 84 stations above ground. The Metro serves 14 out of the 15 boroughs in Oslo, St. Hanshaugen in the center of the Ring being the exception. Two westward going lines run to Kolsås and Østerås in the neighboring municipality of Bærum in the county of Viken. In 2019, the Metro served 119 million individual travels.

Figure 2.2 – The Oslo Metro schematic map. The stations with a co-located depot are marked in red. Names marked with a red star are the associated stations where depots have cleaning equipment. Copyright Truls Lange Civitas

As of 2020, Sporveien operates 115 trains of the type *OS MX3000* produced by Siemens, consisting of three cars each. Usually the trains are driven as two coupled trains, as shown in Figure 2.3, but during periods of low demand, and predominantly on Line 1 towards Frognerseteren where the stations are shorter, the trains are driven individually. During deadheading, the trains may be coupled three and four together to move up to twelve cars at the time with a single driver. Because of the limited length of the stations, the trains are without passengers while driving more than two coupled sets. The interior of the MX3000 is open and modern (see Figure 2.4) with a moderate number of seats and high total passenger capacity. Each three-car train has a rated capacity of 400 passengers.

Figure 2.3 – Two coupled Siemens OS MX3000 trains. Photo: User "Falk2", Wikipedia

Figure 2.4 – The interior of Siemens OS MX3000 train. Photo: Sean Hayford O'Learly

2.3 Cleaning, storing and other restrictions

The Siemens OS MX3000 train series was produced exclusively for the Oslo Metro and in 2010 the last of the older trains was replaced with an MX3000 model. Having a custom-made and homogeneous fleet of trains greatly simplifies the infrastructure requirements and maintenance routines. All trains are stored overnight at one of several depots — which consists of a rail yard and garage complex — when they are not in traffic. For about four to five hours each night, and one to two hours longer on weekend nights, there is no passenger traffic.

The depots at *Avløs* and *Ryen* are larger and at a side track to the stations with the same names. The stations *Vestli*, *Ellingsrudåsen* and *Stortinget* are also used as depots, with more limited storage capacity. Only parts of the capacity at Vestli and Ellingsrudåsen may be utilized without interfering with trains in service. The storage space at Stortinget station is on the old turning loop, a part of the track from before the eastern and western network was combined in the city center (see Figure 2.5). At that time, all trains on the eastern lines turned around using this loop, which is built on a different level than the newer main track. Today the loop is still used as an easy way of turning around trains coming from an eastern line going to another eastern line, e.g. Line 5 from Ellingsrudåsen to Line 3 Mortensrud. Before the end of the operating day, one of the two tracks in the loop may be used as storage without interfering with trains in service. The second loop track is held clear for turning maneuvers and are not used as storage until later in the night when most other trains are in their depots and the timetabled traffic has ended.

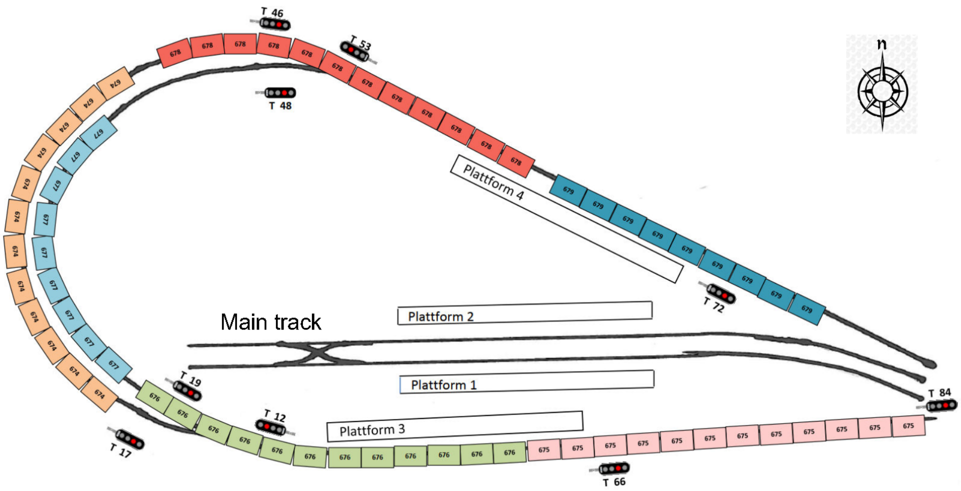

Figure 2.5 – Layout of Stortinget station with the main track going under the looping track from the east. Colored boxes are spots for storing trains overnight. Illustration from Sporveien.

Figure 2.6 – Two Siemens OS MX3000 trains on the maintenance platform at Ryen depot. Photo: Sporveien

As seen on the map of the Metro (Figure 2.2), the five locations with storing capacity have their names marked with red background. Two of the depots, Avløs and Ryen, have equipment to clean the trains interior. Trains that are stored on the turning loop at Stortinget station are first prepared for storage at Ryen, where their interior may be cleaned. This means Stortinget can be thought of as a depot with interior cleaning capability. These three depots with interior cleaning capabilities are marked with a red star in Figure 2.2. The depot at Ryen also have a washing tunnel for exterior cleaning.

The trains' interior and exterior are cleaned for hygienic and aesthetic reasons and to reduce corrosion and wear. Cleaning is performed at the depot where a train is stored overnight, but only if the depot has the necessary facility. The cleaning capacity is limited by the total depot capacity, so time needed to clean is not a limiting constraint. Interior cleaning is a fairly simple process performed by two to three cleaning personnel who clean the floor, walls and windows and replace garbage bags. The process takes about 20 minutes per train and is done at least every other day on each train. Exterior cleaning, which is only available at the Ryen depot, should be performed at least once every five days. Here a cleaning operator drive the trains through a washing tunnel, not unlike how a car wash tunnel works. Other, larger maintenance tasks are performed on the trains while they are on service platforms as seen in Figure 2.6. There is a second service floor on a level below the train and scaffolding on the sides and above such that maintenance personnel may access all parts of the train exterior.

The full fleet of trains leased by Sporveien numbers 115. During rush hour on weekdays 105 trains are in normal traffic, and on Saturdays or Sundays at most 98 and 82 trains respectively. The remaining trains are out of service for preventive maintenance, repair, upgrades or just held as backup. Since the fleet is homogeneous, they circulate the sets that are out of service so that incoming trains replace outgoing ones. Long-term maintenance, repairs and upgrades are not considered in this thesis.

2.4 Utility and costs

Sporveien delivers transport services according to a contract with Ruter, the public transport authority. In addition to customer satisfaction targets, Sporveien's goal is to serve passenger travels as cost-effectively as possible. Important cost drivers are the driver personnel cost, security and maintenance of trains and tracks, power consumption for propulsion and lighting, ventilation and air conditioning at the stations as well as maintenance equipment and spare parts. At the start of a day the trains in service leave the depots and deadhead to the first station of their allocated block for that day. A block is a sequence of in-service trips that a train performs during a day. The aggregation of all blocks describes the flow of trains on all lines that together constitute the timetable.

If Sporveien manage to decrease total deadhead travel length between depots and terminal stations, it will result in reduced power consumption, less wear and lower personnel costs. Sporveien estimates that the average cost of wear and power for a single train is about NOK 30 per km driven. Further, they estimate an average personnel cost during the relevant hours at about NOK 800 per hour. The average driving speed during normal traffic is 30 km/hr. Early in the morning and late in the evening, the traffic is less busy, which allows for increased driving speed, but when calculating the driver personnel cost per km driven, this is more or less offset by the drivers inconvenience allowance. An exception to this is the western branch of Line 1 from Majorstuen towards and including Frognerseteren (see Figure 2.2), an older and steeper part of the network. This branch is usually only served by a single train and have an reduced average speed of 20 km/hr.

According to Sporveiens current rolling stock schedule, the sum of deadheading on a regular weekday is 2,504 km, on a Saturday it is 2,055 km and on a it is Sunday/holiday 1,655 km. For the days in the year of 2020, this sums to 842,312 km of total deadheading between depots and block terminal stations.

To summarize, Sporveien is the largest provider of public transport in the Oslo area, and the Metro accounts for a majority of this. The Metro have a highly interconnected rail network with many possible combinations of depots and blocks for all trains in the homogeneous fleet. Cleaning of the trains is only possible at some of the depots. Deadheading is costly and dependent on the matching of trains to depots and blocks. Finding a better schedule matching trains to depots and blocks, while still fulfilling the cleaning restrictions may be a considerable cost saving measure.

3 Literature Review

In this chapter I explore and present literature relevant to the Multi-Depot Periodic Vehicle Routing Problem (MDPVRP) for depot-block routing of metro trains. The introductory Section 3.1 presents the literature on the Vehicle Routing Problem (VRP) and its extensions, as well as discussing how the problem of this thesis can be modeled as a series of matching problems. Section 3.2 explains my search strategy for covering the literature on metro systems and problems related to the MDPVRP, and presents the findings. In Section 3.3 I discuss the relevance of the MDPVRP in other transport systems and review more thoroughly the literature on the MDPVRP in bus rapid transit (BRT) systems. In Section 3.4 I focus on solution methods for large problem instances by providing a short introduction to the rolling horizon heuristic and present relevant literature on this approach. In the final Section 3.5 I summarize the most important findings in this chapter and discuss this thesis' contribution to the literature.

3.1 Vehicle Routing Problem and its extensions

The MDPVRP is derived from the well-known VRP — introduced by Dantzig and Ramser (1959) — which aims to design a set of minimum cost vehicle routes through a network visiting a set of delivery locations, so that each route starts and ends in a depot, often while satisfying some side constraints. Pickup and delivery applications like goods or mail delivery or transportation problems like passenger bus routing are typical examples of the VRP. The rich literature on the VRP covers many derivatives and application areas, Golden et al. (2008) being a thorough survey and Adewumi and Adeleke (2018) being a more recent one.

When a network include multiple depots, the extended VRP is called Multi-Depot Vehicle Routing Problem (MDVRP). Like with VRPs the goal of a MDVRP is to find a set of routes involving all depots that minimizes the total travel cost (e.g. distance), while satisfying constraints like customer demand and vehicle capacity. Sometimes the MDVRP extends to deciding the location of all or some depots, like the decision of where to build ports to better accommodate supply ships serving a network of off-shore oil rigs (Uyeno & Willoughby, 1995). Even without deciding the location of depots, the MDVRP is proven to be NP-hard (Bertossi et al., 1987), which means the problem is not solvable in polynomial time using known algorithms. For a review of the MDVRP literature see Montoya-Torres et al. (2015).

A routing problem defined over a discrete number of periods, like days, extends the VRP to the Periodic Vehicle Routing Problem (PVRP). The goal is still to minimize total travel cost, but across all time units in the planning horizon. An example of the PVRP is public waste collection which must be preformed periodically and are somewhat flexible as to the exact day and time frequency. Minimizing the total travel cost for waste collection vehicles by chosing routes and frequency constitute a typical PVRP. For a survey on recent advances in the PVRP literature, see Section 5 of Adewumi and Adeleke (2018).

The Metro is a connected rail network where trains on all lines may use any depot in the network, but only some of the depots have cleaning equipment. The cleaning requirements are defined over a period of days which make the problem studied in this thesis a combination of MDVRP and PVRP. This combination is called Multi-Depot Periodic Vehicle Routing Problem (MDPVRP), sometimes unfortunately named Multi-Period Multi-Depot Vehicle Routing Problem which is easily confused with the Multi-*Product* Multi-Depot Vehicle Routing Problem, which is not relevant here. However, the version of the MDPVRP studied in this thesis is different from most other MDPVRPs — like the one studied in Hadjiconstantinou and Baldacci (1998) — in two important aspects: Firstly — illustrated by solution a) in Figure 3.1 — each vehicle route starts and ends at the same depot in a traditional MDPVRP. The trains at the Metro is not required to return to the same depot, which increase the number of legal paths in a network, as illustrated by solution b). Secondly, because the timetabled blocks in this thesis are *locked*, or taken as given, each block is represented by a single "delivery" node. All block nodes must be visited and no vehicle may visit more than one. This is illustrated by solution c) in Figure 3.1 with the number of nodes reduced to the number of vehicles for clarity.

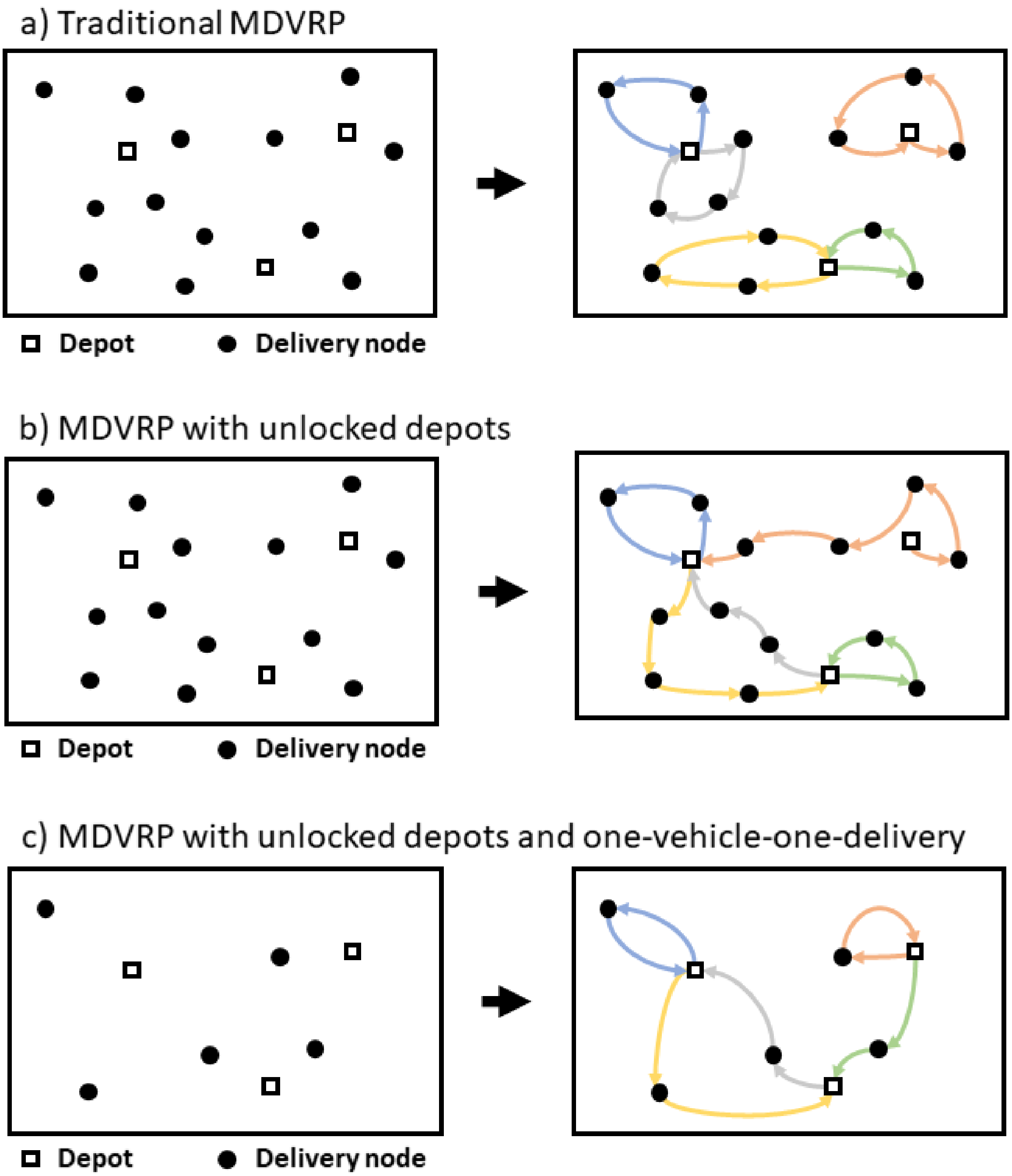

Figure 3.1 – a) A solution from a traditional Multi-Depot Vehicle Routing Problem (MDVRP). A single period is illustrated for clarity. b) A MDVRP where vehicles are not locked to a particular depot and c) a MDVRP like the one in this thesis, with unlocked depots *and* where each vehicle may only visit a single delivery node. In the last solution, the number of delivery nodes is reduced to the number of vehicles for clarity.

When each node is a block and no vehicles may visit more than one block before returning to a depot, the problem may be represented as a bipartite matching problem (Bertossi et al., 1987). To see why this is the case, consider each pair of same-colored edges from the paths in c). Each vehicle is matched with a depot and block in the morning, representing the first edge going from a depot to a node, and a block and depot in the evening, representing going back from the node to a depot. The depot and block nodes in each of the morning and evening parts are separable to become two disjoint and independent sets with edges going from one set to the other, see Figure 3.2, which are the definition of bipartite graphs.

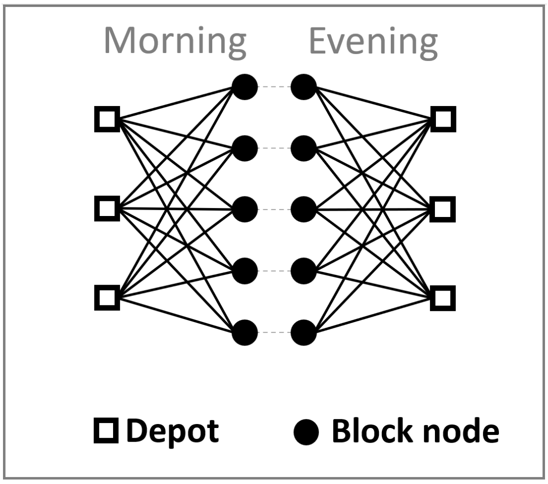

Figure 3.2 – The matching options of depots and block nodes as two complete bipartite, acyclic graphs.

Kepaptsoglou et al. (2010) study a similar problem where vehicles associated with a timetabled block are matched with depots to minimize deadheading. The authors represent the problem with a bipartite, acyclic graph. This uncomplicated network flow problem is equivalent to a matching problem (Derigs, 1988) and may be modeled as such. Considering not only a single half-day, which is clearly a simple, acyclic bipartite graph, but the complete planning horizon does not increase the complexity significantly. An extension to the MDPVRP is given with Figure 3.3. Although this graph is richer — in that it has more nodes and edges — the extension to a series of bipartite graphs is only a linear increase in complexity.

