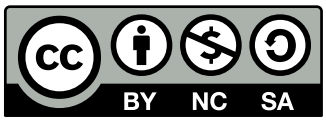
Aalto-yliopisto
Aalto-universitetet
Aalto University

Master's Programme in Mathematics and Operations Research

Optimization of Rolling Stock Rotations for Long-distance Trains in Finland

Salla Nicholls

Master's Thesis
2026



Aalto-yliopisto
Aalto-universitetet
Aalto University

Author Salla Nicholls

Title Optimization of Rolling Stock Rotations for Long-distance Trains in Finland

Degree programme Mathematics and Operations Research

Major Systems and Operations Research

Supervisor Asst. Prof. Philine Schiewe

Advisor Dr Per Jernström

Collaborative partner VR-Yhtymä Oyj

Date 27 April 2026 **Number of pages** 76+4 **Language** English

Abstract

Determining rolling stock rotations is an essential part of a railway operator's planning process. Rolling stock refers to the physical train compositions used to operate timetabled train services, and a rotation is a feasible sequence of services operated by one composition. An optimal rolling stock rotation plan minimizes costs and lost revenue, while ensuring that all timetabled train services are covered and operational constraints are respected.

In this thesis, the rolling stock rotations are optimized for the Finnish passenger railway operator VR's long-distance trains. Two types of rolling stock rotation plans are considered: cyclic standard week plans and acyclic exception week plans. A standard week plan is repeated over multiple weeks, while the exception week plans are used for weeks with deviations from the standard week's timetable.

Two MILP models based on multicommodity flow formulations are developed. They minimize operational costs and lost revenue caused by inadequate capacity allocation with respect to passenger demand. The constraints considered include the limited availability of train compositions and wagons, distance-based maintenance, composition orientation and the maximum length of multiweek rotations in a standard week plan. In addition, the allocation of restaurant wagons to trains is constrained.

Due to the complexity of the problem, obtaining high-quality solutions in a reasonable time is difficult. Therefore, alternative solution methods for solving the models are developed and evaluated with real-life timetables provided by VR. The results demonstrate the validity of the models and show that generating an initial feasible solution to warm start the solver significantly improves the computational performance. Furthermore, the use of decomposition heuristics further accelerates finding high-quality solutions for larger timetables.

As is typical for rolling stock rotation models, the models developed in this thesis are tailored to the operating environment of a specific railway operator, in this case, VR. Nevertheless, this work contributes to the literature by incorporating orientation modelling, exception weeks and applying cut separation, initial feasible solution generation and decomposition techniques in the solution process.

Keywords rolling stock rotation optimization, multicommodity flows, decomposition method, initial feasible solution, MILP

Aalto-yliopisto
Aalto-universitetet
Aalto University

Tekijä Salla Nicholls

Työn nimi Kaukojunaliikenteen runkokiertojen optimointi Suomessa

Koulutusohjelma Mathematics and Operations Research

Pääaine Systems and Operations Research

Työn valvoja Apulaisprofessori Philine Schiewe

Työn ohjaaja TkT Per Jernström

Yhteistyötaho VR-Yhtymä Oyj

Päivämäärä 27.4.2026 **Sivumäärä** 76+4 **Kieli** englanti

Tiivistelmä

Runkokiertojen määrittäminen on olennainen osa junaoperaattorin suunnitteluprosessia. Rungoilla tarkoitetaan fyysisiä junayksiköitä, joilla operoidaan aikataulun mukaiset junavuorot. Kierrolla puolestaan tarkoitetaan sarjaa junavuoroja, jotka yksi junayksikkö operoi. Optimaalisessa runkokiertosuunnitelmassa operointikustannukset sekä menetetty myynti on minimoitu niin, että jokaiselle junavuorolle on osoitettu junayksikkö ja operatiiviset rajoitteet on huomioitu.

Tässä työssä kehitetään kaksi optimointimallia suomalaisen matkustajajunaliikenneoperaattori VR:n kaukojunaliikenteen runkokiertojen optimointia varten. Mallit luodaan syklisille malliviikko- sekä asyklisille poikkeusviikkosuunnitelmille. Malliviikkosuunnitelmaa hyödynnetään usean viikon yli, kun taas poikkeusviikkosuunnitelmia käytetään viikoilla, joilla on aikataulumuutoksia malliviikkoon nähden.

Työssä kehitettävät MILP-mallit pohjautuvat monihyödykevirtauksiin (eng. multicommodity flow) ja minimoivat operointikustannukset sekä riittämättömästä kapasiteetista suhteessa matkustajakysyntään aiheutuvan menetetyn myynnin. Mallit huomioivat rajoitteet liittyen kaluston saatavuuteen, kilometripohjaisiin huoltoväleihin, runkojen orientaatioon Helsingin päärautatieasemalla sekä malliviikkojen kiertojen maksimipituuden. Lisäksi huomioidaan ravintolavaunujen allokaatio junavuoroihin.

Ongelman kompleksisuuden vuoksi korkealaatuisten ratkaisujen löytäminen kohtuullisessa ajassa on vaikeaa. Tämän vuoksi työssä kehitetään vaihtoehtoisia ratkaisumenetelmiä, joiden tehokkuutta arvioidaan aidoilla aikatauluaineistoilla. Tulokset osoittavat mallien toimivuuden sekä sen, että hyödyntämällä alkuratkaisua ratkaisuprosessissa voidaan merkittävästi parantaa ratkaisuaikaa. Lisäksi hajoitusmenetelmillä voidaan tehostaa korkealaatuisten ratkaisujen löytämistä.

Kuten runkokierto-optimointimalleille on tyypillistä, tässä työssä kehitetyt mallit on räätälöity yksittäisen junaoperaattorin, tässä tapauksessa VR:n, tarpeisiin. Tästä huolimatta työ edistää tutkimuskirjallisuutta esittäen uusia tapoja mallintaa orientaatioita ja poikkeusviikkoja sekä hyödyntäen ratkaisuprosessissa leikkausten generointia, alkuratkaisuja ja hajoitusmenetelmiä.

Avainsanat runkokierto-optimointi, monihyödykevirtaukset, alkuratkaisu, hajoitusmenetelmä, MILP

Notice of AI usage

Large language models (ChatGPT, Microsoft Copilot) were used to assist in the preparation of this thesis. They were used to support conceptualization, search for relevant papers, LaTeX formatting and rephrasing of text for clarity.

Contents

Symbols and abbreviations

Symbols

$\mathbb{N}$	Set of natural numbers, including 0
$\mathbb{R}$	Set of real numbers
$\mathbb{R}_{\geq 0}$	Set of non-negative real numbers

Abbreviations

IC	InterCity
IFS	Initial Feasible Solution
MCFP	Multicommodity Flow Problem
min.	minimize
MILP	Mixed Integer Linear Programming
MIP	Mixed Integer Programming
RSR	Rolling Stock Rotation
RSRP	Rolling Stock Rotation Problem
S	Pendolino
s.t.	subject to
VR	Valtionrautatiet
w.r.t.	with respect to

1 Introduction

Efficient railway operations require planning that involves coordination of a wide range of decisions and resources under complex operational constraints. To handle this complexity, planning is typically structured in a hierarchical manner [1], distinguishing between strategic, tactical and operational levels. On a strategic level, the line and network structure are determined, while the tactical level involves timetable planning, determining rolling stock rotations and crew scheduling [2]. On the operational level, these plans are adjusted in real time to account for unforeseen disturbances. At each stage, plans aim to maximize efficiency and service quality while respecting operational constraints. To support decision-making across these levels, Operations Research methods, in particular optimization techniques, are widely applied.

Rolling stock rotation (RSR) planning is a central tactical level problem in passenger railway operations due to its direct impact on operational costs, service quality and operational reliability. This planning stage involves assigning rolling stock compositions to timetable-defined train services and determining the sequence in which each composition operates these services. The plans are subject to a range of operational constraints, such as maintenance requirements and availability of rolling stock. A key objective in the planning process is to minimize overall costs. In addition to ensuring that each scheduled train is assigned a feasible composition, passenger railway operators aim to match the capacity of the rolling stock with passenger demand [3]. This reduces lost revenue caused by unmet demand and improves passenger satisfaction.

This thesis focuses on the RSR planning for long-distance train services of a Finnish railway operator, VR. Two types of RSR plans are considered: standard week plans and exception week plans. Standard week plans are repeated over a number of weeks, while exception week plans capture deviations from the standard week plan due to, for example, holidays or track work.

In a previous master's thesis, opportunities for improvement in the planning process of RSRs at VR have been identified [4]. In particular, the current optimization algorithm utilized in planning of RSRs is unable to consider the whole traffic at once, is time consuming and imprecise in the train composition types in the output. In addition, the current optimization algorithm is not applicable for planning of exception weeks. Therefore, planning of exception weeks is done manually, causing unnecessary workload and suboptimal plans. A particular challenge identified with applying the current optimization algorithm to exception weeks is that the model must consider the RSR plan of prior week as input information. This thesis will address these identified limitations. A wide range of mathematical models for optimizing RSR plans exist in the literature [5], however, due to operator specific constraints, these existing models are not directly applicable. Therefore, in this thesis, tailored optimization models incorporating VR's specific constraints are developed.

This thesis addresses the previously identified limitations by developing two Mixed Integer Linear Programming (MILP) models for optimization of the rolling stock rotation problem (RSRP) at VR; one for the standard weeks and another for exception weeks. The models are based on multicommodity flow formulations with integer

requirements [6] and the model formulations are similar to those of composition models [7], commonly used in RSRPs. Multiple solution methods are evaluated to cope with the computational complexity of real-life problem sizes. The key techniques applied in the methods include i) generation of an initial feasible solution (IFS) for warm starting the solution process, ii) decomposition of the problem into subproblems which are solved in parallel [8] and iii) application of large neighbourhood search (LNS) [9]. These methods are compared to solving the model directly with a commercial solver.

In this thesis, the constraints considered in the RSRP are distance-based maintenance, orientation of the compositions at Helsinki station, allocation of restaurant wagons and maximum length of multiweek rotations in standard weeks. In addition, the availability of each type of composition and wagon is assumed to be limited and thus considered a constraint in the optimization. In addition to the previously mentioned objectives of minimizing operational costs and lost revenue caused by unmet demand, undesirable turn times are minimized.

This thesis is structured as follows. Section 2 presents the relevant background for optimization of rolling stock rotations, including an overview of long-distance trains in Finland, the vehicle scheduling problem, the rolling stock rotation problem (RSRP) and introducing formally the operational requirements considered. Section 3 introduces the theoretical foundations for the RSRP models and Section 4 provides a detailed description of the developed models and proposes four solution methods. The models and solution methods are validated through computational experiments and the results of these experiments are reported and analysed in Section 5. Finally, Section 6 concludes and describes the limitations of this work in addition to providing recommendations for future work.

2 Background

2.1 Long-distance trains in Finland

VR (Valtionrautatiet) is the only railway operator in Finland which provides long-distance passenger train services. These train services connect major cities, as illustrated in Figure 1, in which the network of long-distance train routes operated in Finland is presented.

In 2025, VR sold over 16 million tickets for long-distance train services [11] and operated approximately 200 long-distance trains each day in Finland [12]. One of VR's main strategic objectives is to increase the share of train travel, for which matching the capacity of VR's rolling stock with passenger demand is essential. Planning, and in particular the optimization, of rolling stock rotations plays a significant role in reaching this objective.

In the long-distance train operations at VR, the compositions and wagons are used multiple times over the planning horizon on different routes. For example, a composition may operate a train from Helsinki to Tampere and then another train from Tampere to Kuopio before returning to Helsinki, instead of only operating a train between Helsinki and Tampere. This means assigning a composition to a specific train affects its availability for subsequent trains. Systematic planning is required to match the capacity and available compositions to the trains to be operated.

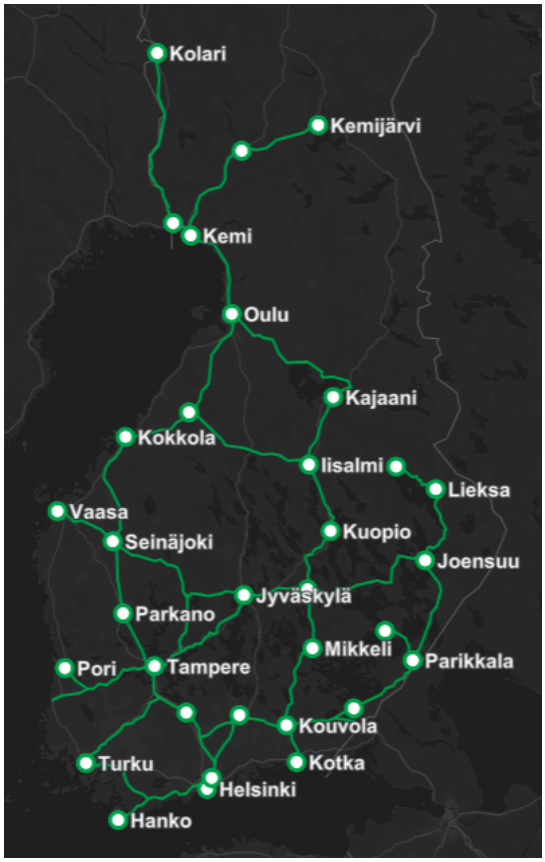


Figure 1: Network of long-distance train routes operated by VR in 2026 between major cities in Finland. Figure adapted from [10].

At VR, the planning of passenger train units and locomotive planning are treated as separate processes. The optimization of locomotive allocation at VR has previously been studied as part of a master's thesis [13]. This thesis focuses on the optimization of train units and compositions with built-in locomotives used in long-distance passenger traffic at VR and assumes locomotives are allocated separately.

2.2 Vehicle scheduling problem

Vehicle scheduling is an essential part of planning in transportation systems. For example aircraft must be assigned to flights [14], buses to timetables [15] and train units to train services. Vehicle scheduling problems are formally defined as optimization problems where one must assign vehicles to timetabled trips such that each trip has a vehicle to operate it, given constraints are satisfied and overall costs are minimized [16, 17]. Each trip has a departure and arrival location as well as a departure time and arrival time. A vehicle can operate two consecutive trips, if the arrival and departure locations and times are compatible with each other.

The vehicle scheduling problem can be thought of as an extension of the routing problem which considers time [18]. The vehicles are routed to carry out the timetabled trips in sequence while taking into account these trips might overlap or otherwise not be compatible to be operated by the same vehicle. The problem can be represented in multiple ways, for example using connection-based networks, in which trips are represented as nodes and connections between trips are modelled as arcs [17].

The objective of a vehicle scheduling problem is to minimize overall costs. The costs can be composed of fixed costs of vehicle acquisition and maintenance in addition to operational costs such as fuel, idle time or personnel required to operate a certain vehicle [17]. The fixed costs are affected by the number and types of vehicles required. Missed revenue due to inadequate assignment of a vehicle to a timetabled trip can also be considered as a cost. For example, if the capacity of an aircraft assigned to a flight is smaller than the expected demand on the flight, the company may lose revenue.

2.3 Rolling stock rotation problem

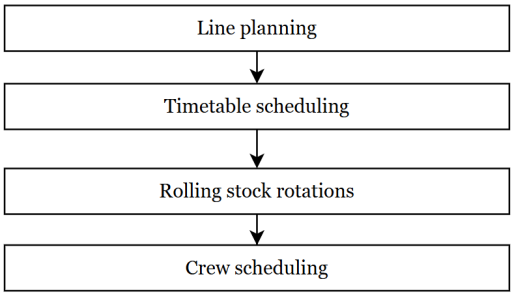


Figure 2: Typical planning process of a railway operator, as described in [19].