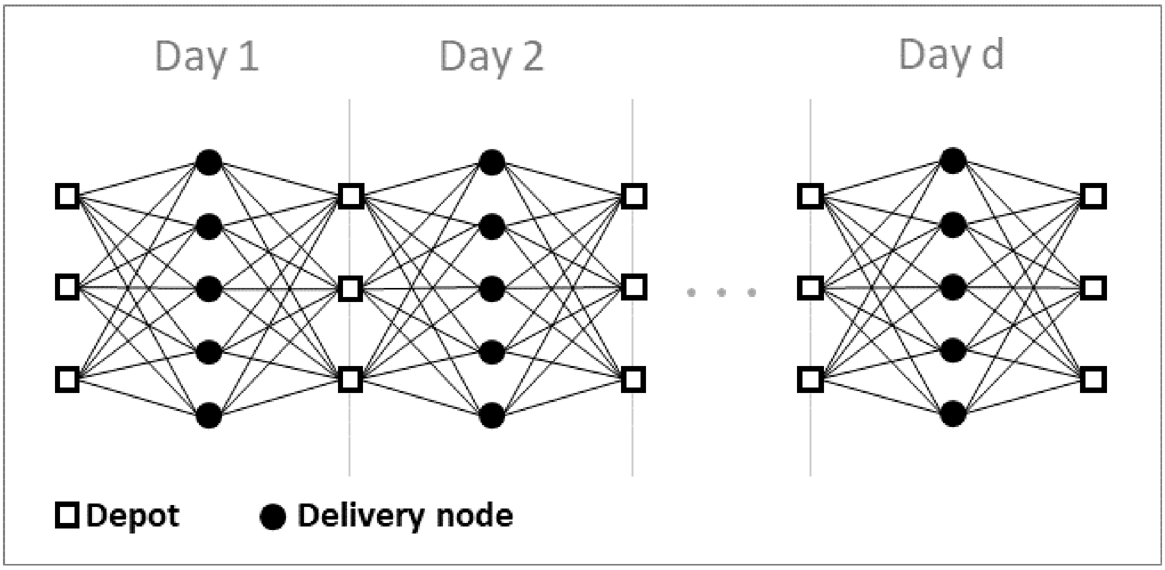

Figure 3.3 – Illustration of the graph structure of the Multi-Depot Periodic Vehicle Routing Problem (MDPVRP) with a single delivery per vehicle in each period. The periodicity does not introduce a more than linear increase in complexity as the complete planning horizon is simply a sequence of separate periods.

In this section I have presented literature relevant to the structure of the MDPVRP studied in this thesis. Moreover, I have discussed different representation of the problem and found that even though it might be represented as a cumbersome sequence of simple graphs, a matching problem formulation is equivalent. Trains are matched with a start depot and a block in the morning, and with the same block and an end depot in the evening each day in a planning horizon under periodic cleaning constraints.

3.2 Literature on the MDPVRP in metro systems

In this section I will present a thorough search for relevant operations research literature with metro systems as an application area. However, I will first introduce terms used to categorize problem areas within the railway industry, where metro systems are considered a member. Several papers have adopted the useful hierarchical structure of terms and illustration from Lusby et al. (2011, p.844), here presented in an adapted form in Figure 3.4. These terms will later be used to categorize the literature found on metro systems.

The many planning problems related to railway systems may be organized sequentially in the order they need to be planned, where each subsequent step is dependent on the decisions made in the previous steps. However, the performance of a particular solution at one step of the planning process is dependent on the subsequent steps. A hypothetical optimal solution to all planning steps would require an incredibly complex model exchanging information between every step in the hierarchy. Such a model is not likely to be computationally tractable, and most research in this area focuses on only one or two planning steps. Figure 3.4 is based on a similar illustration from Lusby et al. (2011) and attempts to give a hierarchical overview of the major planning problems in the railway industry. The *network planning* step is the most basic and concerns the planning of the physical location of tracks, junctions, stations and other physical infrastructure. Projected population patterns and urban development plans often create the basis for the constraints and objective function. *Line planning* covers the decisions of frequency and capacity of lines in the network. These are long term strategic decisions that inform large, hard-to-reverse investments in infrastructure and rolling stock. Strategic level planning problems are only peripherally relevant to this thesis.

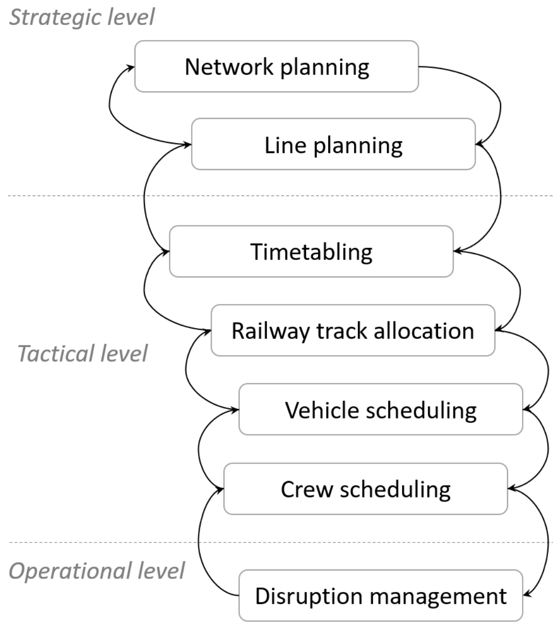

Figure 3.4 – The railway planning problem hierarchy. Illustration adapted from Lusby et al. (2011).

The usual next step is *timetable generation*, or *timetabling* for short, based on the set infrastructure, available rolling stock and required line frequencies. Timetabling aims at determining a periodic timetable for the lines in the railway network that does not violate the physical limitations of the infrastructure, while satisfying some operational constraints. For a survey on the literature on timetabling problems i refer the reader to Cacchiani and Toth (2012). The subsequent step in Figure 3.4 is *railway track allocation* or *train routing* which concerns the detailed routing of lines through junctions, multi-track stations and passing loops where multiple trains must coordinate to ensure conflict free execution, a problem which is usually less precarious to metro systems although necessary. However, as I will return to later in Section 3.3, the Oslo Metro stands out among metro systems as especially sensitive to sub optimal train routing due to the level of interconnection and shared tracks of the different metro lines. The main focus of Lusby et al. (2011) is reviewing the literature on railway track allocation planning problems.

Vehicle scheduling — or when specific to vehicles on rails: *rolling stock scheduling* — is the problem of constructing feasible sequences of trips, or block, as defined by the timetable. A complete schedule is feasible if each trip is assigned the appropriate number and type of rolling stock, and each vehicle performs a feasible sequence of trips. This is the railway equivalent of the more general Vehicle Scheduling Problem (VSP). Scheduling problems are often similar to routing problems, but the former concerns temporal planning and the latter spatial planning. In other words, scheduling is deciding *when* an entity should take a certain action, and routing is deciding *where* an entity should take the action. More often than not, practical problems involves both dimensions and each term is thus often used for problems which does not exclusively concern time or space. In some problems the decision to be made concerns *where* to route a set of vehicles while fulfilling constraints based on *when* such routes are allowed. This is the case in the problem of this thesis. The decision is which routes to choose between depots and blocks, but certain combinations of depots and blocks are not allowed because they are temporally incompatible. Between all planning steps presented in Figure 3.4 routing problems like the one studied in this thesis are most similar to the vehicle scheduling planning problems. Other terms used as synonyms or names for similar problems to vehicle scheduling in the literature combine *rolling stock*, *vehicles* or *trains* with *rostering*, *management*, *planning*, *circulation* or *assignment*. Bunte and Kliewer (2009) seems to be the most recent survey on the vehicle scheduling literature.

A common approach is to consider the scheduling of vehicles and operating crew in the same model, although *crew scheduling* is an independent step on par with the other planning steps in Figure 3.4. Huisman (2004) provides a thorough overview of methods for integrating vehicle and crew scheduling. The most essential crew for operation are the drivers which may or may not be homogeneous in terms of which rolling stock they can be assigned, and schedules are often restricted by maximum allowed working hours and other labor restrictions required by law or employment contracts. Going from the tactical level to the operational level, plans face reality and need real time adjustments. *Disruption management* is the real time management of unplanned events such as infrastructure blockage, failing rolling stock and crew shortage. Research on this problem aims typically to find predefined decision rules and fallback plans as well as methods that would quickly find optimal redirection of rolling stock, crew, passengers and cargo to recover to normal service. See Cadarso et al. (2015) and Lie and Sinnes (2019) as examples on disruption management and recovery for urban rail transit systems.

I have now introduced terminology from railway planning which also are relevant to metro systems. Next in this section I will present a search for literature on metro systems and determine how they relate to the problem studied in this thesis.

To find literature relevant for the MDPVRP in metro systems I used the following search string in a widely used academic search engine (Google Scholar, 2020):

"optimization" ("rapid transit" OR "metro") "depot" "deadheading"

This particular string is based on some preliminary trial searches to find search terms that can identify relevant literature without too many false positives. Terms in quotation marks are required in the search results, but only one of the two terms "rapid transit" and "metro" are strictly needed using this search string. When not including patents and citations, a search on April 14th 2020 provided 143 search results. An evaluation of title and abstract determined if they passed the following three criteria:

1. The document is available in English through my university library
2. It covers a problem from the discipline of operations research
3. Metro systems are mentioned explicitly as a main application area

Through this filtering I reduced 143 search results to 21 confirmed relevant and unique documents, which I further examined to determine scope and problem area. Of the 21 documents, 15 are published papers, three are conference papers and three are PhD theses. All were published between 2003 and 2020, but 10 documents (48 percent) were published in 2018 or later suggesting that the interest in metro systems within operations research has recently increased. Table 3.1 provides a count of documents covering particular planning step as introduced by the problem hierarchy in Figure 3.4. Table 3.2 lists all 21 research documents on metro systems with the planning steps they cover, a short description of the objective function used, if the problem concern more than one depot or a periodic problem. Notes on additional relevant properties are also added.

Most of the documents concern minimizing operating cost or cost drivers like energy consumption. However, none include deadheading costs between depots and blocks as part of their problem. Some optimize for reducing expected delays or passenger wait time through line planning or timetabling. In addition to using objective functions different from the model studied in this thesis, none of the problems in Table 3.2 cover a periodic problem. Without periodicity, the problem structures are fundamentally different from the MDPVRP. Only 8 out of the 21 documents includes a multi-depot problem.

Table 3.1 – Metro planning steps by number of relevant research documents. Note that a single document may cover several steps, so the sum across all planning steps exceed the total of 21 identified documents.

Planning step	No. of Documents
Network planning	2
Line planning	4
Timetabling	8
Railway track allocation	0
Vehicle scheduling	7
Crew scheduling	5
Disruption management	6

To summarize, the literature on metro systems identified here does not include any papers that share the fundamental aspects of the problem studied in this thesis. I therefore conclude that it is highly likely this problem for metro systems has not previously been studied in any published English research literature.

3.3 MDPVRP relevance in transport systems

In this section I move beyond metro system as an application area and explore other transport systems. The goal is to determine if literature on comparable problems are relevant to the MDPVRP studied in this thesis. Operations research covers a variety of different application areas. It is generally easier to find literature within the same application area, as opposed to finding problems with similar structure or methodology from different application areas. Papers tend to cite relevant articles from the same application area, and only rarely cite other application areas with similar problem structure and methodology. This seems to happen because researchers focus on different problems within a application area more often than focusing on particular problem structures and methodologies across different application areas. This leads to diverging use of terms for otherwise similar concepts making it harder to navigate the literature.

For a transport system to be relevant, the version of the MDPVRP covered in this thesis must be applicable and appropriate. I have identified the following two requirements to determine if the MDPVRP is relevant to a transport system:

- **Nontrivial distance** - The typical deadheading distance between depots and terminal stations of a block is more than trivial or nonexistent
- **Reasonable alternative depots** - There are multiple reasonable choices of depots per terminal station

If we first consider the airline industry, most airports store airplanes at hangars co-located with the airports. The distance from the gates to the hangars at an airport is fairly short for the typical case, violating the requirement of nontrivial distance. In some rare cases planes may be re-positioned to other airports without — or with unprofitable few — passengers because the planes are needed there the next morning. But this "deadheading" trip is unavoidable and would have happened early in the morning if not the previous night as necessitated by the flight schedule. The hangars at the new airport are still co-located with that airport.

Long-distance railway is an example of violation of the requirement for reasonable alternative depots as each terminal station usually have only one train depot available in the vicinity. Deadheading to other terminal stations is usually not a real alternative as the distance is too long. Long-distance shipping, and trucking share the same characteristics as long-distance railway on this point. There is a rich body of operations research literature concerning the airline industry, long-distance railway, shipping and trucking, however, they usually do not met both requirements of nontrivial distance and reasonable alternative depots and hence this literature is not relevant to this thesis.

Table 3.2 – Summary of metro research documents. Planning steps from Figure 3.4. "Opportunity cost" in a objective function are ways of calculating societal costs like passengers time spent waiting and traveling. "Open blocks" in the Additional properties-column means locked blocks are not taken as input.

Document	Planning step	Objective function	Depot	Periodic	Additional properties
Freling et al. (2003)	Vehicle scheduling Crew scheduling	Min number of vehicles and crew	Single	No	Open blocks.
Huisman et al. (2005)	Vehicle scheduling Crew scheduling	Min sum of vehicle and crew costs	Multi	No	Vehicle and crew locked to depot. Open blocks.
Cadarso et al. (2013)	Disruption management	Min sum of operating and opportunity cost	Multi	N/A	Open blocks.
Ramos (2013)	Network planning Line planning	Min sum of passenger cost and operator cost	Multi	N/A	
Cadarso and Marín (2014)	Disruption management	Min sum of operating and opportunity cost	Multi	N/A	Open blocks.
Jiang et al. (2014)	Timetabling Vehicle scheduling	Min travel time and min plan deviation	Single	No	
Fuentes et al. (2015)	Crew scheduling	Min operating cost	Multi	No	
Cadarso et al. (2015)	Disruption management	Min sum of operating and opportunity cost	Multi	No	
HassanNayebi et al. (2016)	Timetabling	Min wait time and opportunity cost	N/A	No	Robust stochastic programming.
Liu et al. (2017)	Line planning Timetabling Vehicle scheduling	Min sum of operating and opportunity cost	N/A	No	Multiple vehicle types.
Laporte et al. (2017)	Timetabling Vehicle scheduling	Min operating, capital and opportunity costs	Single	No	
Ding (2018)	Network planning	Max social surplus	N/A	N/A	
Fonseca et al. (2018)	Timetabling Vehicle scheduling	Min sum of operating and transfer cost	Multi	No	Vehicles locked to depot.
Zhang et al. (2018)	Line planning	Min timetable deviation and train size	Multi	No	
Ortega et al. (2018)	Disruption management	Min sum of operating and opportunity cost	Single	No	
Zhou et al. (2018)	Timetabling	Min net energy consumption	N/A	No	
HassanNayebi et al. (2018)	Timetabling	Min wait time	N/A	No	
Fuentes et al. (2019)	Crew scheduling	Min operating cost	N/A	No	
Chang et al. (2019)	Disruption management	Min delays	Single	No	
Blanco et al. (2020)	Line planning Timetabling	Min operating cost	N/A	No	
Huang et al. (2020)	Disruption management	Min sum of delay cost and opportunity cost	Single	No	

In fact, most metro systems do not fulfill the requirement of reasonable alternative depots. Metro lines are usually constructed independently reducing the number of alternative depots available to trains on a particular line. The Prague Metro is an illustrative example as seen on the map in Figure 3.5. The three lines apparently meet at the station *Můstek*, *Florenc* and *Muzeum*, but the rails and platforms are not common between the lines. The separate line platforms are merely co-located in the same station building. Each line has its own set of rolling stock and often just one or two depots per line, making the problem of depot allocation trivial. Compare this to the map of Oslo Metro in Figure 2.2 where all lines share rails and platforms in the city centre with five depots spread out in the network. It follows that the MDPVRP is only a significant problem for metro systems with a interconnected network design like one at the Oslo Metro.

Figure 3.5 – The Prague Metro schematic map. Stations Můstek, Florenc and Muzeum in the city center are shared by two lines, but the rail infrastructure does not overlap. Copyright Adam Sporka

City bus networks — also known as bus rapid transit (BRT) when sufficiently developed and of a certain size — are similar to metros as they are highly integrated and have a large number of stops, lines, depots and vehicles. BRT systems typically include dedicated roadways and high capacity vehicles designed to reduce stop time by having passengers rapidly board and disembark. Both requirements of nontrivial distance and reasonable alternative depots are met for BRT systems. Thus, minimizing deadheading in BRT systems can represent huge cost savings in large cities. For instance, Nasibov et al. (2013) report that the BRT system in Izmir, the third largest city of Turkey with a population of about 4.3 million inhabitants in the province (Wikipedia, 2020a), has 1,424 buses that operate on 293 routes using 10 depots. The total daily deadheading distance at the time of the study was computed to 16,851 km, or nearly halfway around the world. Mathirajan et al. (2010) deal with an even larger bus-depot matching problem in Bangalore, India with 5,031 buses and 30 depots. A 2018 US report found that 14.5 percent, or 582.1 million driven bus kilometers a year, corresponds to deadheading trips (APTA, 2020). There are important differences between BRT and metro systems, like the operating costs per vehicle and the route availability in the network, but no other transport system is more similar to the metro than BRT. Furthermore, contrary to metro systems, there is a rich literature on depot routing problems for BRT systems.

To identify relevant literature on BRT systems, I used a similar approach to that of metro systems and searched Google Scholar (2020) using the following search string:

"optimization" "bus rapid transit" "deadheading"

When not including patents and citations the search provided 152 results on April 21st 2020. These results were evaluated based on their title and abstract and rejected if they did not meet the following criteria:

1. The document is available in English through my university library
2. It covers a problem from the discipline of operations research
3. The problem concerns matching of any combination of block, bus and depots with an objective of minimizing deadheading

After this filtering, 17 unique documents remained, all papers published in a journal. The publication year is in the range of 1983 and 2020 with 4 papers, or 24 percent, published in 2018 or later as compared to 48 percent of those regarding metro systems. In fact a majority of the papers on BRTs are published before 2003, the earliest publication year for research on metro systems in Table 3.2. See Table 3.3 for a similar overview of the BRT papers, but note that all BRT papers concern a VRP and are more relevant to the problem of this thesis than any research document listed in Table 3.3.

Table 3.3 – Overview of bus rapid transit (BRT) research papers. Note that the Problem type-column is *not* comparable to the Planning step-column in Table 3.2. "Open blocks" in the Additional property-column means locked blocks are not taken as input. Each paper listed covers a multi-depot problem.