The rolling stock rotation problem (RSRP) arises in the planning process of railway operations. Rolling stock refers to the physical train units or compositions used to operate timetabled train services. Figure 2 illustrates the typical sequential planning process of a railway operator, in which the rolling stock rotations are determined after the timetable planning phase and prior to crew scheduling. The planning of rolling stock rotations is a key part of the tactical level [2] of railway planning.

The rolling stock rotation problem (RSRP) can be viewed as an extension of the vehicle scheduling problem in the context of (passenger) railway operations. The RSRP incorporates constraints such as composition formation, maintenance and capacity considerations. Optimization of rolling stock rotations can significantly reduce operational costs and improve the matching of rolling stock capacity and passenger demand. It has been, for example, successfully applied by railway operators NS and DB [3].

Formally, the RSRP consists of assigning rolling stock (compositions) to timetabled train services and creating rotations that determine the sequence in which each composition operates the trains [5]. These rotations span the planning horizon, which is typically a day [7, 20, 21], or a week [22, 5]. Rotations can be cyclic or acyclic. Cyclic rotations allow to repeat the planned rotations seamlessly over longer time periods. In this thesis, a cyclic rotation plan is called a standard week plan. Consequently, the rolling stock rotation for a standard week is defined as a set of cycles which collectively cover all timetabled trains [5].

The terminology used to describe concepts related to the RSRP varies in the literature; therefore, the following key concepts used in this thesis are formally defined. The definitions are adapted from the following sources: **Train** from [23], **Composition** from [20], **Turn** from [24] and **Rolling stock rotation** from [5].

Definition 1 (Train)**.** *Let* t *be a train. A train* t *has a departure day, departure time, arrival time, departure station* dep(t)*, arrival station and a distance between these stations Km*(t).

Definition 2 (Composition)**.** *Let* c *be a composition, that can be used to operate train* t*. A composition* c *has type T*(c) *and a total capacity of C*(c) *and may include a number of additional capacity Ed-wagons* C(Ed) *and a restaurant* ERd*-wagon. The composition is operated as a fixed unit over the planning horizon.*

Definition 3 (Potential turn)**.** *Let* $f = (t,t')$ *be a potential turn between trains* t *and* t′*. A potential turn between trains* t *and* t′ *exists if train* t′ *can be operated by the same composition immediately after train* t *considering timing and operational rules.*

Definition 4 (Turn)**.** *A turn* $f = (t,t')$ *is a potential turn that is selected to be in the rolling stock rotation plan.*

Definition 5 (Rolling stock rotation)**.** *Let* $\{R_1, R_2, ..., R_n\}$ *be the rolling stock rotations for a given timetable. Each rotation* R_i*,* $i \in \{1, ..., n\}$ *includes a set of trains* $\{t_1, t_2, ..., t_m\}$ *operated in order defined by the turns* $\{f_1, f_2, ..., f_m\}$*. Each train* t *is in exactly one rotation. For standard week RSR plan, flow* f_m *connects train* t_m *to* t_1 *in each rotation to ensure cyclicity. The length of a rotation is defined as the number of weeks it spans.*

The set of constraints to consider when planning and optimizing a rolling stock rotation depends on the specific operational context and the railway operator's requirements. As a result, optimization models for the RSRP differ in their characteristics and the constraints they incorporate. These differences have been summarized by

several studies [3, 25, 22, 23, 26]. Composition-related aspects include limitations on the availability of composition units, the order of units, the orientation of the compositions or train units and whether coupling and uncoupling of train units is allowed. Turn-related aspects include whether turns are predetermined or part of the decision process and if deadheading (see [3] for definition) is allowed. Passenger capacity can be incorporated as a constraint, in the objective function or in a dynamic way. Operational aspects include maintenance requirements, which can be distance-, time- or appointment-based, personnel availability and consideration of depot capacity and topology. Planning-related aspects involve consideration of regularity and robustness of the RSR plans, length of the planning horizon and whether the rotations are cyclic or acyclic. Models vary in the scope in which these requirements are considered or incorporated into the models. For example, an industrial RSRP model developed for DB [27] considers a wide range of constraints including vehicle configuration, turn duration rules, railway topology, deadheading, service paths, vehicle orientation, coupling and decoupling, regularity patterns and maintenance constraints. Across the different applications, maintenance requirements are typically of interest [28, 29]. Finally, RSRP models differ in their choice of core model and the applied solution methods.

Multiple approaches for modelling the rolling stock rotation problem exist, including train-unit assignment [21], hypergraphs [30, 24] and compositions models [7]. The latter two are the most commonly used models for optimizing the RSRP [3]. Composition models are typically based on the multicommodity flow problem (MCFP) formulation with integer requirements, where train units are represented as individual commodities [7]. The underlying network [31] representation of the problem can be divided into connection-based and time-space networks [17], both of which are used as the underlying network of MCFP formulations of the RSRP. Furthermore, either network representation can be implemented with either arc- or path-based formulations [32, 33].

Typically, mathematical formulations of the RSRP lead to NP-hard optimization problems [34, 27], making them computationally challenging to solve. Several solution methods have been proposed [22], including the Branch-and-Price algorithm, decomposition methods such as column generation and Lagrangian relaxation as well as using commercial MIP solvers. A notable example is the ROTOR optimization algorithm, developed for and used by DB for the optimization of rolling stock rotations of intercity express trains [27], which is based on the hypergraph formulation.

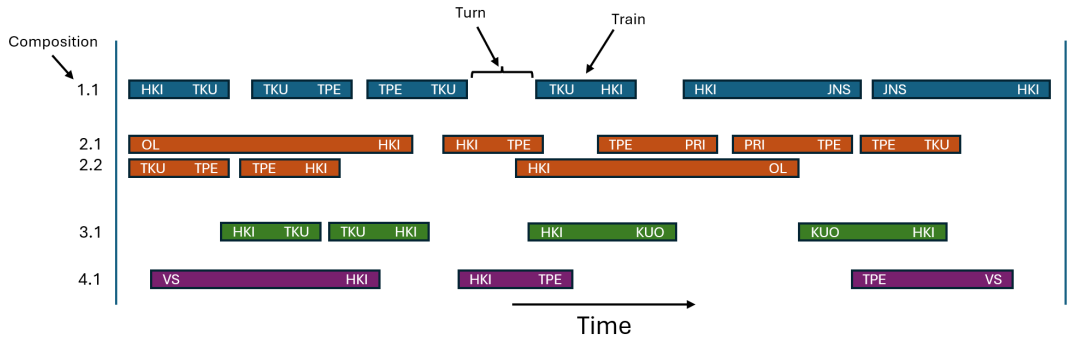


Figure 3: Example of a rolling stock rotation plan for 5 compositions. The rolling stock rotation plan is composed of four rotations: blue, orange, green and purple. Green, blue and purple rotations are of length one week and orange is of length two weeks. Each block is a train to be operated and the white space between two consecutive blocks is a turn. Only the selected turns are visible in this visualization. Labels on the left indicate individual compositions. For example composition 1.1 operates the blue rotation and compositions 2.1 and 2.2 the orange rotation. Compositions allocated to the same rotation must be of the same type.

Regardless of the chosen solution method, the output of an optimized RSRP is typically a set of rotations with assigned compositions. A small example of a such output is visualized in Figure 3 with 21 trains. Each block represents a timetabled train and the different colours represent separate rotations. The example has 4 rotations with different colours. The blue, green and purple rotations are of one week, which is repeated. The compositions for blue and green rotations start at Helsinki (HKI), operate the trains included in the rotation over the week and then return to HKI at the end of the week. The purple rotation starts at Vaasa (VS), operates three trains over the week and returns to Vaasa at the end of the week. The orange rotation is over two weeks. That is, one composition will operate the upper trains in the rotation during the first week and then the lower trains during the following week. The other composition (of the same type and capacity) will operate the lower trains during the first week and the upper trains during the following week. In reality, more trains are operated per week than in this small example.