Paper	Problem type	Objective function	Periodic	Additional properties
Wilhelm and Parks (1983)	Block-depot matching and depot location and size	Min cost of operation, deadheading and construction	No	Buses handled in aggregate. Implies block-bus are locked. Open blocks
Maze et al. (1983)	Bus-depot matching and depot location and size	Min cost of operation, deadheading and construction	No	Block-bus locked
Sharma and Prakash (1986)	Block-bus-depot matching	Pri: min deadheading Sec: min deadheading range	No	Buses handled in aggregate.
Mathirajan (1987)	Block-bus-depot matching	Min deadheading	No	Heterogeneous vehicles/depots
Agrawal and Dhingra (1989)	Block-depot matching and depot location and size	Min cost of deadheading and construction	No	Buses handled in aggregate.
Mathirajan (1993)	Bus-depot matching	Min deadheading	No	Block-bus locked
Uyeno and Willoughby (1995)	Block-bus-depot matching and depot location and size	Min cost of operation, deadheading and construction	No	Buses handled in aggregate. Considers weekday, Saturday and Sunday/holiday blocks, but independently
Perre and Oudheusden (1997)	Block-bus-depot matching	Min deadheading	No	Buses handled in aggregate. Heterogeneous vehicles/depots
Willoughby (2002)	Block-bus-depot matching and depot location and size	Min cost of operation, deadheading and construction	No	Buses handled in aggregate. Considers weekday, Saturday and Sunday/holiday blocks, but independently
Mathirajan et al. (2010)	Bus-depot matching	Pri: min deadheading Sec: min deadheading range	No	Buses handled in aggregate. Block-bus locked
Kepaptsoglou et al. (2010)	Block-bus-depot matching	Min deadheading cost and depot imbalance	No	Buses handled in aggregate. Heterogeneous vehicles/depots
Nasibov et al. (2013)	Block-bus-depot matching	Min deadheading	No	Buses handled in aggregate. Heterogeneous vehicles/depots
Dávid and Krész (2018)	Block-bus-depot matching	Min travel costs and operating costs	Yes	Buses handled in aggregate. Heterogeneous vehicles
Baldoquin and Campo (2018)	Block-bus-depot matching	Pri: min deadheading Sec: min deadheading range	No	Heterogeneous vehicles. Considers weekday, Saturday and Sunday/holiday blocks, but independently. Open blocks
Xu et al. (2018)	Block-bus matching	Min travel costs	No	Departure-duration restrictions for crew shifts. Open blocks. Bus-depot locked.
Moreno et al. (2019)	Block-bus matching	Min fleet size and deadhead costs	No	Open blocks. Bus-depot locked.
Sevim et al. (2020)	Block-depot matching	Min fleet size and deadhead costs	No	Open blocks.

Even though many of the early BRT papers cover very large instances, the computing power of the day was far from adequate to solve these instances to optimality. The papers from the 80s and 90s were restricted to defining the problem and solving mock instances as a proof of concept. Later papers take advantage of the increased computing power and tackle larger problems and more complex and integrated models. Baldoquin and Campo (2018) consider a problem of minimizing deadheading and, secondary, minimizing the range between shortest and longest deadheading trips, where a heterogeneous fleet of vehicles are routed to form blocks including deadheading trips to depots. The authors consider three sets of routes, namely those from a weekday, Saturday and a Sunday/holiday timetable. However, each timetable are used as input in three independent runs, implying that the solutions are repeatable for all days of the particular type and thus independent from each other. The problem in this thesis extends on the two-way matching of vehicles to depots and blocks by introducing periodic cleaning constraints forcing dependencies between days, even those of equal type.

Unlike the research literature on metro systems, almost all papers identified in Table 3.3 include minimizing of deadheading in their objective function. Some include other costs as well as deadheading, like Kepaptsoglou et al. (2010) which also aims for reducing depot utilization imbalance. Some papers assume buses belong to a specific depot and cannot change depot. Locking buses to depots reduces the problem to a block-depot matching problem rather than a double matching problem of matching vehicles to both depots and blocks, like the problem studied in this thesis.

Another dissimilar factor for most of the BRT papers is considering vehicles in aggregate. Only the *number* of vehicles going between a certain depot and a given terminal station are recorded, not single identifiable vehicles. This aggregation is also utilized in Dávid and Krész (2018), which is the only paper identified to present a periodic model. The periodicity is introduced to include vehicle maintenance at least once every three days in service. Although the problem from Dávid and Krész (2018) is the most similar to the one studied in this thesis, it is modelled as a commodity flow model and not as a matching problem. All allowable combinations of locations (garages, blocks and maintenance facilities), days and inspection states are represented by state nodes with edges covering the allowable transitions between states. The standard commodity flow model is augmented by adding capacity restrictions on depots and maintenance stations as well as constraints on blocks to ensure demand is met. They also allow for vehicles to cover more than one block per day, blurring the demarcation to the VSP compared to the problem studied in this thesis.

Dávid and Krész test this model on a real life instance of 238 vehicles, 109 garages, 6 maintenance locations and an average of 131 daily blocks. They vary the length of maintenance periods, vehicle types and days, but all scenarios are solved to optimality or near optimality in about 40 to 82,000 seconds on a standard desktop computer. The longest planning horizon the authors test is 3-weeks which provided optimal or almost optimal (<1 % duality gap) results within 16,000 to 82,000 seconds, which the authors deemed promising considering the practical application of this planning model. They do not comment on the length of planning horizons in the practical applications they envision, but the needed planning horizon at the Oslo Metro is months to a year rather than weeks. Given the solution time reported by the authors (rendered in Figure 3.6) it seem rather optimistic to conclude that this approach may solve similar or larger instances for planning horizons of months to a year. Each scenario S-2 to S-6 in the figure corresponds to instances with two vehicle types and 2 to 6 days as the maximum allowed time between maintenance checks per vehicle. The solution time clearly

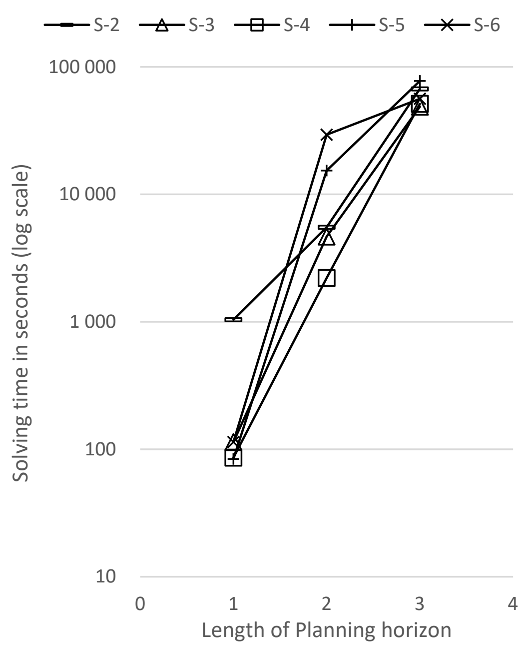

Figure 3.6 – Run time data from Dávid and Krész (2018, p. 7) in seconds on a logarithmic scale — starting on 10 — for each scenario per 1, 2 and 3 weeks of planning horizon.

increase exponentially with at least an order of magnitude on average per one week increase in planning horizon across all tested scenarios. Although it remains to be tested, an extrapolation from this data suggest that a planning horizon of 12 weeks would require 10-100 trillion seconds to solve, or about 300-3,000 millennia. Even with adjustments for the extrapolation, increased computing power and streamlined algorithms, it seems highly unlikely to expect exact solutions within reasonable time for long planning horizons using this method.

The state based commodity flow model presented in Dávid and Krész (2018) is an interesting modeling alternative to the approach given in this thesis. Still, it does not solve the challenge of superlinear growth in computing time with increased planning horizon. One way of achieving adequate solutions in reasonable time is utilizing heuristics to decrease solution time, which is introduced in the next section.

3.4 Literature on the Rolling Horizon Heuristic

This section provides a short introduction to the Rolling Horizon Heuristic (RHH) and a brief review of the literature behind the approach, with an emphasis on use cases similar to the one in this thesis. A heuristic approach is needed for the MDPVRP because it is NP-hard and solving realistic instances of the problem with long planning horizons is likely to be impractical, as indicated with the run time data from from Dávid and Krész (2018) shown in Figure 3.6. A rolling horizon heuristic is chosen because it decomposes the problem across time, the dimension extended in long planning horizon instances. A technical explanation of the implementation of the RHH used in this thesis is provided in Section 6.1.

The RHH is a matheuristic, a type of heuristics which make use of mathematical programming models to find solutions more quickly (Archetti & Speranza, 2014). As with all heuristics, this solution method is not guaranteed to find the optimal feasible solution. It is in fact not even guaranteed to find any feasible solution even if they exist (Uggen et al., 2013). A RHH decomposes the full problem by dividing the planning horizon into a set of shorter disjoint sub-horizons, solving each iteratively by fixing the variables in previously solved sub-horizons when solving subsequent sub-horizons. For long horizon integer programming (IP) problems solved using B&B, the solution time exceeds the solution time needed for a sequence of short horizon problems with a combined length equal to the long horizon problem. When using a RHH, increasing the planning horizon corresponds to adding more short horizon problems to the sequence, which in theory only produce a linear increase in solution time. If the variables within a sub-horizon are dependent on variables in other sub-horizons, the combined solution from a RHH may prove worse than the global optimum. This is the reason this heuristic approach does not guarantee an optimal solution. Longer sub-horizons are less likely to fail in finding feasible solutions, but they increase the total solution time.

Archetti and Speranza's (2014) survey on matheuristics identifies RHH as a decompositional approach. The RHH decomposes the original IP problem across the time dimension into shorter and easier IP subproblems, solves them sequentially and reconstructs feasible solutions from the solutions of each subproblem. This approach is therefore dependent on the ability of a problem to be decomposed based on a division across time, and to properly integrate the solution to subproblems. Furthermore, the subproblems must be more than proportionally easier to solve than the original problem since the RHH must solve a number of these problems sequentially. In lieu of any of these properties the RHHs becomes an inefficient solution method.

Only one of the research documents in Table 3.3 studies a periodic problem, but with an exact solution method and no heuristics. Most research utilizing the RHH is found in manufacturing scheduling (Rakke et al., 2011) where it proves an effective way of solving planning problems with long time horizons. Quick solution methods are paramount in manufacturing where more reliable and recent data becomes available and frequently demand updated schedules. Baker (1977) is one of the very earliest explorations of a RHH approach in manufacturing scheduling and found it effective, yet sensitive to the problem structure and parameter choices like the length of each period. Baker also found that the RHH is especially useful if plans may be revised when gaining new information.

De Araujo et al. (2007) provide a more novel example of utilizing rolling horizon to solve large mixed-integer programming (MIP) problems in manufacturing. The authors employs a version of the RHH called *relax-and-fix* which considers all variables when solving each subproblem, but most variables are LP relaxed. Linear programming (LP) relaxation of integer variables involves removing the integer restriction, while keeping any upper or lower variable bounds. The full planning horizon is divided into K sub-horizons, one for each subproblem. With each iteration a central period is defined over a sub-horizon. In iteration 1, the central period covers the first sub-horizon, and in iteration 2 it covers the second and so on, each time rolling the horizon one step forward. The central period retains the original model formulation, but for the remainder of the planning horizon the model is relaxed. This relaxed period beyond the central period *forecast* the consequences of decisions in the central period to avoid short sighted solutions. In the first iteration the subproblem consists of a central period covering the first sub-horizon with integer and binary variables, and LP relaxed variables in each subsequent sub-horizon. This relaxed problem is solved, and the decisions from the central period is then fixed before moving on to the next iteration and the next sub-horizon. The successive iterations solve a partially fixed problem, each with a greater number of fixed variables and fewer LP relaxed variables than the one before. In the end the forecast period shrinks to nothing as all variables are fixed and a solution to the full problem remains. See the solid literature review on the relax-and-fix in Uggen et al. (2013) for more background on this heuristic.

The version of the RHH used in this thesis is a fixed length forecasting relax-and-fix approach. The central period in this version is piloted by a forecast period which only extend some d time units from the end of the central period, and not all the way to the end of the planning horizon like in de Araujo et al. (2007). The variables in the section beyond the forecast period is simply not included when solving each subproblem. There are other ways of relaxing the problem in the forecast period, for example by excluding certain constraints or replacing the model with a simplified formulation (Mohammadi et al., 2010). In this thesis I have opted for a LP relaxation as it seems to be the most used method. Both Alonso et al. (2000) and Marín (2006) uses a relax-and-fix approach with LP relaxed forecasting periods to solve problems within transportation, but for air traffic control and airplane taxi planning, respectively, and not related to rail transport. To my knowledge, mainly because few papers considers periodic vehicle matching problems at all, no one have used the relax-and-fix or other versions of the RHH to solve a block-vehicle-depot matching problem.

The relax-and-fix approach with a fixed length forecasting period is also used in Rakke et al. (2011), but for a problem in shipping of liquefied natural gas. The authors combine the RHH with an improvement heuristic. Improvement heuristics can utilize the relax-and-fix structure to improve the solution by un-fixing parts of the solution found by the RHH and re-solve to uncover better solutions. During each pass of the improvement heuristic different constraints are fixed and un-fixed, often across other dimensions than time to capture dependencies not found in the chronological RHH approach. This way, new and better solutions may be found. This assumes a feasible solution is found using relax-and-fix, which is not guaranteed in the basic algorithm (Uggen et al., 2013, p.360).

In this thesis I explore a fixed length relax-and-fix RHH and compare it to an exact solution methods. The RHH in this thesis is combined with solution space reducing heuristics, which are presented in Section 6.2.

3.5 Summary and contributions

This chapter have reviewed the literature relevant to this thesis. The assessed literature touch upon three main aspects: First, the problem of thesis — the MDPVRP — with its background, variants, alternative formulations and why it is relevant for the Metro. Secondly, the application area of metro systems and how they compare to other transport systems, with an emphasis on the literature from the similar BRT systems. And finally, on the proposed heuristic solution method.

The MDPVRP in this thesis is novel in that the routing of vehicles between depots and terminal stations to minimize deadheading have not been previously studied for metro systems, nor has the periodic approach that arise with the multi-day cleaning constraints. Among transportation systems, it seems that metro systems particularly understudied. A central contribution of this thesis is formulating a model for the MDVRP based on the sequence of double matching problems discussed in 3.1 tailored to the situation at the Metro. Besides the commodity flow model of the similar problem proposed by Dávid and Krész (2018), I have not identified any previous work on the depot-block routing of metro trains.

The MDVRP is NP-hard (Bertossi et al., 1987), and because the MDPVRP is an extension of this problem, it is too. Thus, it is difficult to solve realistic instances over long planning horizons to optimality within reasonable time. This is also noted with the performance of the solution method proposed by Dávid and Krész (2018). I propose a heuristic solution method to the IP model for long planning horizons, and compare computational time and solution quality to an exact solution method. To my knowledge, this has not been done before for depot-block routing problems of any application area.

4 Problem Description

This thesis tackles the optimal matching of rolling stock to overnight depots and timetabled blocks at the Oslo Metro while fulfilling short-term cleaning requirements. Finding the optimal matching plan entails minimizing the total mileage of deadheading from depots to the first station of a block each train is covering that day, and from the final station of the block back to a depot in the evening. This chapter provides a description of this problem identified as a version of the Multi-Depot Periodic Vehicle Routing Problem.

The Metro provides the backbone of public transport in and out of the Oslo city centre with a homogeneous fleet of trains servicing an interconnected set of lines. The rail network consist of a set of stations and depots connected by rails and junctions such that any train can service any station and use any depot. The minimum distances between stations and depots are known. The distance traversed while plying a block is not influenced by which depots the trains are assigned.

Sporveien has defined timetables for all metro lines and within a day of operation each timetable is divided into a number of blocks where each block consists of a set of train movements to ensure that the timetable is covered. For instance could such a set of movements be: Two coupled trains start at Vestli station at 06:30, serve Line 4 and arrive at Bergkrystallen by 07:15, then turn around and at 07:25 serve the same line back to Vestli, and repeat. A block often involves trains plying the same metro line throughout the day, but sometimes they move to other lines during the day as demand changes. Each block has a defined start time and station in the morning, with a corresponding end time and station in the evening.

Every day trains are assigned a block for that day. Each block is served by either one train or two coupled trains. For blocks served by two coupled trains, both trains must originate from the same depot as the coupling is performed at the depot. As the timetable is different on the weekends, there are separate blocks for Saturdays and Sundays. Public holidays uses the Sunday timetable. Because some redundancy is needed for long term maintenance and unexpected breakdowns, Sporveien have more trains available than necessary to cover all blocks, especially on Saturdays and Sundays/holidays. The trains must deadhead in the morning from the depot where they were stored to the first station of their allotted block, and conversely they deadhead from the final station of the block to a depot for storing and maintenance the following night. Deadheading trips are costly without directly serving any customers, and therefore Sporveien aims to minimize this cost by minimizing the total length of deadheading trips. The total deadheading length is minimized by finding the optimal combination of start depot, block and end depot for each train each day over a planning horizon.

The depots are spread out in the rail network and are accessible from any metro line. All depots have a maximum storing capacity which cannot be exceeded. Some depots are part of the main track or a station and have restricted access during operating hours. There are certain non-compatible combinations of depots and blocks as the blocks start or end at a time when the depots are unavailable. The trains serving these blocks must therefore be matched with other block-compatible depots.

All trains are required to undergo an interior and exterior cleaning at least once within a given number of operating days. Cleaning is performed after hours at depots with the appropriate equipment, which only a subset of the depots have. If a depot is equipped with both interior and exterior cleaning equipment trains stored here can be cleaned both internally and externally during the same night. Cleaning capacity during a single night is only limited by the total storing capacity at a depot, but exterior cleaning is not performed during the night before Saturdays or Sundays/Holidays due to collective labor agreements.

This problem formulation is highly adapted to the current situation at the Oslo Metro aiming for realistic accuracy and practical usability. Existing infrastructure and timetables are fixed and assumed deterministic. Extraordinary cleaning needs, accidents, breakdowns or other unforeseen events are not considered. Assuring conflict free traffic control of trains in the network is also outside the scope of this thesis, and all trains are assumed to traverse the minimum distance between depots and stations without disruption. A final simplification is assuming the propagation time in and out of the depots is instantaneous.