2.4 Rolling stock compositions

Planning a rolling stock rotation for a given timetable depends strongly on the available rolling stock and its characteristics. Different composition types may have different operational requirements, capacity or compatibility with certain train services. These directly affect the feasibility of assigning compositions to train services. Therefore, the compositions and their characteristics considered in this thesis are introduced next.

The long-distance trains of VR in Finland are mainly operated by two types of compositions, which are Pendolino (S) [35] (Figure 4b) and InterCity (IC) [36] (Figure 4a), which are considered in this thesis. Pendolino-compositions include six wagons and a built-in locomotive, which cannot be detached. In contrast, IC-compositions

(a) InterCity-composition with a Sr2-locomotive. Photo by Olli Litmanen, 2026.

(b) Pendolino-composition. Photo by Olli Litmanen, 2026.

Figure 4: The two main composition types used to operate VR's long-distance trains. An IC-composition can have in addition to the locomotive 3 to 7 wagons, while Pendolino-composition always has 6 wagons and a built-in locomotive.

are composed of a flexible number of wagons and a separate locomotive. However, it is assumed that each IC-composition has always three mandatory base-wagons and a locomotive. In addition, an ERd-wagon (restaurant wagon) can be added to the composition and up to three additional capacity Ed-wagons. A restaurantless IC-composition can have up to two Ed-wagons. Thus, there are in total seven possible combinations of wagons that an IC-composition can have.

A key decision to be made in the planning process of rotations at VR is the number of each type of IC-compositions to use, as the total number of Ed-wagons and ERd-wagons available is assumed to be limited. For example, if multiple IC-compositions in the planned rotation have the maximum number of three Ed-wagons, the rest of the IC-compositions will not have any extra Ed-wagons. Similarly, some trains will be operated without an ERd-wagon.

Composition name	ERd-wagon	Ed-wagons	Capacity
IC_0	0	0	257
IC_1	0	1	370
IC_2	0	2	483
IC_R	0	0	298
$IC_{R,1}$	1	1	411
$IC_{R,2}$	1	2	524
$IC_{R,3}$	1	3	637
S	0	0	286

Table 1: Composition types and their capacities considered in the optimization. There are seven subtypes of IC-compositions, which vary by their ERd-wagon (restaurant) and Ed-wagon count. A S-composition always includes a restaurant wagon.

The capacity of a composition depends on the seats available. The three base-wagons of an IC-composition have in total of 257 seats, an ERd-wagon has 41 seats

and an Ed-wagon has 113 seats [36]. A Pendolino-composition has 286 seats [35]. These form the capacity of each composition type. In Table 1, the eight different composition types, their names and capacities considered in this thesis are listed. An IC-composition can thus be composed in seven different ways depending on if an ERd-wagon is included and how many additional Ed-wagons are used.

2.5 Operational requirements considered

For an RSRP optimization model to be applicable for practical use, it must incorporate requirements and constraints specified by the railway operator. This section presents the requirements for the optimization of the RSRP set by VR in the scope of this thesis. The objective is to obtain RSR plans in which operational costs, lost revenue caused by inadequate capacity allocation and undesirable turn times are minimized. As listed in Section 2.3, there exists a wide range of potential requirements to include in the models, many of which are operator specific.

Requirement / Characteristic	Modelling approach
Compositions	
Limited availability	✓
Types	Heterogenous
Ordering of train units	-
Orientation of compositions	✓
Restaurant wagon allocation	Distance-based
Turns	
Predetermined	-
Flexible	✓
Deadheading	-
Operational	
Maintenance	Distance-based
Depot capacity	-
Depot topology	-
Personnel availability	-
Planning	
Standard week	✓
Exception week	✓
Limited multiweek rotations	✓
Planning horizon	7-days (weekly)
Cyclicity	cyclic/acyclic
Quality-related	
Regularity	-
Robustness	-

Table 2: Characteristics of RSRPs considered in this thesis. Inclusion in the models is indicated by a checkmark (✓) and exclusion with a dash (-). Textual entries describe the modelling approach.

Table 2 presents requirements and characteristics considered in this thesis alongside other requirements which fall outside of the scope of this thesis or are not required for the optimization models. The selected requirements are discussed in detail in following subsections.

2.5.1 Planning horizon

The rotations are planned on a 7-day weekly basis. Two types of rolling stock rotation plans exist at VR. The first is planning rotations for a standard week, which is then repeated over multiple weeks. Therefore, the rotation plans for standard weeks must be cyclic.

The second type of plans are exception week plans, which include changes to the timetable w.r.t. the standard week's timetable. These changes can be due to holidays, track work or extra trains being operated for events. Each exception week is planned separately for one week at a time and is therefore acyclic. As discussed in [4], rotation plans for these exception weeks at VR are currently manually planned without mathematical optimization. The prior and subsequent weeks w.r.t. the exception week are assumed to follow the standard week plan and therefore the plans must be compatible over the week changes.

2.5.2 Multiweek rotations

To ease planning and operation, each rotation should only span a limited number of weeks. This is typically two to three weeks. In addition, often the timetable does not even permit to create RSR plans in which all rotations would repeat over exactly one week. Therefore, multiweek rotations must be allowed but also limited in length. The multiweek property is illustrated in Figure 3, where compositions 2.1 and 2.2 are part of a 2-week rotation. Composition 2.1 will operate the trains of the upper line during the first week and the trains of the lower line during the second week and composition 2.2 will do vice versa.

It is important that compositions in the same multiweek rotation are of the same type and capacity. Thus, for example a 10-week rotation would require 10 of the same type of compositions allocated to that rotation. The shorter the rotations, the more different types of composition can be used. To the best of the author's knowledge, limiting the length of multiweek rotations has not been addressed before in the existing literature on rolling stock rotations.

In exception week planning, multiweek rotations are not considered, as the exception week plan is acyclic. Instead, one must make sure the location and capacity of the composition before the exception week, i.e. at the end of the standard week, match the start of the exception week and that at the end of the exception week, the standard week can be continued. While these constraints are modelled in this thesis on a coarse level, manual planning is also necessary to make adjustments, especially when the exception weeks deviate a lot from the standard week.

2.5.3 Maintenance requirements

Compositions must undergo regular maintenance, and thus this requirement must be accounted for in the planning of rolling stock rotations. In this thesis, the maintenance requirement is distance-based. Maintenance of a composition is assumed to be possible during a turn at Helsinki station, if the duration of the turn is at least n hours. While some RSRP models [37, 38] explicitly allocate the maintenance slots to optimize their use, the models developed in this thesis must only coarsely model the maintenance requirements via a single distance-based limit U_m, similar to [37]. This ensures maintenance opportunities occur often enough for each composition.

2.5.4 Compositions and turns

The models developed in this thesis consider a few train-specific requirements. Firstly, the composition operating a train with distance over $L_{R,km}$ should include, if possible, a restaurant wagon. That is, the composition operating this train should be an IC-composition with an ERd-wagon or a Pendolino composition, as it always includes a restaurant wagon. Secondly, in the exception week model, the locations and types of compositions during the prior and subsequent week must be taken into account.

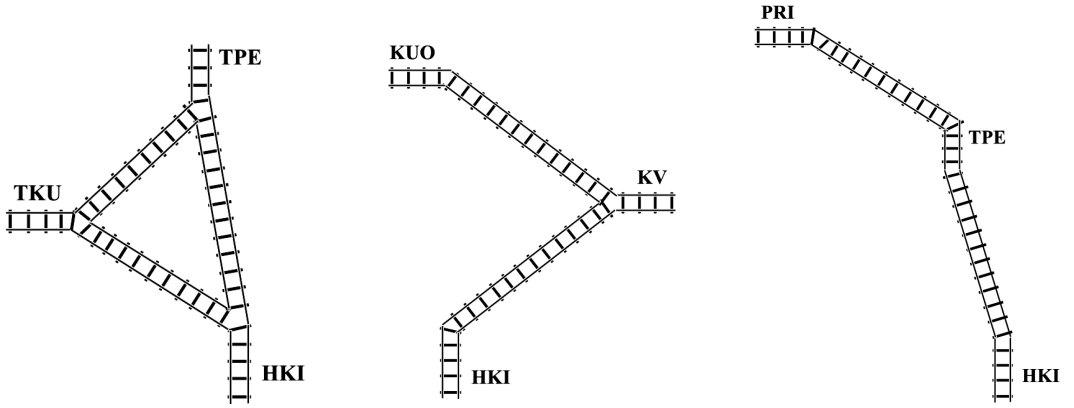
The availability of each type of composition and wagons described in Section 2.4 is assumed to be limited and consequently not all demand can necessarily be met. Therefore, the models must allow the demand of trains to be higher than the capacity of the composition operating the train, while penalizing such insufficient capacity. While the primary objective of the models is not to minimize the number of compositions used, the models will utilize less compositions than available, if that is optimal.

It is required that there is a turn that takes place at Helsinki station at least once per week in each rotation. This is due to Helsinki being the depot for wagons. At the depot, wagons are maintained and extra Ed-wagons can be added to or removed from IC-compositions. While changing the number of Ed-wagons on an IC-composition during the rotation is not modelled here, the possibility for these changes allows finer allocation of Ed-wagons closer to actual operation. Therefore, it is highly in the railway operator's interest to make sure RSRs are such that each rotation includes at least one long enough Helsinki turn during the week.

The turns considered in the models developed in this thesis are flexible and thus are determined as part of the optimization. Deadheading is not allowed and therefore all (potential) turns occur between trains arriving and departing at the same station. The (potential) turns differ by station and duration and some turns might be operationally undesirable. For example, a turn duration may be insufficient for relocating a composition to a nearby rail yard, but is long enough that the composition occupies the platform at the station blocking it from other compositions' use. Therefore, a penalty value is associated with each turn duration at a specific station, depending on these qualities. In general, use of short turn durations tend to increase cumulative delays and thus they are undesirable as they make the rotations less robust. This approach enables to implicitly incorporate robustness to the model.

2.5.5 Composition orientation

A key constraint in planning of rolling stock rotations for long-distance trains at VR is the orientation of the compositions when departing and arriving at Helsinki station. The orientation of compositions units has been previously considered in RSRP models for the purpose of coupling of train units [27, 39]. However, in the context of Finland's long-distance trains, only the orientation of whole compositions at Helsinki station is of interest. Specifically, the composition must be oriented such that the extra class wagon of the composition is located nearest to the station building. In addition to customer satisfaction, as Helsinki station is a dead-end station, the locomotive should not be situated at the end of the station to allow for easy change of locomotive, if needed.



(a) Railway topology of Helsinki (HKI), Turku (TKU) and Tampere (TPE).

(b) Railway topology of Helsinki (HKI) to Kuopio (KUO) via Kouvola (KV).

(c) Railway topology of Helsinki (HKI) to Tampere (TPE) to Pori (PRI)

Figure 5: Examples of railway topology in Finland.

It is assumed that any composition departing from Helsinki station is oriented in the correct way, and thus only the changes in orientation before returning to Helsinki station must be considered. The orientation of a composition can change either during the operation of a train or during a turn between operating two trains. For example, orientation of the composition changes during a train from HKI (Helsinki) to KUO (Kuopio) at KV station (Kouvola) (Figure 5b) and orientation also changes during a turn between operating trains HKI-KUO and KUO-HKI (Figure 5b). It is also possible that during a turn, the orientation does not change. An example of this is a turn between trains HKI-TPE and TPE-PRI (Pori) (Figure 5c). Operating the trains HKI-TKU, TKU-TPE and TPE-HKI (Figure 5a) results in the composition being oriented incorrectly when arriving to Helsinki and thus this operating order is not feasible. As a composition only has two possible orientations, the orientation constraint can be implemented through the parity of the total orientation changes during the rotation. In particular, the orientation must change an odd number of times between departing and arriving to Helsinki station.

3 Methods

This section presents the mathematical framework used to model and solve the RSRPs considered in this thesis. The problems are formulated using a connection-based multicommodity flow approach. First, the theoretical background of multicommodity flows is introduced and adapted for the standard and exception week variants of the RSRP. In addition, the cut separation method used in the models is described.

3.1 Multicommodity flow theory

Multicommodity flow formulations are widely used in communication and transport planning as they have an underlying network and an objective of routing commodities [40, 33]. Fioole et al. [7] proposed an edge-based flow problem with integer variables to model the rolling stock rotation problem. In this thesis, a similar approach is taken by formulating the models with connection-based [31, 21] multicommodity flows, consistent with other RSRP models, such as the hypergraph [30] and composition models [7, 20].

A multicommodity flow problem (MCFP) consists of routing multiple types of commodities through a network. The different commodities flow through the network via shared capacited edges of the underlying network, and thus are dependent on each other. The definition for multicommodity flows used in this thesis is adapted from one of the earliest formulations of the MCFP by Ahuja et al. [40] in 1993 and from Salimifard et al. [33]. Two equivalent formulations of the MCFP exist in the literature; an edge-based (arc-based) [32] and path-based [33] formulation. The edge-based approach is adopted in this thesis. First, a general formulation of the MCFP is presented and then applied to the RSRP.

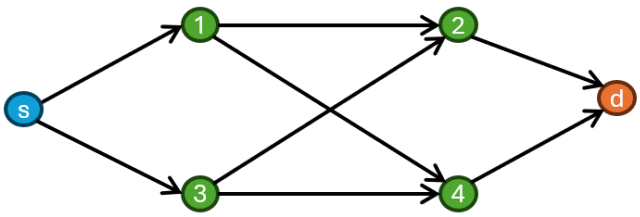


Figure 6: Network G with source node $s \in$ S, destination node $d \in$ D and regular nodes (green). The black arrows correspond to directed edges between the nodes.

Starting with formal definitions for a directed network, directed path and cycle adapted from [41].

Definition 6 (Directed network)**.** *A directed network G=(V,E) consists of a set V of elements together with a subset E of the ordered pairs* (i, j) *where* $i, j \in V$*. Elements of set V are called nodes and elements of set E edges.*

Definition 7 (Directed path)**.** *Let G=(V,E) be a directed network and let* $\{v_1, \ldots, v_n\} \subseteq V$ *be a sequence with distinct nodes. A sequence is a directed path if* $(v_i, v_{i+1}) \in E$ *for*

all $i \in 1, \ldots, n-1$. *The cost of a path* $\{v_1, \ldots, v_n\}$ *is defined as* $\sum_{i=1}^{i=n-1} c(i, i+1)$, *where c(i,j) is the cost of edge (i,j).*

Definition 8 (Cycle)**.** *Let G=(V,E) be a directed network. A directed path* $\{v_1, \ldots, v_n\} \subseteq V$ *is called a cycle, if* $n \geq 2$ *and* $v_1 = v_n$.

The MCFP is formally defined on a directed network adapted from [40]. In addition, in Figure 6 a graphical representation of the problem is presented. In this thesis, in the MCFP it is assumed that all of a specific commodity must be routed through the same path.