To summarize, the objective is to minimize the total deadheading length traversed between depots and blocks by all trains over all days in the planning period. The deadheading length is solely dependent on the distance between chosen depots and the first and final stations of matched blocks. The problem is thus to find the optimal allocation of trains to depots and blocks, each day in the planning period while satisfying depot capacities, flow consistency constraints and cleaning requirements of trains' interior and exterior.

5 Mathematical Model

This chapter presents a mathematical formulation of the Multi-Depot Periodic Vehicle Routing Problem (MD-PVRP) as described in Chapter 4. Section 5.1 provides a description of the modelling assumptions and Section 5.2 presents the notation used. In Section 5.3 the full model is formulated and explained.

5.1 Model assumptions

As mentioned in Chapter 2, the whole of Stortinget depot and parts of Ellingsrudåsen and Vestli depots are unavailable during operating hours. These depots overlap with the main track or platforms at the nearby station. The depots at Ellingsrudåsen and Vestli are therefore modeled as two separate depots, one part is unavailable during operating hours and the other is always fully available. See Table B.1 in Appendix B for details on how these are modeled. The constrained parts of the depots are not compatible with blocks starting or ending within the same time period. To account for this, subsets of all blocks compatible with each depot are defined. See set notations B^x_n and B^y_m in Table 5.1.

5.2 Model definitions

In this section I present notation used in the model formulation of the MDPVRP studied in this thesis. The notation is summarized in Table 5.1 for model sets and subsets, Table 5.2 lists the parameters and Table 5.3 shows the model variables and associated domains.

There are four main sets: the set of trains T, the set of days D, the set of depots N and the set of blocks B. Each set is indexed by the non-capitalized version of the set letter, with m being a secondary index to depots N. There are no defined subsets of trains T since all trains are homogeneous. D is partitioned into the subsets weekdays D^W, Saturdays D^S and Sundays/holidays D^H. Of depots N, there are two overlapping subsets, N^I and N^E, which are depots with cleaning equipment for interior cleaning and exterior cleaning, respectively. The set B has several defined, overlapping subsets: B^{double} contains all blocks that require two coupled trains. B_d are the subsets of blocks compatible with day d such that if day 1 is a weekday, B_1 contains all blocks from the weekday timetable. B^x_n and B^y_m are the subsets of blocks compatible with starting at depot n and ending in depot m, respectively. B^x_{AVL} would for instance be the subset of blocks compatible with starting the day from the *Avløs* (AVL) depot. β^0 is a single element representing train out of service, e.g. not taking any timetabled blocks. This element is part of each block subset, except B^{double}.

Table 5.1 – Tabular overview of set notation.

Notation	Description
T	Set of trains, indexed by t
D	Set of days in the planning period, indexed by d. $D = D^W \cup D^S \cup D^H$
D^W	Subset of days labeled weekdays
D^S	Subset of days labeled Saturdays
D^H	Subset of days labeled Sundays and holidays. $\emptyset = D^W \cap D^S \cap D^H$
N	Set of depots, indexed by n, m
N^I	Subset of depots with interior cleaning equipment
N^E	Subset of depots with exterior cleaning equipment
B	Set of blocks, indexed by b
B^{double}	Subset of B containing only blocks that require two coupled trains
B_d	Subsets of B, each compatible with day d
B^x_n	Subsets of B, each compatible with starting at depot n
B^y_m	Subsets of B, each compatible with ending at depot m
β^0	Mock element representing train out of service

The parameters presented in Table 5.2 covers information needed for uniquely defining an instance of the problem studied in this thesis. S_n denotes the maximum storing capacity in number of trains at depot n. The cost parameters C^x_{nb} and C^y_{mb} are the deadheading distance traveled from depot n to the start of block b and from the end of block b to a depot m. P_t is the depot where train t is stored at the start of day 1 in the planning horizon. The number of trains needed for servicing a block b is denoted by A_b. Exterior cleaning is regulated by parameters Q^E_t and $\mathscr{E}$, where the former is the initial cleaning history of train t — e.g. how many days train t have gone without exterior cleaning before day 1 — and the latter is the number of consecutive days all trains must have their exterior cleaned at least once.

Table 5.2 – Tabular overview of parameter notation.

Notation	Description
S_n	Maximum storing capacity at depot n
C^x_{nb}	Distance traveled from depot n to the start of block b
C^y_{mb}	Distance traveled from the end of block b to depot m
P_t	Initial position of train t
A_b	Number of trains needed for covering block b
Q^E_t	Days lapsed without exterior cleaning of train t prior to day 1
$\mathscr{E}$	Number of concecutive days for which all trains' exterior must be cleaned at least once

Decision variable notation is presented in Table 5.3. The binary variables x_{tdnb} and y_{tdbm} constructs the matching of trains to depots and blocks. x_{tdnb} equals 1 if train t at the start of day d drives from depot n to block b, and 0 if it does not. Likewise does y_{tdbm} equal 1 if train t at the end of day d drives from block b to be stored overnight at depot m. The binary variables δ_{dnb} are designed to flag if a block $b \in B^{double}$ is assigned two trains originating from the same depot. These variables equal 1 if, on a day d, a block b is served by exactly two trains from the same depot d, and 0 otherwise. Finally, the exterior cleaning variables z_{td} denote the number of days since last exterior cleaning for a train t on day d.

Table 5.3 – Tabular overview of decision variable notation.

Notation	Description
x_{tdnb}	1 if train t on day d is matched with depot n and block b in the morning, 0 otherwise
y_{tdbm}	1 if train t on day d is matched with block b and depot m in the evening, 0 otherwise
δ_{dnb}	1 if block $b \in B^{double}$ is served by two trains from depot n on day d, 0 if no trains are serving b from depot n
z_{td}	1 if train t is cleaned on day d, 0 if not

5.3 Model formulation

Objective function

$$\min z = \sum_{t\in T} \sum_{d\in D} \sum_{n\in N} \left(\sum_{b\in B_d \cap B^x_n} C^x_{nb} x_{tdnb} + \sum_{b\in B_d \cap B^y_n} C^y_{nb} y_{tdbn} \right) \tag{1}$$

The objective function (1) sums the traveled deadheading distance between all compatible pairs of depot to block and block to depot over all trains t and all days d. If a train starts the day at depot n and drives to the beginning of block b a cost C^x_{nb} is added. Conversely, a train going from the end of a block b back to to a depot n adds a travel cost C^y_{nb}. This objective function is minimized.

Constraints

$$\sum_{l\in N} \sum_{b\in B_d\cap B_m^y} y_{tdbm} = 1 \qquad t\in T,\ d\in D \qquad (2)$$

$$\sum_{t\in T} \sum_{b\in B_d\cap B_m^y} y_{tdbm} \leq S_m \qquad d\in D,\ m\in N \qquad (3)$$

$$\sum_{b\in B_1\cap B_{P_t}^x} x_{t1P_tb} = 1 \qquad t\in T \qquad (4)$$

$$\sum_{n\in N} x_{tdnb} = \sum_{m\in N} y_{tdbm} \qquad t\in T,\ d\in D,\ b\in B_d \qquad (5)$$

$$\sum_{b\in B_d\cap B_n^y} y_{tdbn} = \sum_{b\in B_{d+1}\cap B_n^x} x_{t(d+1)nb} \qquad t\in T,\ d\in D\setminus|D|,\ n\in N \qquad (6)$$

$$\sum_{t\in T} \sum_{m\in N} y_{tdbm} = A_b \qquad d\in D,\ b\in B_d\setminus\beta^0 \qquad (7)$$

$$\sum_{t\in T} x_{tdnb} = 2\delta_{dnb} \qquad d\in D,\ n\in N,\ b\in B^{double}\cap B_d\cap B_n^x \qquad (8)$$

$$\sum_{n\in N^I} \sum_{b\in B_d\cap B_n^x\setminus\beta^0} x_{tdnb} + \sum_{m\in N^I} \sum_{b\in B_d\cap B_m^y\setminus\beta^0} y_{tdbm} + \sum_{m\in N} y_{td(\beta^0)m} \geq 1 \qquad t\in T,\ d\in D \qquad (9)$$

$$z_{td} = \sum_{m\in N^E} \sum_{b\in B_d\cap B_m^y} y_{tdbm} \qquad t\in T,\ d\in D^W \qquad (10)$$

$$\sum_{d\in D^S\cap D^H} z_{td} = 0 \qquad t\in T \qquad (11)$$

$$\sum_{d'=d}^{d+\mathscr{E}-1} z_{td} \geq 1 \qquad t\in T, d\in D\setminus\{|D|-\mathscr{E}+1, |D|-\mathscr{E}+2, \ldots, |D|\} \qquad (12)$$

$$\sum_{d'=1}^{\mathscr{E}-Q_t^E+1} z_{td} \geq 1 \qquad t\in T \qquad (13)$$

$$x_{tdnb} \in \{0,1\} \qquad t\in T,\ d\in D,\ n\in N,\ b\in B_d\cap B_n^x \qquad (14)$$

$$y_{tdbm} \in \{0,1\} \qquad t\in T,\ d\in D,\ m\in N,\ b\in B_d\cap B_m^y \qquad (15)$$

$$\delta_{dnb} \in \{0,1\} \qquad d\in D,\ n\in N,\ b\in B^{double}\cap B_d\cap B_n^x \qquad (16)$$

$$z_{td} \in \{0,1\} \qquad t\in T,\ d\in D \qquad (17)$$

The set of constraints (2) ensure that exactly one block and one depot is assigned to each train, each day. Constraints (3) ensure that no depots exceed their max storing capacity. All trains must begin at their starting position on the first day of the planning period, which is ensured by constraints (4). The next two sets of constraints force flow consistency. Constraints (5) enforce that a train starting to ply a block in the morning will complete the block in the evening. Additionally, they also make sure that the set of constraints (2) defined by the evening variable y also holds for the morning variable x. If the previous set enforce consistency in blocks, the set of constraints (6) enforces consistency in depots. All trains going to a depot in the evening must leave the same depot the day after and together with constraints (3) they ensure that the maximum depot capacity also are held by x. Constraints (7) ensure that each block is assigned the correct number of trains each day. The set of constraints are defined by the evening variable y, but combined with constraints (5), it also holds for the morning variable x. The large set of constraints (8) require trains assigned to the same double block ($b\in B^{double}$) to leave from the same depot in the morning.

Interior cleaning is regulated by constraints (9) where at least one of the variables x and y must include a depot with interior cleaning capacity each day, or being out of service. This means all trains must start and/or end their day in a depot with interior cleaning equipment, or not be assigned a service block that day. This ensures no trains are allowed two consecutive in-service-days without interior cleaning. The next four sets of constraints concerns exterior cleaning, and the exterior cleaning variable z. Constraints (10) set z_{td} to 1 if a train t returns to a depot with exterior cleaning equipment $m \in N^E$ on a weekday D^W and 0 if the train does not. Exterior cleaning is not performed on Saturdays and Sundays/holidays, so constraints (11) ensure that z_{td} is zero for all non-weekdays. All trains are required to have their exterior cleaned at least once in every $\mathscr{E}$-length segment of the planning horizon, which is ensured by constraints (12). Moreover, trains start the planning horizon with a cleaning history and may need their first cleaning earlier than required by constraints (12). Constraints (13) take the individual trains' cleaning histories into account and require trains to be cleaned within the first $\mathscr{E}+1$ days minus the number of days a train have gone without cleaning prior to day 1, denoted Q_t^E.

The remaining constraints define variable domains. All decision variables are binary, as defined by constraints (14)-(17). Notice that the variables x_{tdnb} and y_{tdbm} are defined over blocks b of the respective subsets $B_d \cap B_n^x$ and $B_d \cap B_m^y$, which are the blocks for a specific day d compatible depot n and depot m, respectively.

As a final note, I will add that constraints (2) and (3) include variable y where it seems to be equally valid for x. As seen, these constraints does indeed hold for x through constraints (5) and (6). The reason for choosing y rather than x is specific to the current timetable at the Oslo Metro where fewer combinations of blocks and depots are compatible in the evening than in the morning. That is, the size of B_m^y tends to be smaller than B_n^x. Both sets of constraints (2) and (3) takes the sum of $b \in B_d \cap B_m^y$, which are fewer than if defined over over x and B_n^x. However, for instances with a shorter horizon, a different consideration dominates and implies one should opt for using the x variable in these constraints instead. The initial starting positions P_t reduce the relevant number of x_{t1nb} variables by a factor of $|P_t|$ because none of the variables where $n \neq P_t$ are included. Nevertheless, the reduction in x variables on the first day of a long planning horizon is less significant than the reduction of y variables due to the fact that $|B_m^y|$ is less than $|B_n^x|$. I therefore conclude that it is most effective to define constraints (2)-(3), (7) and (9)-(10) by y rather than x.

6 Rolling Horizon and Solution Space Reduction

This chapter presents the heuristic solution methods studied in this thesis. Section 6.1 provides a technical description of how the Rolling Horizon Heuristic (RHH) is implemented in this thesis. The challenge of symmetry is explored and various solution space reducing measures are presented in Section 6.2. The final Section 6.3 defines the four specific heuristic solution methods explored and compared in Chapter 8.

6.1 Implementation of the Rolling Horizon Heuristic

The MDPVRP is proven to be NP-hard as discussed in Section 3.1, and as demonstrated in Chapter 8, exact solution methods are not capable of solving realistic instances with long planning horizons within reasonable time. Both main variables x_{tdnb} and y_{tdbm} are defined across all main sets of trains t, days d, depots n and m and blocks b. This makes problem instances of even short planning horizons challenging to solve to optimality. Tackling long planning horizons requires the use of heuristic solution methods to produce good solutions within reasonable time. The RHH is used successfully in the literature for other IP problems where time is a prominent dimension of the problem, which this is the reason I use the approach in this thesis. The main drawback of the RHH is its myopic nature where problem dependencies across time are not taken into account, especially if the forecasting feature is weakly applied. An effective RHH implementation provides a considerable reduction in solution time compared with an exact approach.

The RHH is iterative, where each iteration builds upon the solution of the previous iteration. Only a subsection of the full planning horizon is considered in each iteration. I refer to the length of this subsection as the time window (TW). I further divide the TW into two time periods, the first being the central period (CP) containing the full model to be solved and the second being the piloting forecasting period (FoP), see Figure 6.1. The FoP extends the considered interval beyond the CP to avoid short-sighted solutions. To reduce computational effort, the FoP may be relaxed either by removing the integer restrictions or using a simplified model. In this thesis I chose to use LP-relaxed decision variables in the FoP.

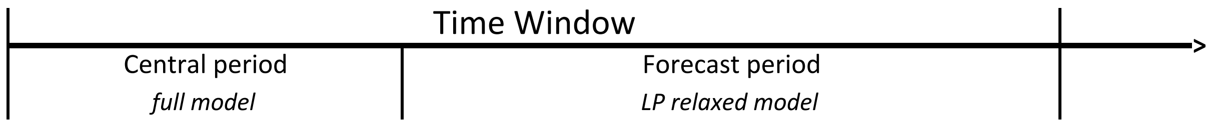

Figure 6.1 – First iteration of the Rolling Horizon Heuristic algoritm.

In the first iteration a satisfactory solution is found for the TW and the solution for variables in the CP are saved, or *frozen*, as constraints for all subsequent iterations. In each new iteration the TW is shifted a given number of time units equal to the length of CP, such that the CP is defined from the end of its previous position and the FoP from the end of the new CP, as illustrated in Figure 6.2. However, the FoP will never extend past the end of planning horizon. If the distance between the end of the CP and the end of the planning horizon is shorter than the given FoP length, the FoP is cut short for the last iterations where this is relevant.

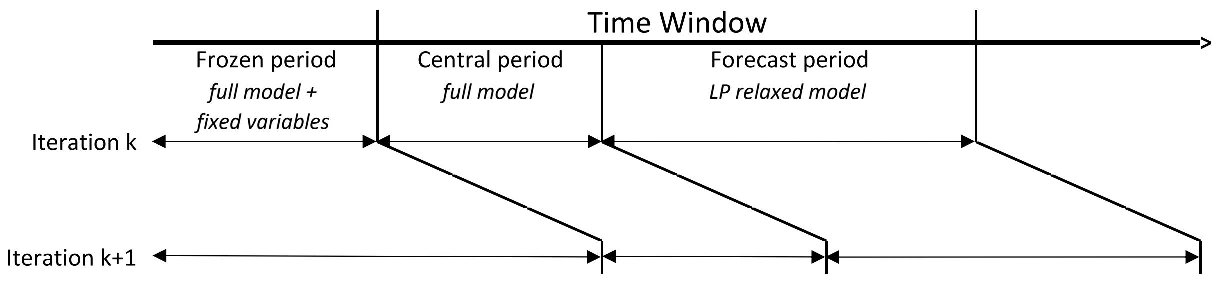

Figure 6.2 – Increments of the Rolling Horizon Heuristic. A similar figure is given in Mercé and Fontan (2003) and Rakke et al. (2011)

I call the trailing period of frozen solutions the frozen period (FzP) which integrates the solutions from each subproblem. The FzP starts out as an empty interval when $k = 1$ and accumulates fixed decision variables as each new subproblem is solved until it encapsulates the full planning horizon and provides a solution to the full problem. The implementation of the RHH in this thesis is based on Rakke et al. (2011).

The FzP is included in every iteration as this ensures flow consistency and periodic cleaning constraints are held in the transition between FzP and CP. As with the initial depot position of the trains given in the full problem, the trains depot position at the end of the FzP reduces the solution space of the x-variable at the first day in each central periods. E.g. if a train t' is fixed to depot n' on day d' at the end of the FzP, $x_{t'(d'+1)nb}$ may only equal 1 if $n = n'$ as the train must leave the same depot it arrived the day before. This reduces the solution space for the x-variable by $\frac{n-1}{n}$ for the first day of the CP in each iteration.

The FoP counteracts short-sighted solutions by basing a solution for variables in the CP on information from a larger part of the planning horizon. Fractional values are not allowed in the full model, but in the FoP it is as the purpose is to forecast the effect of the solution in CP. In later iterations, the variables in the FoP are included in the CP and thus made binary. If the dependencies between variables are strong across time, a longer FoP is needed, but this increases the solution time of each subproblem. The choice of length for the FoP is a quality/time trade-off, dependent on the problem structure. In Chapter 8, I tune the length of the FoP to find the best implementation of the RHH solution method.

6.2 Symmetry and Solution Space Reduction

The solution time for most IP problems is dependent on the solution space. Techniques that reduces the size of the solution space may therefore reduce the solution time. The challenge is, however, to avoid excluding too many good solutions in the process. In this section I will explain how symmetry and near symmetry may be exploited to reduce the solution space while still retaining good solutions. I also present another solution space reduction measure that excludes particularly costly decisions.