Definition 9 (MCFP)**.** *Let K be a set of commodities and let* q_k *denote the amount of commodity* $k \in K$. *Let G=(V,E) be a directed network with a capacity function u:* $E \to \mathbb{R}$ *and commodity-specific cost function* c^k*:* $E \to \mathbb{R}$. *The multicommodity flow problem (MCFP) consists of routing all commodities* $k \in K$ *from their source nodes* $s_k \in V$ *to destination nodes* $d_k \in V$ *through the directed network G where different commodities may flow through the same edges. The commodities must be routed such that, for every edge* $(i, j) \in E$, *the total flow of all commodities through* (i, j) *does not exceed its capacity* $u(i, j)$ *while total cost of routing commodities through the network is minimized. The cost of routing a commodity is defined as the cost of path it takes. Each type of commodity* $k \in K$ *is assigned a single directed* $s_k - d_k$ *path.*

The problem is formulated as a Mixed Integer Linear Programming (MILP) problem (1) with binary decision variables $f^k_{i,j}$, which denote whether commodity k flows through edge (i, j). Parameter q_k denotes the amount of commodity k to be routed.

$$\min \quad \sum_{k \in K} \sum_{(i,j) \in E} \mathrm{c}^k_{(i,j)} f^k_{(i,j)} \tag{1a}$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in E} f^k_{(i,j)} - \sum_{j:(j,i) \in E} f^k_{(j,i)} = \begin{cases} 1, & \text{if } i = d_k \\ -1, & \text{if } i = s_k \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in V, k \in K, \tag{1b}$$

$$\sum_{k \in K} q_k f^k_{(i,j)} \leq \mathrm{u}_{(i,j)} \qquad \forall (i, j) \in E, \tag{1c}$$

$$f^k_{(i,j)} \in \{0, 1\} \qquad \forall (i, j) \in E, k \in K \tag{1d}$$

The objective function (1a) minimizes the total cost of routing the commodities through the network. Flow conservation is ensured with constraint (1b). At intermediate nodes, the inflow equals outflow of each commodity, ensuring no loss of commodities during the routing. At the source node s_k, the outflow of each commodity k equals 1, as each commodity is treated as a whole, making the flow balance of the node -1. Similarly, at the destination node d_k, the inflow of each commodity k equals 1. The amount of commodity k to be routed is denoted by q_k, which is taken into account in the capacity constraint. The capacity constraint (1c) ensures the flows of

the commodities routed through each edge (i, j) do not exceed the capacity of that edge $u_{(i,j)}$. Finally, flow variables of each edge are defined as binary (1d).

To apply this formulation to the RSRP, additional constraints are required to ensure operational requirements are fulfilled. While this MCFP formulation does not allow each commodity to be split and routed along multiple routes, the formulation does not impose constraints on the amount of commodity passing through a given node.

3.2 Multicommodity flow formulation of the RSRP

MCFP element	RSRP element
Node	Train
Edge	Potential turn
Flow	Turn
Commodity	Composition
Path/Cycle	Rotation

Table 3: MCFP to RSRP mapping.

The RSRP can be formulated as a MCFP by establishing a mapping between the two problems. Table 3 illustrates this mapping. Each node in the network represents a train and each edge corresponds to a potential turn between trains. A flow through an edge corresponds to a turn between trains in the solution. The commodities to be routed are the compositions. The compositions must be routed through the networks such that exactly one composition is routed through each train. In the case of exception weeks, a rotation corresponds to a path from a source to a destination in the network. "Dummy"-turns from source nodes to destination nodes can be added to account for unused compositions during the exception week. With the standard week RSRP, source and destination nodes are removed and back-in-time edges are added instead such that a cycle in the updated network corresponds to a rotation. Similar approaches are used in train assignment [21], train unit scheduling [32] and scheduling of electric buses [15].

The two RSRP considered in this thesis are formally defined in their simplest form as

Definition 10 (RSRP for standard week)**.** *Let G=(V,E) be a directed network where nodes represent trains and edges represent potential turns between trains. Let K be a set of the different composition types.*

The RSRP for standard week consists of selecting a set of cycles such that each node is included in exactly one cycle. Each cycle represents a rotation. Let n be the length of a rotation (cycle) in weeks. To operate the rotation, n compositions of type k must be assigned to the rotation.

and

Definition 11 (RSRP for exception week)**.** *Let G=(V,E) be a directed network where nodes represent trains and edges represent potential turns between trains. Let K be a set of available compositions. Source nodes $s_k \in V$ represent the last train operated during the prior week by each composition and destination nodes $d_k \in V$ represent the first train to be operated by each composition during the subsequent week.*

The RSRP for exception weeks consists of routing each commodity k through network from source node to destination node such that each intermediate node is covered exactly once.

Next, the RSRP is formulated as a MILP model derived from model 1. To ensure each train is operated, a train covering constraint is formulated

$$\sum_{k \in K} \sum_{i:(i,j) \in E} f^k_{(i,j)} = 1 \quad \forall j \in V \setminus \{s, d\} \tag{2}$$

, which requires the inflow of each node equal to 1. The addition of this constraint transforms the standard MCFP to a flow-covering problem. The quantity of each composition (commodity) to be routed is exactly 1 and the capacity of each potential turn (edge) is also 1. As all edges are between trains, the train covering constraint also constraints the edge capacities and thus constraint 1c can be omitted from the RSRP formulation.

The RSRP with sources and destination formulated via multicommodity flows becomes

$$\min \quad \sum_{k \in K} \sum_{(i,j) \in E} c^k_{(i,j)} f^k_{(i,j)} \tag{3a}$$

$$\text{s.t.} \quad \sum_{j:(i,j) \in E} f^k_{(i,j)} - \sum_{j:(j,i) \in E} f^k_{(j,i)} = \begin{cases} 1, & \text{if } i = d_k \\ -1, & \text{if } i = s_k \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in V, k \in K, \tag{3b}$$

$$\sum_{k \in K} \sum_{i:(i,j) \in E} f^k_{(i,j)} = 1 \quad \forall j \in V \setminus \{s, d\}, \tag{3c}$$

$$f^k_{(i,j)} \in \{0, 1\} \quad \forall (i, j) \in E, k \in K. \tag{3d}$$

The objective (3a) minimizes the operational costs of routing compositions through the network. The cost of a turn $c^k_{(i,j)}$ is defined as the cost of operating train j. The flow conservation constraint (3b) remains the same as in the standard formulation. Constraint (3c) guarantees each train is operated by one composition and (3d) that each composition remains whole and each turn is used at most once.

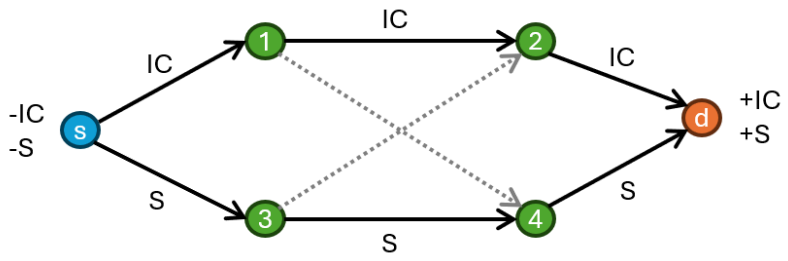


Figure 7: Example solution to a simple RSRP with two composition (IC and S) and four trains (green nodes) represented as a MCFP. Each numbered node corresponds to a train between two cities. Black edges represent the turns selected in the solution and dashed edges unused potential turns. The IC-composition operates trains 1 and 2, while the S-composition operates trains 3 and 4 in this solution.

In Figure 7, a solution to the rolling stock rotation problem with source and destination nodes represented as a MCFP is illustrated. Two compositions are available at the source node (blue), which must be routed through the network. All nodes (trains) must be visited exactly once, therefore, each node must have exactly one composition flowing into it and one flowing out of it. The balances of the source and destination nodes are also illustrated. The source node has a negative balance (only flows out), and the destination node has a positive balance (only flows in). In the example solution denoted by black edges, the IC-composition flows through nodes 1 and 2, meaning that it operates train 1 followed by train 2 before reaching the destination node. Similarly, the Pendolino (S) composition flows through nodes 3 and 4, thereby operating those trains. The dashed edges represent unused potential turns in the network.