Symmetry in IP makes problems mathematically more difficult to solve than practically necessary. Imagine a problem with a set of homogeneous entities only separated by the index values. VRP is a typical case where homogeneous vehicles are allocated different routes in the network. Given a feasible solution, switching the routes of two vehicles v and w starting from the same depot produces a new symmetric solution with the same objective value, but which is mathematically different. All equivalent route switches involving identical vehicles creates new mathematically unique, but practically equal solutions.

The problem studied in this thesis involves a homogeneous fleet of trains, where the only initial discriminant is their starting location and cleaning history. Instances with a large fleet, few depots and short cleaning intervals will necessarily have fully identical trains. The current situation at the Metro consists of seven depots, interior cleaning at least every other day and exterior cleaning at least every five days combines to 25 states, each a unique allowed combination of these properties. A detailed calculation for this number is provided in Appendix C. The fleet of 105 trains is distributed at one of 25 states each day, which means symmetry is extensive. Moreover, the details of the situation — like the fact that all trains at the largest depot, Ryen, share the same single state — further exacerbates the symmetry.

Near symmetric solutions are also prevalent in the problem of this thesis. Four of the seven depots are modeled from two physical depots which pairwise share location. Switching compatible blocks between otherwise similar trains situated in each part of a split depot will produce an identical or very similar solution. When using branching algorithms like the B&B, computational effort is wasted when exploring isomorphic or near-isomorphic branches.

Margot (2010) proposes several strategies to mitigate symmetry, many quite effective in the right circumstances, but hard to implement in software because the overhead of checking for symmetry tend to be higher than the expected savings. Thus, the most fruitful approach is often to study the problem structure and the symmetry nature and manually implement mitigation efforts. A common strategy for dealing with symmetric formulations is adding symmetry breaking inequalities. This means adding constraints to the original formulation that cut some of the symmetric solutions, but retains at least one of the symmetric solutions. If the gains in reduced solution space is higher than the increased computing from adding the new constraints, this is a net positive alteration of the model.

Rakke et al. (2011) uses a different heuristic approach to deal with almost symmetrical solutions when implementing a RHH in a MIP problem. Their problem is similar to the one studied in this thesis as the solution space is large and populated with many almost symmetrical solutions. To decrease computational time they implement solution space reduction (SSR) by limiting the number of variables generated. In their case, they limit the contracts available to a liquefied natural gas carrying ship on a given day. The list of available contracts to start each day alternates so that no contract is unavailable to any ship all the time.

I use a SSR heuristic similar to the one implemented in Rakke et al. (2011) to decrease computational time. The main binary variables, denoted x_{tdnb} and y_{tdbm} in Chapter 5, are reduced in number. When generating the variables, an algorithm takes an integer parameter SSR^{param} as input and iterates a counter per block denoted c_b. Using the modulus operator, a variable is only produced when the remainder of $\frac{t+d+c_b}{SSR^{param}}$ equals zero. If $SSR^{param} = 3$ and train $t = 1$ only the variables for the block counter 3, 6, 9 and so on are generated on the first day $d = 1$. Basing the reduction on all three of t, d and c_b spreads it out to avoid systematic patterns that excludes all variables of a certain train, day, depot or block. When $SSR^{param} = 3$, only a third of the original variables remains.

Applying such a crude SSR, as compared to excluding variables based on a pattern adapted to the problem structure, increase the chance of removing all feasible solutions in the process. A particular concern for the problem of this thesis is limiting access to the few depots with exterior cleaning equipment. If the number of trains reaching the maximum limit of days without exterior cleaning exceeds the capacity of the exterior cleaning depot(s), no feasible solution exists. To remedy this, I make an exception to exterior cleaning depots in the SSR heuristic. All variables x_{tdnb} and y_{tdbm} where $n, m \in N^E$ are therefore always generated.

As mentioned in Section 5.1, certain combinations of depots and blocks are incompatible due to timing conflicts. All other combinations are available, even the most expensive options with regards to contributions to the objective function. If the distribution of deadheading lengths in depot-block combinations is wide, and the longest options are seldom or never included in good solutions, removing the variables that allow for these combinations might deem a reasonable way of reducing solution time without deteriorating quality. In the cost threshold (CT) solution methods I therefore do not generate variables of depot-block combinations above a certain cost threshold. As with the SSR, variables involving exterior cleaning depots overrules this strategy and are nevertheless generated.

6.3 Summary of Solution Methods

Using the heuristic techniques discussed in this chapter, I construct five unique solution methods comprising of different combinations of these techniques as well as an exact solution method. The five solution methods are summarised in Table 6.1.

The exact solution method is based on the model described in Chapter 5 and implemented without any heuristics. The second solution method is a pure RHH implementation without solution space reducing heuristics. The lengths of CP and FoP at all RHH solution methods are initially set to 2 and 4 days, respectively. The third, fourth and fifth solution methods are extensions of the second. In the third, a SSR heuristic is added to reduce the number of generated decision variables. In this solution method the parameter SSR^{param} is set to 3, which is the fraction of variables retained. The fourth solution method also reduces the solution space, but by an upper limit on the cost incurred by a certain depot-block assignment, denoted CT for short. The value of this parameter is initially set to 22.5 km. All initial parameter values mentioned in this chapter is selected based on experience from informal test runs during code development. Chapter 8 covers a more systematic parameter tuning. The last and fifth solution method is a combination of the third and fourth. Now both solution space reducing measures are introduced on top of the RHH.

Table 6.1 – Summary of solution methods tested in this thesis.

Identifier	Description
Exact	Exact solution method as described in Chapter 5. No heuristics included.
Pure RHH	Pure Rolling Horizon Heuristic, without any other heuristics.
RHH+SSR	Rolling Horizon Heuristic combined with solution space reducing heuristic.
RHH+CT	Rolling Horizon Heuristic with costly variables excluded.
RHH+SSR+CT	Rolling Horizon Heuristic with solution space reduction and costly variables excluded.

7 Data Generation

This chapter provides an explanation of how test instances in the computational study are constructed. Data specific to the Oslo Metro is provided by Sporveien in conversations and unpublished internal documents and reflect their current situation and timetables. Section 7.1 provides deadheading cost calculations and the distance matrix. Section 7.2 presents the procedure and test instances for comparing the performance of exact and heuristic solution methods. In the final Section 7.3 instances for testing the selected heuristic solution method performance on long planning horizons and other relevant scenarios are given.

7.1 Cost and distance calculations

The goal of the model presented in this thesis is to minimize deadheading length at the Oslo Metro because this is a costly, non-revenue generating activity. Cost savings is thus the ultimate motivation. To get a better understanding of the significance of the results, I estimate the cost-equivalent of a deadheading km. This cost-equivalent is estimated in this section together with the distance matrix between all relevant depots and stations at the Metro.

I present the operational cost figures provided by Sporveien in Table 7.1. In order to arrive at a cost estimate per km deadheading, I make some simplifying, but realistic assumptions. It is reasonable to consider both wear and power consumption to be linearly correlated with distance, I therefore use average cost figures as an estimate for the marginal cost. Moreover, I make an approximation that all deadheading is performed with one driver per second train. Most trains are operated coupled together two and two, but some are driven as a single train and some deadheading trips back to a depot are performed with more than two trains in a set. Therefore, with one exception detailed in the next paragraph, this approximation seems reasonable. This leads to a personnel cost per train-kilometer in normal circumstances to be $\frac{1}{2} * \frac{800\,[NOK/hr]}{30\,[km/hr]} = NOK\ 13.33$. Adding the average cost of wear and power, the total cost per train-kilometer deadheading becomes NOK 43.33, where 69 percent is contributed from wear and power costs and 31 percent from personnel costs.

Table 7.1 – Operational cost figures at the Oslo Metro as provided through conversations with representatives of Sporveien. See Section 2.4 for more details.

Attribute	Value
Avg. cost of wear and power per train	30 NOK/km
Avg. cost of personnel	800 NOK/hr
Avg. driving deadheading speed	30 km/hr
Avg. driving deadheading speed on Line 1	20 km/hr

The personnel cost is given on a per hour basis, which means increasing the driving speed lowers the personnel cost per km driven. The average driving speed is fairly constant throughout the day, but during low traffic in the early morning and late evening trains are driven at higher speeds. I assume this to be offset by the driver inconvenience allowance when working outside regular working hours. There is, however, one exception on Line 1 from Majorstuen station towards Frognerseteren. On this line segment the trains usually serve uncoupled as single trains, and with the reduced average speed of 20 km/hr. This makes the personnel cost per train-kilometer for this line segment to be $\frac{800\,[NOK/hr]}{20\,[km/hr]} = NOK\ 40$. The total cost per train-kilometer deadheading is therefore NOK 70, or 62 percent higher on the Frognerseteren branch compared to the rest of the network. To adjust for this exception, and for ease of calculation, I increase the calculated distances on this branch by 62 percent and uniformly use the cost per train-kilometer deadheading of NOK 43.33.

Distances between terminal stations and depots are given in Appendix A. The matrix is constructed from distances given in a technical drawing (Sporveien AS, 2015) of the rail network at the Metro. Some assumptions about where and how trains may turn and use junctions are necessary, and I base these assumptions on the existing infrastructure, driver schedules and understanding of where and when trains may turn while not disturbing normal traffic. The increased cost due to the time it takes to turn trains is adjusted for by adding an equivalent length to sections requiring trains to turn around. An example of a path where a train would need to turn is the distance between the depot at Avløs and the terminal station at Østerås on Line 2, which require the train to turn and switch track at Smestad station. See Figure 2.2 for reference. As mentioned above, paths using the line segment between Majorstua station and Frognerseteren terminal station are given with a 62 percent longer distance to adjust for the lower speed and single train constraints. It is worth mentioning that the deadheading between a depot and its associated station may reasonably be neglected, except the distance between Stortinget depot and Stortinget station, which require a trip to Ryen depot for preparation before storage. This distance is added to the appropriate figures involving Stortinget depot in Appendix A.

7.2 Test setup and instances for solution method comparison

The current situation at the Metro is that 105 trains are available at any given time. Depots, and their capacity and equipment is provided in Table 7.2. As mentioned in Section 5.1, the depots at Ellingsrudåsen and Vestli are modeled as two separate depots due to the difference in block compatibility. A train can maximum be in service two days without interior cleaning, and a total of five days without exterior cleaning. That is, at least once during five consecutive days, each train must have had their exterior cleaned. However, only in-service days counts towards this limit for interior cleaning. A train assigned an out-of-service block is not expected to have its interior be any dirtier than the day before.

Table 7.2 – Equipment and max capacity of trains at depots for the Oslo Metro. Depots with interior or exterior cleaning equipment marked by an "X" in the appropriate column.

Depot (shorthand)	**Max capacity**	**Interior cleaning**	**Exterior cleaning**
Avløs (AVL)	20	X	
Ellingsrudåsen-A (ELÅA)	4		
Ellingsrudåsen-B (ELÅB)	4		
Ryen (RYV)	60	X	X
Stortinget (STTD)	22	X	
Vestli-A (VESA)	6		
Vestli-D (VESD)	4		
Sum	120	3	1

Part of the depots at Ellingsrudåsen and Vestli, and the complete depot at Stortinget overlaps with the corresponding track or station and cannot be used for storing during operating hours for the given station. Before the first train arrives at the station, track and platforms must be cleared and, conversely, it is only available the following night after the last train in traffic has left the station. The storing capacity of the original depots of Ellingsrudåsen and Vestli are split between the modeled part depots, but the travel distance relative to other depots and stations stay the same.

Sporveien's current timetable forms the basis for all blocks used. Recall that a block is the schedule a train follows during a day in service. A train may serve different lines during a single day and it may be involved in one or more coupling or decoupling actions to fulfill the requirements of a given block. Sporveien provided a *driver* schedule listing all drivers' required tasks on weekdays, Saturdays and Sundays. This schedule lists where the drivers are supposed to start the day, time and place for all individual trips and the coupling or decoupling actions. Implied in the driver schedules are the movement of all trains. I am therefore able to construct a complete block schedule based on trains rather than drivers since each driver may drive one or two trains, and couple and decouple sets throughout the day. This new schedule includes a list of unique train blocks, with type of day, the time when the first and last stations are served, and lastly how many trains are required for this particular block.

From this train block schedule a set of all blocks is generated, with subsets grouping blocks by type of day and subsets representing compatibility with depots based on the availability of the depots and the time for the first and last station in a block. The subset of blocks requiring two trains is also generated, as well as the list providing the number of required trains per block.

I randomly generate the initial train positions and exterior cleaning history to avoid bias and achieve realistic start conditions. The initial positions are drawn from a lot containing all possible depot spots such that no depot exceeds its capacity. The exterior cleaning history is generated in a similar way. If not otherwise stated, all instances in this thesis start on a Monday. Exterior cleaning is only performed during weekdays, and the maximum number of days allowed between each exterior cleaning is five. Therefore, each train starts the planning horizon having its latest exterior cleaning done 2, 3 or 4 days prior. As a reasonable simplification, the initial exterior cleaning history is drawn uniformly from the set $\{2, 3, 4\}$.

These are the basic input data reflecting the current situation at the Metro. In most test instances that follow, only the number and type of days in the planning horizon varies. If not stated otherwise, all the data mentioned above are equal between all instances.

A main contribution of this thesis is the development of an effective heuristic solution method to solve large, realistic instances of the MDPVRP for the Metro. To evaluate the performance of a heuristic approach I propose a four stage testing scheme, displayed in Figure 7.1. In Stage I, a pure version of the RHH, without any additional heuristics, is compared with an exact solution method in four realistic instances with short planning horizons. Secondly, in Stage II, the pure RHH solution method is compared to the performance of extended RHH solution methods. The best contender of these is then tuned in Stage III before tested further on long planning horizon test instances and other relevant scenarios in Stage IV, as explained in Section 7.3

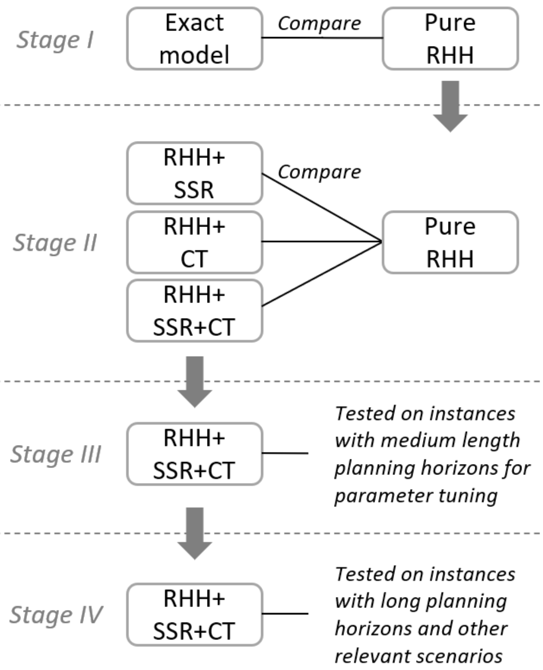

Figure 7.1 – Four stage performance testing scheme of solution methods.

To determine the effectiveness of RHH compared to the exact solution method, as per Stage I in Figure 7.1, four instances are used for a comparison between the two methods. The two first instances consist simply of three and five weekdays. The last two instances also contains non-weekdays and simulates a normal week and a two week horizon with the second Monday being a public holiday. Smaller instances are used to get meaningful results from the exact solution method within realistic solution time. See Table 7.3 for an overview of the number and type of days in each of these test instances.

Table 7.3 – Test instances for comparing the exact and pure RHH solution methods. Corresponds to Stage I in Figure 7.1.

Test instance	Days	Weekdays	Saturdays	Sundays/holidays
3D(3w-0s-0h)	3	3	0	0
5D(5w-0s-0h)	5	5	0	0
1W(5w-1s-1h)	7	5	1	1
x2W(9w-2s-3h)	14	9	2	3

In Stage II, three extended heuristic solution methods are compared to the pure RHH to determine which has the best quality/time trade-off. To compare the performance of these four methods, they are tested on four instances of medium-length. Table 7.4 provides an overview of these test instances. The first two instances are the same as the last two in Table 7.3. The third and fourth instance are three and four normal weeks, respectively.

Table 7.4 – Test instances for a comparison of four heuristic solution methods: Pure RHH, RHH+CT, RHH+SSR, RHH+CT+SSR. Corresponds to Stage II in Figure 7.1.

Test instance	Days	Weekdays	Saturdays	Sundays/holidays
1W(5w-1s-1h)	7	5	1	1
x2W(9w-2s-3h)	14	9	2	3
3W(15w-3s-3h)	21	15	3	3
4W(20w-4s-4h)	28	20	4	4

7.3 Test Instances for long planning horizons

Stage III concerns tuning the parameters of the RHH+CT+SSR solution method for decreasing computational time without substantial degradation of solution quality. The five relevant parameters are:

1. The value of the solution space reducing parameter SSR^{param}
2. The value of the cost threshold (CT)
3. The length of the central period (CP)
4. The length of the forecasting period (FoP)
5. The solution quality stop criteria at each iteration

Two alternative values of each parameter are tested with the test instance x2W(9w-2s-3h), shown in Table 7.3. This test instance is chosen because it is not trivial in length, and it includes a three day interval of non-weekdays. Thus, the solution methods are tested for performance at multi-iteration instances and longer intervals of consecutive non-weekdays. Each parameter is changed sequentially holding the other constant at either the original value, or a new better value if one is identified. Table 7.5 shows for which values each parameter is tested, and the order the parameters are tuned.

Table 7.5 – Test instances for tuning the parameters of RHH+CT+SSR. Corresponds to Stage III in Figure 7.1.

Parameter	Initial value	Alternative value A	Alternative value B
Value of SSR^{param}	3	4	2
Value of cost threshold (CT)	2250	2000	2500
Length of central period (CP)	2	1	3
Length of forecasting period (FoP)	4	3	5
UB-LB gap stop criteria (%)	2	5	1

Note that the solution quality stop criteria, the maximum UB-LB gap, is specific to each solution method. Which means that using a CT and SSR heuristic, the solver stops when a solution is found within 2 percent of the lower bound (LB) at the specific run. It is not guaranteed that successful runs produce feasible solutions within 2 percent of the optimal value of the *exact* solution method. Moreover, the solution quality stop criteria is set at each iteration of RHH involving both integer variables in the CP and LP relaxed variables in the FoP such that the gaps calculated at each iteration may not be meaningfully aggregated to the full planning horizon. I.e. a solution quality stop criteria of 2 percent for a RHH does not mean the aggregated solution is within 2 percent of the optimal solution of this heuristic solution method. It is unclear whether the aggregated solution falls within a narrower or wider gap. This means it is generally not useful to use the given UB-LB gap to compare solution quality between solution methods. For comparing solution quality, I compare the deadheading length savings relative to Sporveiens equivalent status quo solution.