The previous example uses source and destination nodes. These nodes can be directly applied when modelling the RSRP for an exception week. The source nodes represent the last trains operated by each composition at the end of the prior week. Similarly, the destination nodes represent the first trains to be operated by the compositions during the subsequent week. As the final trains during the prior week terminate at multiple stations, multiple source nodes are created, one for each composition and its corresponding final train. Similarly, multiple destination nodes are introduced for each composition and its corresponding first train during the subsequent week. The intermediate nodes correspond to all the trains that must be operated during the exception week.

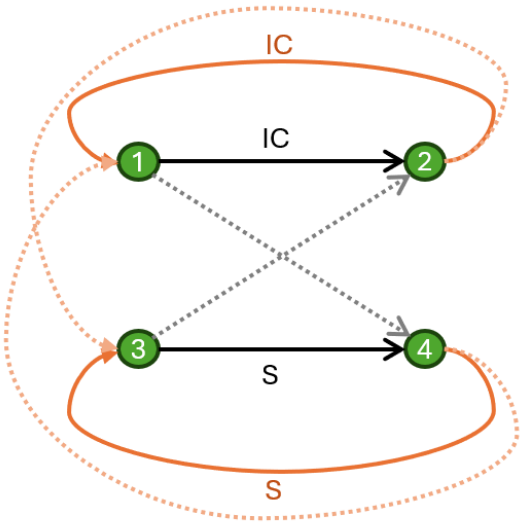


Figure 8: Example solution to a RSRP with back-in-time turns. Orange edges correspond to back-in-time turns, black edges to regular turns and dashed edges are unused potential turns. IC-composition operates trains 1 and 2 repeatedly and S-composition operates trains 3 and 4 repeatedly.

With standard weeks, the source and destination nodes are removed and replaced with back-in-time edges to enforce periodicity. These edges connect the end of the planning horizon (week) to its beginning, forming a link between the trains at the end and the beginning of the week. As a result, feasible rolling stock rotations correspond to node-disjoint cycles in the network i.e., cycle covers. This structure is illustrated in Figure 8, where the back-in-time edges are drawn in orange. In addition to enforcing periodicity, the back-in-time edges are used to limit the number of each composition type available during the standard week and thus replace the explicit consideration of each composition by source and destination nodes. Consequently, the back-in-time edges enforce cyclicity of the rotations and ensure conservation of the compositions.

3.3 Cut separation

An infeasible path cut separation method was applied to maintenance constraints in the RSRP in [42]. In this approach, cuts are dynamically created and added to the MILP formulation, in order to exclude infeasible subpaths from candidate solutions to the RSRP. A similar technique is applied in this thesis to eliminate rotations that do not include a stop at Helsinki at least once every week as well as to exclude solutions with rotations which exceed the given maximum allowed rotation length.

Let $R = \{R_1, R_2, ..., R_n\}$ denote the rotations of a feasible solution to the RSRP. Each rotation is composed of a set of turns used $\{(t_1, t_2), (t_2, t_3), \ldots, (t_{m-1}, t_m)\}$ and is operated by a composition c. The turns belonging to different rotations are disjoint. If a rotation violates the Helsinki visit constraint or exceeds the maximum allowed

rotation length, the rotation is excluded by adding the constraint

$$\sum_{(t,t')\in R_i} f_{(t,t')} \leq |R_i| - 1 \tag{4}$$

to the problem, where $|R_i|$ is the number of turns in the rotation R_i. This constraint ensures this specific rotation R_i will not be a part of any feasible solution. These cuts generated by the code during the solving process are also referred to as *lazy constraints* [43]. This terminology will be used throughout the thesis.

4 Models and solution methods

In this section, the notation used in the optimization models is defined, after which the models are derived and discussed in detail. Subsequently, four alternative solution methods which aim to reduce computation time are introduced and motivated. Finally, test sets used for validating the models and comparing the proposed solution methods are described.

4.1 Model notation

The notation used to define the models in this thesis are summarized in three tables. From now on, flows are referred to as turns. Variables and functions are denoted by *italics* while sets and indices are denoted in roman, upright text.

The composition indices for exception weeks behave slightly differently to the standard week model. In the standard week, each distinct composition type is a unique element (see Table 1), resulting in $|C| = 8$ different compositions. While the exception weeks use the same composition types, the individual compositions are additionally numbered. This enables keeping track of each individual composition during the optimization process, resulting in $|V| = |IC| + |S|$ composition elements. To make this distinction clear, compositions are notated by v instead of c in the exception week model.

Symbol	Meaning
Sets and elements	
$t \in T$	Trains in timetable
$s \in S$	Source trains
$d \in D$	Destination trains
$v \in V$	Compositions in the exception week
$c \in C$	Composition types in the standard week
$t \in T_R \subseteq T$	Trains with $Km(t) \geq L_{R,km}$
$(t, v^*) \in T_v^* \subseteq T \times V$	Trains with a predefined required composition in exception weeks
$(t, t') \in F \subseteq T \times T$	Potential turns between trains T
$(s, t) \in F_S \subseteq S \times T$	Potential turns from source trains S
$(t, d) \in F_D \subseteq T \times D$	Potential turns to destination trains D
$(t, t') \in F^* \subseteq F$	Back-in-time turns, defined for standard week
$(t, t') \in F_P$	Set of preferred turns

Table 4: The sets and elements used in the models.

Table 4 summarizes the sets and elements used in the model. The set T contains all the timetabled trains in the week for which rolling stock rotations must be planned. Each train t in the set represents a unique train which is scheduled once during the week. Each train has a departure and arrival station, departure and arrival time,

duration, cost per each composition type, distance in kilometres, in addition to an estimate of the passenger demand and average revenue per passenger. The source trains $s \in S$ and destination trains $d \in D$ are used in the exception week model, where they represent the prior and subsequent standard week's last and first train, respectively. For convenience, the set T_R is defined, which includes the trains whose distance exceeds the restaurant wagon kilometre limit $L_{R,km}$. A restaurant wagon is enforced to be a part of the composition operating each of these trains.

As previously described, the notion of compositions differs slightly between the models. In the exception week model, the compositions $v \in V$ represent individual physical compositions. Each composition has a specific type and must be assigned to exactly one rotation. Consequently, multiple compositions in the set V may be of the same type. In the standard week model, the compositions C represent types of available compositions, with each composition c being of a unique type. Each type c may be used for multiple rotations, subject to the upper limits on the total availability of that specific composition type.

The set of turns F contains all potential (feasible) turns between trains. Since deadheading is not allowed in the RSRPs considered in this thesis, a turn connects two trains for which the first train arrives and the second one departs from the same station. The set of potentail turns is generated prior to optimization, which is described in more detail in Section 4.4.1. The standard week model additionally uses back-in-time turns $F^* \subseteq F$, which allows to connect the end of the week to the beginning, ensuring periodicity of the rotation. The set of preferred turns F_P is a set of turns used in the prior week's RSR. This set is used in generating an initial feasible solution for the exception week model, as minimizing deviations from the prior week is desired.