Except for x2W(9w-2s-3h), the test instances in Table 7.3 and Table 7.4 are representations of generic weeks with the repeating pattern of five weekdays, a Saturday and a Sunday — or shorter planning horizons with only weekdays. These are reasonable choices when testing pseudo-realistic planning horizons of limited length, but when testing realistic, long planning horizon instances, the mix of weekdays, Saturdays and Sundays/holidays should closely simulate a real calendar. For performance testing of heuristic solution methods on long planning horizons, I construct test instances of a quarter year, half a year and a full year. The optimal length of the planning horizon is not known as Sporveien currently lack the capability for creating good plans for longer planning horizons. For the purposes of this thesis I have, based on consultations with Sporveien, decided to test instances with planning horizons of a quarter year, half a year and a full year. The Norwegian calendar for the year of 2020 is used as a basis, and the third quarter and second half of the year are used as a template for the two first instances, while the whole calendar is used for the final test instance with a long planning horizon. This is shown in Table 7.6.

Table 7.6 – Instances for testing the performance of RHH+CT+SSR on long planning horizons. Corresponds to Stage IV in Figure 7.1.

Test instance	Days	Weekdays	Saturdays	Sundays/holidays
Third quarter	92	66	13	13
Second half-year	184	131	25	28
Full year	366	254	52	60

I specifically chose the *third* quarter and *second* half-year because they contain fewer public holidays and thus are easier to handle. The Metro runs a Sunday timetable on public holidays, and on all non-weekdays no exterior cleaning is usually performed. This causes a conflict of constraints if four or five non-weekdays occur consecutively. All trains are to have their exterior cleaned once in any interval of five days, but exterior cleaning is not performed on non-weekdays. Moreover, the only depot with equipment for exterior cleaning, Ryen, has a capacity to clean at most 60 trains. Five-day intervals with one weekday and four non-weekdays lead to no feasible solutions as there is not enough capacity to clean the whole fleet of 105 trains on this single day. This shows that if a planning horizon include a five-day interval consisting of more than three non-weekdays, no solutions are feasible.

Suprisingly, this occurs only once during the year 2020. The Easter holidays on April 9th-13th are five consecutive non-weekdays: two public holidays, a Saturday, a Sunday and a new public holiday. In practice, Sporveien solves this by accepting less clean trains and allowing some exceptions to the policy of not performing exterior cleaning during the holidays. For this thesis, I modify the planning horizon for the full year instance and change April 9th and 10th to regular weekdays. I do not expect this small modification to significantly change the overall results for a full year. Moreover, this is a conservative assumption as, compared to what Sporveien does in practice, it makes the objective value worse (higher) as weekday timetables require more deadheading than the Sunday/holiday timetable.

All three instances with long planning horizons starts on a Wednesday, however, the full year instance starts on January 1st which is a public holiday. Compared to previous instances which have all started on a Monday, trains' exterior cleaning history at the start of the planning horizon are changed accordingly. For the third quarter and second half-year, trains may have been cleaned on the Tuesday, Monday or Friday prior to the Wednesday giving each train an exterior cleaning history of either 0, 1 or 4 days. With the full year instance, the situation is similar as it also starts on a Wednesday, but because it is a public holiday having a initial exterior cleaning history of 4 days is not allowed. Only the values 0 and 1 is allowed for this instance.

8 Computational Study

In this chapter, I conduct a computational study of the solution method variants presented in Chapter 6 based on the four stage testing scheme presented in Figure 7.1. The focus of the chapter is comparison of the computational time/solution quality trade-off between an exact and several heuristic solutions to instances of varying planning horizons. First, in Section 8.1, a study of the performance difference between the exact and the pure RHH solution method is conducted. Then, the four different versions of heuristic solution methods are compared to identify the best performing in terms of solution quality and computational time in Section 8.2. Finally, in Section 8.3 the best performing heuristic solution method is tuned for increased efficiency and tested on instances with long planning horizons. A study of the results in terms of performance and potential for Sporveien concludes the chapter.

All solution methods are implemented in Xpress Mosel modeling language, version 3.10.0, and solved with the commercial optimization software FICO Xpress-IVE version 1.24.08 64-bit and version 28.01.04 of Xpress Optimizer. All instances are solved using a 3.4GHz AMD Ryzen 5 2600 processor with 6 cores and 12 threads, with 16 GB RAM in 64-bit Windows 10 Home. Each run is initially set to terminate if a feasible solution of less than 2 percent difference to the best bound is found or the total run time of 36,000 seconds (10 hours) is reached. These termination criteria are based on preliminary test runs which demonstrate that most often the solution gap is the deciding criteria and the first solution found usually produce a gap of less than 2 percent. The gap termination limit is tuned in Section 8.3 and for the long planning horizon instances, also in Section 8.3, the max run time limit is extended to 86,400 seconds (24 hours).

When evaluating the solution quality of feasible solutions in mathematical programming, it is customary to report the smallest fraction, or *gap*, proven to contain the objective value of the optimal solution. The best solution found is at most this gap away from the optimal solution. The gap is calculated as: $\frac{UB-LB}{UB}$, where the upper bound (UB) — in a minimization problem — is the best feasible solution found and the lower bound (LB) is optimal solution to a relaxed version of the original problem. That is, the optimal solution to the original problem cannot be better than the optimal solution to the relaxed version of the problem, and conversely, the best solution found constitute an upper bound for the optimal solution. The B&B algorithm first solves the LP relaxed problem in the root node, which produces the first LB, and this bound is updated when traversing the B&B tree. Any feasible solutions found compete for being the UB and as soon as LB=UB the incumbent solution is proven optimal.

When evaluating solutions to the exact solution method, the optimizing software proceed as described above, and report a proven optimality gap. However, when using a RHH method we lack a relaxed form of the full problem. Only LBs of the subproblems are calculated in each iteration, and these cannot be aggregated to a LB for the full problem in any meaningful way. Therefore, when evaluating the solution quality of the RHH solutions, the LBs used are the solutions to LP relaxations of the corresponding exact solution method. These are calculated separately for practical reasons, with different initial values for the trains' starting positions and external cleaning histories. The deviation is unfortunate, but it is only significant for short planning horizons and the same LP solution is used when comparing all RHH based solution methods.

As mentioned in Section 7.3, to compare the solution quality between different solution methods, I use the relative savings compared to the current situation at Sporveien. At each result I therefore report the relative savings in percentages compared to the equivalent status quo deadheading lenght.

8.1 Comparison of exact method vs pure RHH

In this section, I investigate the relative performance of a pure RHH solution method to an exact one. Four test instances with short- to medium-length planning horizons are tested. The solution time and solution quality of the exact and pure RHH solution methods are compared and contrasted with Sporveien's status quo driving plan and the corresponding LP solutions. Table 8.1 shows the results from running the test instances presented in Table 7.3.

Table 8.1 – Results from performance testing of the exact method and a pure RHH solution method for test instances presented in Table 7.3. Corresponds to Stage I in Figure 7.1. A LP relaxed model is solved separately to calculate the integrality gap to the pure RHH full horizon solutions. Status quo deadheading lengths are estimated in Section 2.4.

Test instance		**Exact**	**Pure RHH**
3D(3w-0s-0h)	Deadheading length (km)	5,861	5,920
	UB-LB gap (%)	0.53	N/A
	IP-LP gap (%)	0.60	1.58
	Change from status quo (%)	-22.0	-21.2
	Total solution time (s)	759	242
5D(5w-0s-0h)	Deadheading length (km)	9,699	9,843
	UB-LB gap (%)	0.00	N/A
	IP-LP gap (%)	0.18	1.64
	Change from status quo (%)	-22.5	-21.4
	Total solution time (s)	10,449	2,371
1W(5w-1s-1h)	Deadheading length (km)	12,786	12,872
	UB-LB gap (%)	0.12	N/A
	IP-LP gap (%)	0.14	0.81
	Change from status quo (%)	-21.2	-20.7
	Total solution time (s)	27,876	1,495
x2W(9w-2s-3h)	Deadheading length (km)	No solution found	25,395
	UB-LB gap (%)	-	N/A
	IP-LP gap (%)	-	0.81
	Change from status quo (%)	-	-20.7
	Total solution time (s)	36,000	15,257

For the instances solved, the exact solution method finds solutions closer to proven optimality than pure RHH. For the 5D(5w-0s-0h) test instance the exact method even finds the optimal solution. However, this optimal solution is only 1.1 percentage points better than the best solution found by pure RHH, using the Sporveien status quo solution as a baseline. The pure RHH finds its first <2 % Gap solution four times faster than the exact solution for this instance. At the 1W(5w-1s-1h) test instance, the difference in computing time is even starker with the pure RHH solution time being 5.4 percent of solution time of the exact solution method, while the difference in savings is only 0.5 percent in favor to the exact method. Ultimately, at the longest test instance x2W(9w-2s-3h), the exact method does not find any solutions within the time limit.

There is a genuine, although expected, trade-off between solution quality and computational time between the exact method and the pure RHH. The difference in solution quality is moderate. For the three test instances the exact method saves on average 21.9 percent on status quo, while the pure RHH saves 21.1 percent on average. The difference in savings amounts to about 6,700 km or NOK 292,000 over the course of a year, which is only a small part of the total estimated deadheading length of 842,312 km under Sporveien's current plan. This extrapolated yearly savings is not to be read literally, but more akin to an order of magnitude estimate of how much worse a heuristic solution method is compared to an exact solution method. The focus of this study, however, is decreasing computing time to solve long planning horizon instances within reasonable time.

When disregarding the x2W(9w-2s-3h) instance, the pure RHH solution time is on average 20 percent that of the exact solution method, and with a larger discrepancy as planning horizons grow longer. The solution time using the exact method seems to increase exponentially with increased planning horizon. The planning horizon roughly doubles in length between 3D(3w-0s-0h) and 1W(5w-1s-1h), but the exact method solution time is 37 times longer with the second instance. With 1W(5w-1s-1h) the solution time is already nearly 8 hours, so using this exact method to solve instances with radically longer planning horizons, for months and up to a year, is practically infeasible. The pure RHH finds solutions of comparable quality in significantly shorter time and at a slower solution time increase with increased instance size. The heuristic solution method is therefore preferable to the exact method when solving instances with planning horizons longer than 1 week using a comparable computer and solution time tolerance.

It is worth noting that the pure RHH spent longer time solving the 5D(5w-0s-0h) instance than the 1W(5w-1s-1h) instance. This is likely not due to any inherent properties of the otherwise smaller instance, but rather an effect of the random generation of the initial train positions and exterior cleaning history. The commercial solver may also choose different B&B solving avenues which may provide significantly different solution times in otherwise similar instances. I expect this variance to even out with the longer test instances solved using a higher number of RHH iterations.

8.2 Comparison of four heuristic solution methods

This section constitutes Stage II of the testing scheme outlined in Figure 7.1, and concerns the performance comparison of four variants of the Rolling Horizon Heuristic (RHH) approach. The four solution methods are tested on four instances with medium-length planning horizons presented in Table 7.4. The test results are shown in Table 8.2. Overall, the same trade-off pattern between solution quality and computational time is repeated in this section, although somewhat weaker.

Table 8.2 – Results from performance testing four heuristic solution methods on test instances presented in Table 7.4. Corresponds to Stage II in Figure 7.1. The pure RHH results for the first and second instance are the same as in Table 8.1. a LP relaxed model is solved separately for calculating integrality gap for all solution methods. Status quo deadheading lengths are estimated based on the figures provided in Section 2.4.

Test instance		Pure RHH	RHH+CT	RHH+SSR	RHH+CT+SSR
1W(5w-1s-1h)	Deadheading length (km)	12,872	13,187	13,128	13,304
	IP-LP gap (%)	0.81	3.26	2.80	4.17
	Change from status quo (%)	-20.7	-18.8	-19.1	-18.0
	Total solution time (s)	1,495	1,894	600	866
x2W(9w-2s-3h)	Deadheading length (km)	25,395	25,658	26,113	26,243
	IP-LP gap (%)	1.80	2.81	4.50	4.98
	Change from status quo (%)	-19.7	-18.8	-17.4	-17.0
	Total solution time (s)	15,257	8,461	3,705	2,868
3W(15w-3s-3h)	Deadheading length (km)	38,651	39,149	39,602	39,744
	IP-LP gap (%)	1.28	2.54	3.65	4.00
	Change from status quo (%)	-20.6	-19.6	-18.7	-18.4
	Total solution time (s)	9,667	5,354	10,913	5,542
4W(20w-4s-4h)	Deadheading length (km)	51,558	51,742	53,465	53,032
	IP-LP gap (%)	1.30	1.65	4.82	4.04
	Change from status quo (%)	-20.6	-20.3	-17.7	-18.3
	Total solution time (s)	20,210	11,333	16,056	6,112

More often than not the extended RHH solution methods finish sooner than the pure RHH. On the other hand, the pure RHH finds the solutions with the best objective value at all instances. The CT variants are as fast or faster than their non-CT counterparts, while almost always producing worse solutions. A similar story may be observed for the SSR variants, however the computing time decrease is even larger than CT offers. When pairwise comparing runs of the same instances, solution methods with a CT component is on average 26.8 percent faster and methods with a SSR component is 39.0 percent faster than their counterparts.

The relative savings compared to status quo are fairly stable for each solution method across all instances, implying that instances with longer planning horizons do not have worse solution quality. However, the more complex heuristic solution methods produce solutions with poorer objective values on average. The pure RHH saves on average 20.4 percent on the status quo, RHH+CT saves 19.4 percent, RHH+SSR saves 18.2 percent and finally RHH+CT+SSR saves 17.9 percent on average. RHH+CT+SSR is the fastest solution method on average, and the solution method where solution time grows the slowest with increased instances size.

The purpose of comparing the computational time of these solution methods is determining which to use for the instances of considerably longer planning horizons. To assess which solution method is most appropriate, both the magnitude and growth rate of computational time is relevant. I have established that RHH+CT+SSR is on average the fastest solution method for the instances tested. Applying a linear regression to the run time data in Table 8.2 — excluding the x2W(9w-2s-3h) instance as it is non-regular — provides an estimate of the growth in solution time. I use these linear functions to predict the computing time for an instance with a planning horizon of a full year. Of course, extrapolating so far outside the data set is problematic, however, the exact numbers are not important. Rather it is the relative difference in computing time that lends weight as decision support for choosing the most appropriate solution method. A linear extrapolation predicts solution times for a full year instance at 74 hours for the pure RHH and RHH+SSR, 42 hours for RHH+CT and for RHH+CT+SSR only 27 hours.

The pure RHH and RHH+SSR are both slowest in measured solution time and the methods where the growth in solution time increase most quickly. The differences in magnitude and growth of solution time between the two remaining methods are smaller, and needs to be contrasted with the difference in solution quality. With the configuration tested in this thesis, RHH+CT produces on average solutions that saves 2.5 percent more deadheading than RHH+CT+SSR compared to Sporveien's status quo solution. This corresponds to an estimated monetary value of about NOK 900,000 a year, which might be substantially enough to be worth an expected 57 percent longer solution time. However, in the interest of brevity and to increase the chance of solving a full year instance within 24 hours, I choose to test RHH+CT+SSR on the long horizon instances. Finally, I will add that if Sporveien has access to substantially more computing power or can accept up to an order of magnitude longer computational times, the pure RHH produces considerably better solutions and might therefore be preferable over the other solution methods.

8.3 Performance analysis on long planning horizons

The previous section has deemed the RHH+CT+SSR solution method most appropriate for solving long horizon problem instances. This section covers the computational study of test Stages III and IV as presented in Figure 7.1. In Stage III, the parameters of RHH+CT+SSR are tuned for improving solution quality or computational time. Stage IV covers the testing of the tuned solution method on three realistic, long horizon test instances presented in Table 7.6.

Five parameters are tested at two alternative values aiming for improving either solution quality or computational time compared to the initial values, as presented in Table 7.5. The x2W(9w-2s-3h) test instance is used for tuning the parameters. Table 8.3 shows the tuning results.

Increasing the value of SSR^{param} to 4, and thus retaining only a fourth of the matching variables not involving an exterior cleaning depot, leads to no feasible solutions. Both SSR and CT reduce the solution space by excluding matching options. Infeasibility occurs when the combination of these heuristics removes too many options and no feasible solutions remain in the solution space. Decreasing the SSR^{param} value to 2 produced a better solution than the initial value, but at 2.5 times the solution time. The better objective value should be evaluated by how much deadheading is reduced compared to status quo. The solution from using the initial value reduces deadheading by 17.0 percent, while a SSR^{param} value of 2 produces a reduction of 18.2 percent. This savings difference of 1.2 percentage points is significant, but not crucial. Due to the substantial increase in solution time, I choose to keep the initial parameter value of $SSR^{param} = 3$.

Decreasing the CT to only allow for block-depot pairs with a distance of less than 20.0 km, except those involving an exterior cleaning depot, also leads to no feasible solutions. Increasing it to 25.0 km reduced the solution time by about a third, and surprisingly produced a slightly better objective value, which might be due to between-run variance. Because this is a Pareto improvement, the new value of CT = 25.0 km is retained.

The lengths of the RHH periods influence the number of iterations, the time spent solving each iteration

Table 8.3 – Results from tuning the parameter of RHH+CT+SSR. Corresponds to Stage III in Figure 7.1. Gray background indicate the selected value, which is used in all subsequent runs. A comparison of the results from the alternative values are given as a percentage in parenthesis.

Parameter		Objective value	Solution time
Value of SSR^{param}	Alt. value A - 4	Infeasible	N/A
	Initial value - 3	26,243	2,868
	Alt. value B - 2	25,846 (98.5 %)	7,421 (258.8 %)
Value of cost threshold (CT)	Alt. value A - 20.0 km	Infeasible	N/A
	Initial value - 22.5 km	26,243	2,868
	Alt. value B - 25.0 km	26,089 (99.4 %)	1,925 (67,1 %)
Length of central period (CP)	Alt. value A - 1 day	27,984 (107.3 %)	638 (33.1 %)
	Initial value - 2 days	26,089	1,925
	Alt. value B - 3 days	26,130 (100.2 %)	8,173 (424.6 %)
Length of forecasting period (FoP)	Alt. value A - 3 days	Infeasible	N/A
	Initial value - 4 days	26,089	1,925
	Alt. value B - 5 days	26,134 (100.2 %)	2,848 (147.9 %)
UB-LB gap stop criteria (%)	Alt. value A - 5	25,973 (99.6 %)	3,008 (156.3 %)
	Initial value - 2	26,089	1,925
	Alt. value B - 1	25,981 (99.6 %)	13,144 (682.8 %)

and the solution quality as more or less variables are included in each iteration. Decreasing the length of CP greatly decreases the solution time to just 33.1 percent of the solution time using the initial value. However, the objective value deteriorate from to a 17.5 percent savings with the initial parameter value to 11.5 percent savings of deadheading compared to Sporveien's status quo. This loss of a third of the deadheading savings is too large to justify the decreased computing time. I therefore choose to proceed with the initial value of CP length = 2 days. The alternative parameter values of FoP produced either infeasible solutions or a worse solution at higher solution time. Thus, the initial value of the FoP length is also retained.

The last parameter to be tuned is the UB-LB gap stop criteria, which together with the max time limit, determines when to terminate a RHH iteration. It is initially set at 2, meaning if a solution is found within two percent of the best lower bound of the optimal value, the solution is retained and the RHH increments to the next iteration. Both increasing and decreasing the parameter value to 5 and 1, respectively, provides a slight improvement in objective value of 0.4 percent. However, the solution time does surprisingly increase to 156.3 percent of the initial value with increasing the gap limit to 5, and a full 682.8 percent with a decrease to 1. None of the alternative values are worth the solution time increase.

In the final testing Stage IV, the tuned version of RHH+CT+SSR is tested on three realistic test instances of a quarter year, half-a-year and a full year length, as introduced in Table 7.6. For these instances, the max run time limit is increased to 86,400 seconds (24 hours). A longer run time limit is justified by the longer planning horizons of up to a year. Even longer run time limits might be justified for the longest planning horizons if Sporveien only need to run these once a year, although more frequent runs for replanning or scenario testing are likely use cases. The test results from RHH based solution methods are compared with an LP-relaxation of the exact method to compute an IP-LP gap, however, this is not possible for these long horizon instances. The LP-relaxed model did not finish within 86,400 seconds, even for the quarter year instance. For comparison, the LP-relaxed model for the 4W(20w-4s-4h) instance took about 30,000 seconds to complete which is six times the computing time of the LP-relaxed model for the x2W(9w-2s-3h) instance at half the size. Instead, I continue to use the deadheading reduction compared to Sporveien's status quo solution as a common point of reference and measure of solution quality. The test results from long horizon instances are shown in Table 8.4.

The RHH+CT+SSR solution method manages to produce feasible solutions to the third quarter and second half-year instances within 86,400 seconds, but not for the full year instance. Both successful solutions are of the same relative quality as the shorter instances tested in Section 8.2, when comparing to Sporveien's status quo solution. This indicates that the solution quality is likely to be similar to the solutions to the shorter

Table 8.4 – Results from testing the tuned version of RHH+CT+SSR on long horizon realistic instances presented in Table 7.6. Corresponds to Stage IV in Figure 7.1. Status quo deadheading lengths are estimated based on the figures provided in Section 2.4.

Test instance		**RHH+CT+SSR**
Third quarter	Deadheading length (km)	173,585.86
	Change from status quo (%)	-18.7 %
	Total solution time (s)	24,200
Second half-year	Deadheading length (km)	346,512.14
	Change from status quo (%)	-18.6 %
	Total solution time (s)	76,811
Full year	Deadheading length (km)	No solution found
	Change from status quo (%)	-
	Total solution time (s)	86,400

test instances, even though there is no IP-LP gap to evaluate. The second half-year solution is a reduction of 18.6 percent, or 79,296 km of deadheading — an equivalent cost reduction of NOK 3,409,713 over the planning horizon, a substantial cost saving.

As expected from the solution time on the medium horizon length instances in Table 8.2, no solutions are found for the full year instance within 86,400 seconds. The linear extrapolation based on the medium-length instances predict a solution time of 23,766 seconds for the third quarter test instance, a mere -1.8 percent deviation from the actual solution time. By contrast, the second half-year solution time was considerably higher than predicted. The solution time of the second half-year is more than three times that of the third quarter, diverging substantially from the linear ratio between solution time and planning horizon length.

To investigate this unexpected violation of the linear relationship between solution time and planning horizon length, I have plotted the run times of each individual iteration in the natural order for the third quarter and the second half-year instances in Figure 8.1. The solid and dotted lines are the 7-day moving average run times of the third quarter and second half-year, respectively.

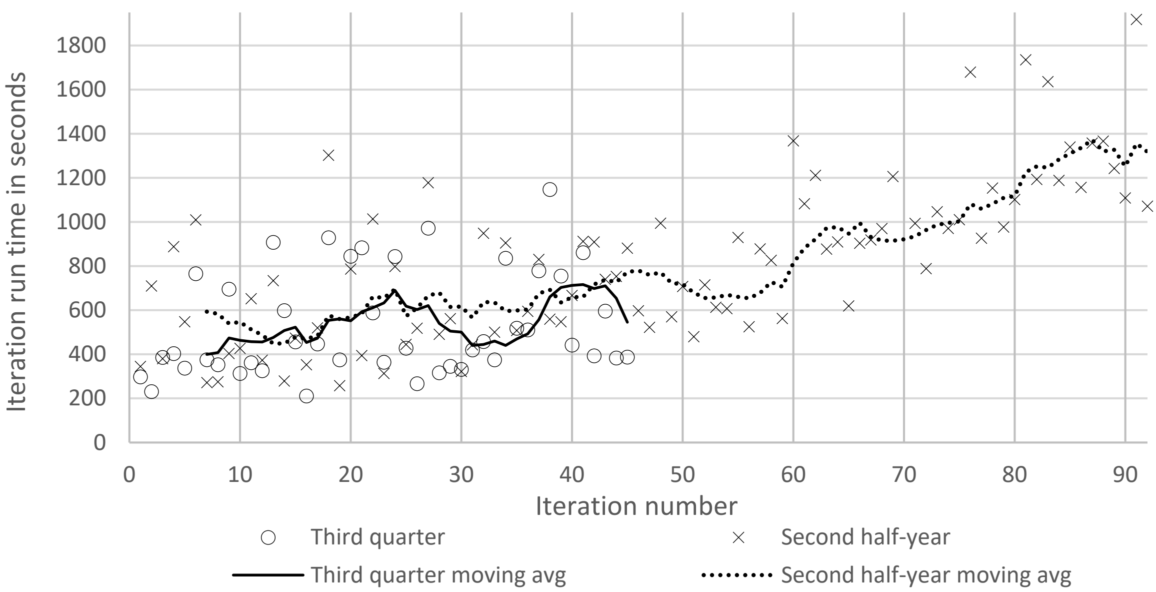

Figure 8.1 – Iteration run times for all iterations of RHH+CT+SSR at the third quarter (92 days) and second half-year (184 days) instances.

The iteration run times for the third quarter instance is, on average, increasing slightly as the iteration number increases. In fact, the average over the final 26 iterations is about 10 percent higher than the first 26 iterations.

The gradual increase in run time is more pronounced with the second half-year instance as it escalates beyond the end of the third quarter. Due to the noisiness of the data, it is hard to draw definitive conclusions, but the growth in iteration run times might very well be exponential rather than linear. For planning horizons shorter than a quarter year, this difference is unimportant as the increase in iteration run time is small. However, the full solution time for instances with longer planning horizons might be vastly different depending on the iteration run times growing exponentially or linearly. Relying on the iteration run time data from the second half-year instance and assuming linear growth, I predict a solution time for a full year instance at 215 hours (9 days), while assuming exponential growth I predict a solution time at 885 hours (37 days) using the same model and hardware. It seems that a planning horizon of half-a-year is near the limit of reasonable solution time for this approach.

The implementation of the solution method can probably be improved to decrease or remove the growth in iteration run times. In theory, the iteration run time for RHHs should be constant as the iteration number increase because the later iterations should not be influenced by the FzP. It is unclear what is causing this growth in iteration run times.

The increasing iteration run times leads to superlinear growth in computational time as the planning horizon increases, making it disproportionally harder to solve instances with longer planning horizons. However, a main goal of this thesis is utilizing a heuristic approach to solve realistic instances with long planning horizons within reasonable solution time. With the solution to a half-year long, realistic instance found in about 21 hours on a regular desktop PC, this goal is achieved. If this solution method is to be implemented in a real life use case and even longer instances or shorter solution times are needed, then further improving this solution method might be worthwhile. However, for this thesis I will not pursue the matter any further.

I have thus far in this chapter covered Stage III and IV of Figure 7.1 and discussed the results of the tuned RHH+CT+SSR solution method on instances with long planning horizons. Although scenario analyses are not the focus of this thesis, the depot utilization pattern from the instances tested is particularly interesting. In Figure 8.2 I plott the utilization of all depots with light and dark gray indicating Saturdays and Sundays/holidays of the second half-year instance.

The total available capacity across all depots is 120, while there are 105 trains modeled in this thesis. Max capacity is utilized at all depots at nearly all times, with one major and two significant exceptions. The major exception is Stortinget depot (STTD) with a max capacity of 22 trains, but only have an oscillating utilization between 8 and 12 depot spots. Utilization at Stortinget depot is at its lowest during the weekends, especially on Saturdays. This might indicate that Stortinget depot is at overcapacity and can be downgraded without increasing deadheading. Another significant exception is Avløs depot (AVL) which hits its max capacity consistently on Saturdays, when Stortinget depot is at its lowest utilization. However, Saturdays is generally the only day Avløs depot it fully utilized. Ellingsrudåsen-A (ELÅA), the restricted part of Ellingsrudåsen depot, is also usually at full capacity during weekdays, but reduced by one to three on weekends. Conversely, the depots utilized at full capacity for a sustained period of time indicate that an expansion of the capacity at these depots may lead to savings in total deadheading.

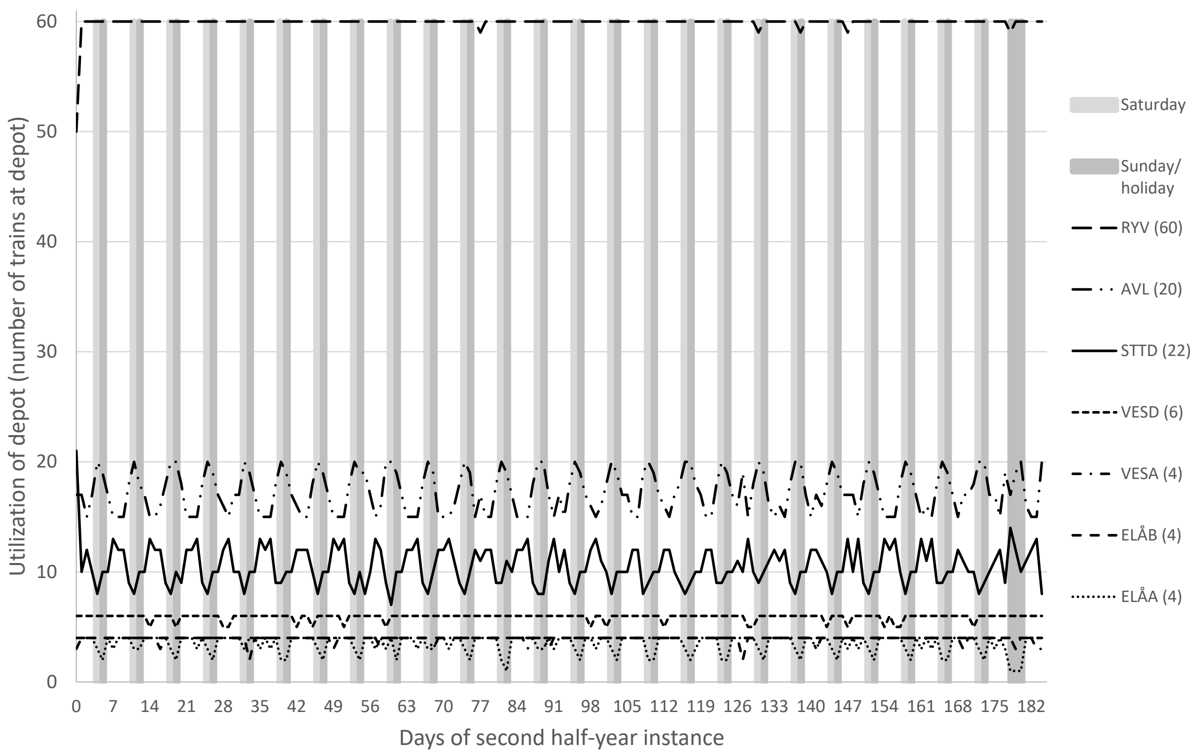

Figure 8.2 – Utilization of depots per day in the solution to the second half-year instance. Depots listed by abbreviation as given in Table A.2 and max capacity in parenthesis.

9 Concluding Remarks

In this final chapter, I conclude the thesis and discuss potential future research on the Multi-Depot Periodic Vehicle Routing Problem (MDPVRP) for metro systems. Section 9.1 provides a short account of the problem, solution methods and results, while in section 9.2, I discuss further research and potential improvements on the approach studied in this thesis.

9.1 Problem and results

This thesis is written in collaboration with Sporveien AS, the operator of the Oslo Metro. To reduce operational costs, Sporveien wants to reduce the driving of empty trains — so called deadheading — between depots and blocks by improving the allocation of their homogeneous fleet of trains to depots and blocks, while satisfying depot capacity constraints and train cleaning requirements. Deadheading is a costly activity, where the main cost drivers are power consumption, wear and personnel costs for train drivers, and it does not directly contribute with increased revenue. The problem identified in this thesis is a version of the Multi-Depot Periodic Vehicle Routing Problem, which to the best of my knowledge has never before been studied or solved. No previous research on metro systems covers neither periodic block-vehicle-depot matching problems nor problems on reducing deadheading.

Based on current operations of the Oslo Metro, I have stated a integer programming formulation of this problem and implemented a Branch-and-Bound algorithm using a commercial solver for solving the full model. The full model is only solvable within reasonable solution time for realistic instances of planning horizons up to a week. I therefore develop a Rolling Horizon Heuristic (RHH) based model to drastically reduce solution time without a considerable reduction in solution quality. Best performing approach is the RHH combined with solution space reduction (SSR) techniques, selected based on a comparative computational study on realistic test instances with planning horizons of medium length. This version is applied on realistic instances with a planning horizon of a quarter year, half-year, and a full year. Sporveien would benefit from such long horizon plans because crew scheduling and long-term maintenance planning will be more accurate when having a rolling stock schedule as a fundamental. The quarter year and half-year instances are solved within 24 hours, while the full year instance is not. Compared to the exact solution method, this heuristic approach is orders of magnitudes faster while still producing high quality solutions.

The solution to the half-year instance provide an 18.6 percent deadheading reduction compared to Sporveien's current solution, equivalent to cost savings of about NOK 3.4 million over six months. Some simplifying assumptions have been made to reduce the scope of the thesis, but the model and solution method may immediately support Sporveien in improving their routing of trains between depots and blocks. The results demonstrate that Sporveien likely has a substantial potential in reducing costs attributed to deadheading. Furthermore, the results also show that cost reductions may also be gained by adjusting the capacity of existing depots.

9.2 Future research

A considerable simplifying assumption of the problem described in this thesis is the absence of modeling the propagation of trains going in and out of depots. Solutions found using the model in this thesis might be practically problematic because they imply that trains leave or arrive at the same depot concurrently. A similar problem arise if trains leave or arrive a depot when trains in traffic are scheduled on the main track outside of the depot. Trains who need to stop and wait might cause propagating delays. One way of extending this model to account for propagation in and out of depots by adding a sub routine where solution candidates are checked for likely propagation problems.

Another possible extension is integrating crew scheduling into the model. Several identified papers in Table 3.2 on metro systems have successfully integrated crew and vehicle scheduling to satisfy both physical restrictions on trains and crew constraints like working hour provisions in the same model.

This thesis includes cleaning constraints for the rolling stock, but trains also undergo a series of long-term maintenance checks over the course of their lifespan. A train could be taken out of service for half a day to several days. Each maintenance level is performed after a certain mileage or time period, and due to limited workshop capacity, too many trains should not be forced to be taken out of service at the same time. On the other hand, performing the checks too early wastes money and train capacity as the next maintenance check will need to happen correspondingly early. Utilizing the differences in driving length per block, an extended model may also incorporate a long-term maintenance schedule that spread out the checks appropriately.

The solution method developed and tested in this thesis may also be improved for increased solution quality and reduced computing time. The RHH is sometimes used in conjunction with an improvement heuristic, which after finding feasible solutions relax some frozen variables and reruns the optimization algorithms to look for better solutions — comparable to a local search heuristic. Another way of improving the implementation is re-engineering the SSR heuristics to better exploit the structure of the symmetry in the problem. In this thesis I have identified the extensive symmetry, but the SSR is applied cruelly which possibly cause infeasibility for a part of the model, while other parts still retains significant symmetry.

Extensive symmetry and near symmetry are be strong arguments for implementing a state based commodity flow model, similar to Dávid and Krész (2018), and solve it using heuristics. Another alternative approach is basing the problem formulation on a Hitchcook-Koopmans Transportation Problem (Ford Jr & Fulkerson, 1956) and integrate the Vogel Approximation Method (Shore, 1970) or the Feasibility Pump heuristic (Berthold et al., 2019) with a RHH, possibly combined with an improvement heuristic. A comparative study of these approaches might be a fruitful endeavour.

Finally, testing the solution approach on other use cases than the Oslo Metro will improve the robustness of the model and possibly reveal weaknesses that are hard to uncover based on a single case study. Other similar use cases include metro systems and BRT systems.

Appendices

A Distance matrix

Table A.1 – Effective travel distances between depots and stations. Depots in bold. Refer to Table A.2 for full name versions. Distances between stations omitted as they are not used. To adjust for slower driving speeds on the western branch of Line 1 between Majorstuen and Frognerseteren, the distances involving Frognerseteren (FRS) and Holmenkollen (HOK) are increased by 62 percent for the relevant segments as described in Section 7.1.

From \ To	**RYV**	**AVL**	**STTD**	STTS	**VESD**	**VESA**	**ELÅA**	**ELÅB**	ØSÅ	BKR	SOG	HFY	HOK	KOL	FRS	MOR
RYV	N/A	20930	6910	6910	16750	16750	11520	11520	17150	3560	15430	2840	21020	23750	28360	13130
AVL	20930	N/A	27840	14020	29070	29070	26270	26270	14680	24490	17560	18090	23150	2820	30490	27880
STTD	6910	24020	N/A	10000	15050	15050	12250	12250	20240	10470	13520	4070	24110	26840	31450	13860
STTS	6910	14020	13820	N/A	15050	15050	12250	12250	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
VESD	22460	29070	29370	15050	N/A	0	28800	28800	25290	27020	23570	20620	29160	31890	36500	30410
VESA	22460	29070	29370	15050	0	N/A	28800	28800	25290	27020	23570	20620	29160	31890	36500	30410
ELÅA	11520	26270	18430	12250	22090	22090	N/A	0	22490	15080	20770	8180	26360	29090	33700	18470
ELÅB	11520	26270	18430	12250	22090	22090	0	N/A	22490	15080	20770	8180	26360	29090	33700	18470
ØSÅ	17150	14680	24060	N/A	25290	25290	22490	22490	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
BKR	3560	24490	10470	N/A	20310	20310	15080	15080	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
SOG	15430	17560	22340	N/A	23570	23570	20770	20770	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
HFY	2840	18090	9750	N/A	13910	13910	8180	8180	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
HOK	21020	23150	27930	N/A	29160	29160	26360	26360	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
KOL	23750	2820	30660	N/A	31890	31890	29090	29090	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
FRS	28360	30490	35270	N/A	36500	36500	33700	33700	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
MOR	13130	27880	20040	N/A	23700	23700	18470	18470	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table A.2 – Full names of the short depot and station names used for brevity in the distance matrix and other places.

Depot/station short name	Depot/station full name
RYV	Ryen
AVL	Avløs
STTD	Stortinget depot
STTS	Stortinget station
VESD	Vestli part D
VESA	Vestli part A
ELÅA	Ellingsrudåsen part A
ELÅB	Ellingsrudåsen part B
ØSÅ	Østerås
BKR	Bergkrystallen
SOG	Sognsvann
HFY	Helsfyr
HOK	Holmenkollen
KOL	Kolsås
FRS	Frognerseteren
MOR	Mortensrud

B Depot split modeling

The depots at Ellingsrudåsen and Vestli are partly overlapping with the platforms at their respective stations. During normal traffic, these platforms are in use and no trains may occupy the track for storage. For modelling purposes these two depots are split in two separate depots with the same location and distance to all other depots and stations. The part of the depot that may conflict with normal traffic is incompatible with certain blocks. This is an example of such an incompatible block-depot combination: Say a given block B ends at 20:00 and there is a 30 min drive to depot N, which is also a station in use until 22:00. The train assigned block B will arrive at depot N at 20:30 and obstructs regular traffic until 22:00. This block-depot combination is therefore incompatible, and no trains can be matched with this particular combination. The whole of Stortinget depot is also unavailable during normal traffic and is thus incompatible with certain blocks.

Table B.1 – A modeling of the two depots with a two part divergent availability. Mark that part 1 is never unavailable as it is only part 2 that overlaps with the station or track.

			Unavailable during the following hours		
Original Depot	Modeled Depots	Capacity	Weekday	Saturday	Sunday/ holiday
Ellingsrudåsen	Ellingsrudåsen-A	4	05:25-24:31	09:06-24:31	09:06-24:31
	Ellingsrudåsen-B	4		never unavailable	
Vestli	Vestli-A	4	05:27-01:10	06:58-01:10	06:57-01:10
	Vestli-D	6		never unavailable	
Stortinget	[no change]	22	06:45-23:20	09:20-23:20	10:10-23:20

C Possible train states

In Sporveiens current situation, as modeled in this thesis, there are seven unique depots as listed in Table A.2. Each train is to have its interior cleaned at least every other day. This means a train may have two interior cleaning states: Either it is cleaned the previous night (zero) or the night before (one). Similarly are the trains' exterior to be cleaned at least once every five days, and possible states of exterior cleaning is thus zero, one, two, three or four days since last cleaning. However, these sets are not independent. All trains situated at a depot with interior cleaning equipment will have its interior cleaned that night, and any train situated in a depot without interior cleaning equipment will not have had its interior cleaned. The interior cleaning state is therefore fully explained by the depot. This is by a lesser degree true of exterior cleaning. Any train situated at the Ryen depot will have had its exterior cleaned the previous night. Trains situated at other depots will not have had their exterior cleaned the previous night, but it might have been cleaned one to four nights ago. A summation of these observations is provided in Table C.1. Therefore, there are 25 possible unique states.

Table C.1 – Possible states of trains at the Metro organized by depot. Depot code names as provided in Table A.2

Depot	Days since previous interior cleaning	Days since previous exterior cleaning	Number of possible combinations
RYV	0	0	1
AVL	0	1, 2, 3 or 4	4
STTD	0	1, 2, 3 or 4	4
VESD	1	1, 2, 3 or 4	4
VESA	1	1, 2, 3 or 4	4
ELÅA	1	1, 2, 3 or 4	4
ELÅB	1	1, 2, 3 or 4	4
		SUM:	25

References

Adewumi, A. O., & Adeleke, O. J. (2018). A survey of recent advances in vehicle routing problems. *International Journal of System Assurance Engineering and Management*, *9*(1), 155–172.

Agrawal, A., & Dhingra, S. (1989). An optimal programme for augmentation of capacities of depots and shipment of buses from depots to starting points of routes. *Indian Journal of Pure and Applied Mathematics*, *20*(2), 111–120.

Alonso, A., Escudero, L. F., & Ortuno, M. T. (2000). A stochastic 0–1 program based approach for the air traffic flow management problem. *European Journal of Operational Research*, *120*(1), 47–62.

APTA. (2020). *Public Transportation Fact Book 2020 Appendix A: Historical Data Tables* (71st, tech. rep.). American Public Transportation Association. Washington DC. [My calculations from Table 9 & 11].

Archetti, C., & Speranza, M. G. (2014). A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, *2*(4), 223–246.

Baker, K. R. (1977). An Experimental Study of the Effectiveness of Rolling Schedules in Production Planning. *Decision Sciences*, *8*(1), 19–27.

Baldoquin, M. G., & Campo, A. J. R. (2018). A model for solving vehicle scheduling problems: a case study. *Revista Facultad de Ingeniería Universidad de Antioquia*, *88*, 16–25.

Berthold, T., Lodi, A., & Salvagnin, D. (2019). Ten years of feasibility pump, and counting. *EURO Journal on Computational Optimization*, *7*(1), 1–14.

Bertossi, A. A., Carraresi, P., & Gallo, G. (1987). On some matching problems arising in vehicle scheduling models. *Networks*, *17*(3), 271–281.

Blanco, V., Conde, E., Hinojosa, Y., & Puerto, J. (2020). An optimization model for line planning and timetabling in automated urban metro subway networks. A case study. *Omega*, *92*, 1–18.

Bunte, S., & Kliewer, N. (2009). An overview on vehicle scheduling models. *Public Transport*, *1*(4), 299–317.

Business Insider. (2018). 13 cities that are starting to ban cars [Online; accessed 10-December-2018].

Cacchiani, V., & Toth, P. (2012). Nominal and robust train timetabling problems. *European Journal of Operational Research*, *219*(3), 727–737.

Cadarso, L., Marín, Á., & Maroti, G. (2013). Recovery of disruptions in rapid transit networks. *Transportation Research Part E: Logistics and Transportation Review*, *53*, 15–33.

Cadarso, L., & Marín, Á. (2014). Recovery of Disruptions in Rapid Transit Networks with Origin-Destination Demand. *Procedia - Social and Behavioral Sciences*, *111*, 528–537.

Cadarso, L., Maroti, G., & Marín, Á. (2015). Smooth and Controlled Recovery Planning of Disruptions in Rapid Transit Networks. *Intelligent Transportation Systems, IEEE Transactions on*, *16*, 2192–2202.

Chang, Y., Niu, R., Wang, Y., Luan, X., D'Ariano, A., & Sama, M. (2019). Train Rescheduling for an Urban Rail Transit Line under Disruptions, In *RailNorrköping 2019. 8th International Conference on Railway Operations Modelling and Analysis*.

Dantzig, G. B., & Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science*, *6*(1), 80–91.

Dávid, B., & Krész, M. (2018). Multi-depot bus schedule assignment with parking and maintenance constraints for intercity transportation over a planning period. *Transportation Letters*, *12*(1), 66–75.

de Araujo, S. A., Arenales, M. N., & Clark, A. R. (2007). Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. *Journal of Heuristics*, *13*(4), 337–358.

Derigs, U. (1988). The Hitchcock Transportation Problem, In *Programming in Networks and Graphs: On the Combinatorial Background and Near-Equivalence of Network Flow and Matching Algorithms*. Springer Berlin Heidelberg.

Ding, L. (2018). *Design of Urban Rail Corridor over Time: For Cities with High Population Densities and Growth Uncertainties* (Doctoral dissertation). The Hong Kong Polytechnic University. Hung Hom, Kowloon, Hong Kong.

Fonseca, J. P., van der Hurk, E., Roberti, R., & Larsen, A. (2018). A matheuristic for transfer synchronization through integrated timetabling and vehicle scheduling. *Transportation Research Part B: Methodological*, *109*, 128–149.

Ford Jr, L. R., & Fulkerson, D. R. (1956). Solving the transportation problem. *Management Science*, *3*(1), 24–32.

Freling, R., Huisman, D., & Wagelmans, A. (2003). Models and Algorithms for Integration of Vehicle and Crew Scheduling. *Journal of Scheduling*, *6*, 63–85.

Fuentes, M., Cadarso, L., & Marín, Á. (2015). A New Approach to Crew Scheduling in Rapid Transit Networks. *Transportation Research Procedia*, *10*, 554–563.

Fuentes, M., Cadarso, L., & Marín, Á. (2019). A hybrid model for crew scheduling in rail rapid transit networks. *Transportation Research Part B: Methodological*, *125*, 248–265.

Golden, B., Raghavan, S., & Wasil, E. (Eds.). (2008). *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer US.

Google Scholar. (2020). Search engine used to find relevant literature [https://scholar.google.com/].

Hadjiconstantinou, E., & Baldacci, R. (1998). A multi-depot period vehicle routing problem arising in the utilities sector. *Journal of the Operational Research Society*, *49*(12), 1239–1248.

HassanNayebi, E., Zegordi, S. H., Amin-Naseri, M. R., & Yaghini, M. (2016). Train Timetabling at Rapid Rail Transit Lines: A Robust Multi-Objective Stochastic Programming Approach. *Operational Research*, *17*(2), 435–477.

HassanNayebi, E., Zegordi, S. H., Amin-Naseri, M. R., & Yaghini, M. (2018). Optimizing headways for urban rail transit services using adaptive particle swarm algorithms. *Public Transport*, *10*(1), 23–62.

Huang, Y., Mannino, C., Yang, L., & Tang, T. (2020). Coupling time-indexed and big-M formulations for real-time train scheduling during metro service disruptions. *Transportation Research Part B: Methodological*, *133*, 38–61.

Huisman, D. (2004). *Integrated and Dynamic Vehicle and Crew Scheduling* (Doctoral dissertation). Erasmus University Rotterdam. Burgemeester Oudlaan 50, 3062 PA Rotterdam, Nederland.

Huisman, D., Freling, R., & Wagelmans, A. P. M. (2005). Multiple-Depot Integrated Vehicle and Crew Scheduling. *Transportation Science*, *39*(4), 491–502.

Institute for Transportation & Development Policy. (2017). Rapid Transit Trends Show Record Growth in 2016, with Huge Increases in China, Brazil [Online; last accessed 18-May-2020].

Jiang, Z., Tan, Y., & Yalçınkaya, O. (2014). Scheduling Additional Train Unit Services on Rail Transit Lines. *Mathematical Problems in Engineering*, *Volume 2014 (2014)*, 1–13.

Kepaptsoglou, K., Karlaftis, M., & Bitsikas, T. (2010). Bus-to-Depot Allocation: Models and Decision Support System. *Journal of Transportation Engineering*, *136*, 600–605.

Laporte, G., Ortega, F., Pozo, M., & Puerto, J. (2017). Multi-objective integration of timetables, vehicle schedules and user routings in a transit network. *Transportation Research Part B: Methodological*, *98*, 94–112.

Lie, S., & Sinnes, J. (2019). *Rescheduling of light rail trains during disruption: an optimization model for Bybanen in Bergen* (Master's thesis). Norwegian School of Economics.

Liu, M., Li, Y., Zheng, F., & Chu, F. (2017). Integration of Timetabling, Multi-type Vehicle Scheduling and User Routing in Public Transit Network considering Fuel Consumption, In *7th International Conference on Industrial Engineering and Systems Management (IESM 2017)*, Saarbrücken, Germany.

Lösch, A. (1954). *The Economics of Location* (Second). New Haven, Connecticut, Yale University Press.

Lusby, R. M., Larsen, J., Ehrgott, M., & Ryan, D. (2011). Railway track allocation: models and methods. *OR Spectrum*, *33*, 843–883.

Margot, F. (2010). Symmetry in integer linear programming. In M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, & L. A. Wolsey (Eds.), *50 Years of Integer Programming 1958-2008* (pp. 647–686). Springer Berlin Heidelberg.

Marín, Á. G. (2006). Airport management: taxi planning. *Annals of Operations Research*, *143*(1), 191–202.

Mathirajan, M. (1987). Optimal Allocation of Buses to depots - A Case Study. *OPSEARCH*, *24*, 228–239.

Mathirajan, M. (1993). An efficient heuristic for minimizing dead kilometers in Urban Transport Systems. *J Transp Manag*, *17*, 125–133.

Mathirajan, M., Hariharakrishnan, C., & Ramachandran, V. (2010). An experimental evaluation of heuristic algorithms for bus-depot matching problem of urban road transport systems. *OPSEARCH*, *47*(2), 143–157.

Maze, T., Khasnabis, S., & Kutsal, M. D. (1983). Application of a bus garage location and sizing optimization. *Transportation Research Part A: General*, *17*(1), 65–72.

Mercé, C., & Fontan, G. (2003). MIP-based heuristics for capacitated lotsizing problems. *International Journal of Production Economics*, *85*(1), 97–111.

Mohammadi, M., Ghomi, S. F., Karimi, B., & Torabi, S. A. (2010). Rolling-horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lotsizing problem with sequence-dependent setups. *Journal of Intelligent Manufacturing*, *21*(4), 501–510.

Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., & Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, *79*, 115–129.

Moreno, C., Falcón, L., Bolaños, R., Subramanian, A., Zuluaga, A., & Echeverri, M. (2019). A hybrid algorithm for the multi-depot vehicle scheduling problem arising in public transportation. *International Journal of Industrial Engineering Computations*, *10*(3), 361–374.

Nasibov, E., Eliiyi, U., Ertaç, M., & Kuvvetli, U. (2013). Deadhead Trip Minimization in City Bus Transportation: A Real Life Application. *PROMET - Traffic&Transportation*, *25*(2), 137–145.

Ortega, F. A., Mesa, J. A., & Pozo, M. A. (2018). Determining Routes for an Auxiliary Fleet in Presence of Emergencies on a Railway Line, In *International Conference on Information Systems, Logistics and Supply Chain*. Springer.

Perre, P. P. V. d., & Oudheusden, D. D. V. (1997). Reducing depot-related costs of large bus operators a case study in Bangkok. *European journal of operational research*, *96*(1), 45–53.

Rakke, J. G., Stålhane, M., Moe, C. R., Christiansen, M., Andersson, H., Fagerholt, K., & Norstad, I. (2011). A rolling horizon heuristic for creating a liquefied natural gas annual delivery program. *Transportation Research Part C: Emerging Technologies*, *19*(5), 896–911.

Ramos, F. L. (2013). *Conjoint design of railway lines and frequency setting under semi-congested scenarios* (Doctoral dissertation). The Polytechnic University of Catalonia. Campus Nord, Carrer de Jordi Girona, 1, 3, 08034 Barcelona, Spania.

Sevim, İ., Tekiner-Moğulkoç, H., & Güler, M. G. (2020). Scheduling the vehicles of bus rapid transit systems: a case study. *International Transactions in Operational Research*, 1–25.

Sharma, V., & Prakash, S. (1986). Optimizing Dead Mileage in Urban Bus Routes. *Journal of Transportation Engineering*, *112*(1), 121–129.

Shore, H. H. (1970). The transportation problem and the Vogel approximation method. *Decision Sciences*, *1*(3-4), 441–457.

Sporveien AS. (2015). Skematisk oversikt med stasjoner, overkj.spor m/detaljer, relérom og likeretterst. [Restricted, not public].

Sporveien AS. (2019). Årsrapport 2019 [Sporveien annual report 2019 in Norwegian].

Statistics Norway. (2018). Over 1 million innbyggere i Oslo tettsted [Online; accessed 10-December-2018].

Uggen, K. T., Fodstad, M., & Nørstebø, V. S. (2013). Using and extending fix-and-relax to solve maritime inventory routing problems. *Top*, *21*(2), 355–377.

Uyeno, D. H., & Willoughby, K. A. (1995). Transit centre location-allocation decisions. *Transportation Research Part A: Policy and Practice*, *29*(4), 263–272.

Wikipedia. (2020a). City of Izmir [Online; accessed 21-April-2020].

Wikipedia. (2020b). Rapid transit [Online; accessed 12-April-2020].

Wilhelm, W. E., & Parks, W. J. (1983). Cost Trade-Off Analysis for Bus Facility Expansion. *Journal of Transportation Engineering*, *109*(1), 124–136.

Willoughby, K. A. (2002). A mathematical programming analysis of public transit systems. *Omega*, *30*(3), 137–142.

Xu, X., Ye, Z., Li, J., & Wang, C. (2018). Solving a Large-Scale Multi-Depot Vehicle Scheduling Problem in Urban Bus Systems. *Mathematical Problems in Engineering*, *2018*, 1–13.

Zhang, M., Wang, Y., Su, S., Tang, T., & Ning, B. (2018). A short turning strategy for train scheduling optimization in an urban rail transit line: the case of Beijing subway line 4. *Journal of Advanced Transportation*, *2018*, 1–19.

Zhou, Y., Bai, Y., Li, J., Mao, B., & Li, T. (2018). Integrated optimization on train control and timetable to minimize net energy consumption of metro lines. *Journal of Advanced Transportation*, *2018*, 1–19.

NTNU

Norwegian University of
Science and Technology