Symbol	Meaning	Domain
Decision Variables		
$a_{t,c}$, $a_{t,v}$	Equal to 1 if train t is operated by composition c (standard week) or v (exception week), 0 otherwise	{0,1}
$f_{(t,t'),c}$, $f_{(t,t'),v}$	Equal to 1 if composition c (or v) operates train t′ immediately after train t, 0 otherwise	{0,1}
$f_{(t',t)}$	Binary auxilary variable equals 1 if turn (t,t') is selected to a rotation, 0 otherwise	{0,1}
o_t	Number of orientation changes by the composition before operating train t	$\mathbb{N}$
$m_{t,c}$, $m_{t,v}$	Number of estimated passengers exceeding the capacity of the composition c (or v) operating train t	$\mathbb{N}$
r_t	Equal to 1 if there is a maintenance opportunity during the turn before train t, 0 otherwise	{0,1}
d_t	Distance travelled after maintenance opportunity by composition operating train t before train t	$\mathbb{R}_{\geq 0}$
d_t^*	Distance travelled by composition operating train t exceeding the maintenance opportunity limit U_m	$\mathbb{R}_{\geq 0}$
d_t^r	Distance travelled by a composition without a restaurant wagon operating train t exceeding the restaurant kilometre limit $L_{R,km}$	$\mathbb{R}_{\geq 0}$
$n_{t'}$	Auxilary variable used to ensure the parity of composition orientations at Helsinki station	$\mathbb{N}$

Table 5: Decision variables used in the models.

The decision variables used in the model are presented in Table 5. Similar to [7], the main decision variables $a_{t,c}$ and $f_{(t',t),c}$ (or $a_{t,v}$ and $f_{(t',t),v}$ for the exception week model) determine whether a given train is operated by a specific composition c (or v) and whether a turn between two trains is selected in the solution. Additional variables are included to model operational constraints related to allocation of restaurant wagons, maintenance requirements and composition orientation. Slack variables are defined for the penalty function method, described in Section 4.3.2. Finally, variables capturing the estimated missed demand of each train are defined for the objective function.

Symbol	Meaning	Domain
Parameters		
D(t)	Estimated passenger demand for train t	$\mathbb{N}$
dep(t)	Departure station of train t	
OC(t,c), OC(t,v)	Cost of operating train t with composition c (or v)	$\mathbb{R}_{\geq 0}$
A(t)	Average revenue per passenger on train t	$\mathbb{R}_{\geq 0}$
C(c), C(v)	Passenger capacity of composition c (or v)	$\mathbb{N}$
Km(t)	Distance of train t in kilometres	$\mathbb{R}_{\geq 0}$
H(t,t′)	Equal to 1 if turn (t,t′) occurs at Helsinki station, 0 otherwise	{0,1}
M(t′,t)	Equal to 1 if maintenance is possible during the turn (t′,t), 0 otherwise	{0,1}
TP(t, t′)	Penalty cost for using turn (t,t′)	$\mathbb{R}$
P_d(s)	Distance accumulated after maintenance opportunity during prior week before train s	$\mathbb{R}_{\geq 0}$
P_o(s)	Orientation parity before source train s	{0,1}
S_d(d)	Distance travelled subsequent week including train d before next maintenance opportunity	$\mathbb{R}_{\geq 0}$
S_o(d)	Required orientation parity subsequent week	{0,1}
Ed(c)	Number of Ed-wagons in composition c	$\mathbb{N}$
U_{IC}	Maximum number of composition of type IC available	$\mathbb{N}$
U_S	Maximum number of Pendolino (S) compositions available	$\mathbb{N}$
U_{Ed}	Maximum number of additional capacity Ed-wagons available for IC-compositions	$\mathbb{N}$
U_{ERd}	Number of ERd-wagons available for IC-compositions	$\mathbb{N}$
U_m	Maximum allowed distance between maintenance opportunities	$\mathbb{R}_{\geq 0}$
$L_{R,km}$	Maximum allowed distance of a train which can be operated with a composition without a restaurant wagon	$\mathbb{R}$
T(v), T(c)	Type of composition v (or c)	
$C_{R,a}$	Number of compositions or wagons of type a in rotation R	$\mathbb{N}$
(t, v*)	Predefined required composition type v* for train t	
M	Large constant used in Big-M constraints	$\mathbb{R}$
w_x	Penalty weight associated with slack variable x	$\mathbb{R}$

Table 6: Parameters used in the models.

The parameters used in the models are summarized in Table 6. These parameters

include values associated with trains and turns in addition to upper limits related to maintenance and composition availability. Global constants such as M and penalty weights are also included.

4.2 Mathematical formulation

4.2.1 Standard week model

The objective function of the RSRP for standard weeks (5) minimizes the total operational costs, lost revenue from missed demand and the total turn penalties.

$$\begin{aligned} \min \quad & \sum_{t \in T} \sum_{c \in C} a_{t,c} \cdot OC(t,c) \\ & + \sum_{t \in T} \sum_{c \in C} m_{t,c} \cdot A(t) \\ & + \sum_{(t,t') \in F} f_{(t,t')} \cdot TP(t,t') \end{aligned} \tag{5}$$

The constraints for the model are derived next. As in the multicommodity flow structure introduced in Section 3.1, each train should be operated by exactly one composition. This is ensured with the constraint

$$\sum_{c \in C} a_{t,c} = 1 \quad \forall\, t \in T.$$

Following the standard multicommodity flow structure, flow conservation constraints are included for each composition type. Each train must be allocated a composition from a prior train and the composition used to operate a train must turn to another train after operation. The constraints

$$\begin{aligned} a_{t,c} &= \sum_{t':(t',t) \in F} f_{(t',t),c} \quad \forall\, t \in T,\ c \in C \\ a_{t,c} &= \sum_{t':(t,t') \in F} f_{(t,t'),c} \quad \forall\, t \in T,\ c \in C \end{aligned}$$

ensure this. The auxiliary variables

$$f_{(t',t)} = \sum_{c \in C} f_{(t',t),c} \quad \forall\, (t,t') \in F$$

are defined for the turns. These are used to penalize undesirable turns in the objective function.

Estimated missed demand for each train-composition combination is defined as follows. The missed demand $m_{t,c}$ is calculated as the difference between demand of a train and the capacity of the allocated composition type. The missed demand of compositions not used for a train minimizes to zero, as $m_{t,c}$ is required to be nonnegative.

$$m_{t,c} \geq a_{t,c} \cdot D(t) - C(c) \qquad \forall\, t \in T, c \in C$$

To enforce the maintenance limits, a set of maintenance constraints are added to the model. The kilometres travelled by a composition between maintenance opportunities is limited by U_m. This is done via accumulating the kilometres travelled per train operated after a maintenance opportunity. First, let us define the variable r_t (Table 5), which indicates whether there was a maintenance opportunity during the turn prior to train t. This is the sum over the possible turns before train t. Only one turn prior to a train will be selected (for exactly one composition c) and $M(t',t)$ indicates whether maintenance is possible during this turn. This constraint is defined as

$$r_t = \sum_{t':(t',t)\in F} \sum_{c\in C} f_{(t',t),c} \cdot M(t',t) \qquad \forall\, t \in T.$$

The distance travelled between maintenance opportunities is tracked by defining the cumulative distance travelled by the composition operating train t prior to that train. This accumulated distance is defined for each train in the timetable. Constraint (6) defines the reset of the accumulated distance through the variable r_t described above. Constraint (7) defines how the the distance is accumulated between turns. If the turn (t, t') is selected ($f_{(t',t)}$=1) and there is not a maintenance opportunity before train t' ($r_{t'} = 0$), the accumulated distance before train t' equals the accumulated distance before train t plus the distance of train t, denoted by $Km(t)$. Finally, the accumulated distance must not exceed the maximum allowed distance between maintenance opportunities U_m. Therefore, in constraint (8), the distance accumulated before train t plus the distance of train t is constrained to be less than or equal to the maintenance limit U_m.

$$d_t \le M \cdot (1 - r_t) \qquad \forall\, t \in T \qquad (6)$$

$$d_{t'} \ge d_t + Km(t) - M \cdot (1 - \sum_{c\in C} f_{(t',t),c}) - M \cdot r_{t'} \qquad \forall\, (t,t') \in F \qquad (7)$$

$$d_t + Km(t) \le U_m \qquad \forall\, t \in T \qquad (8)$$

The consistent orientation of compositions at Helsinki station (HKI) is defined by three constraints. Similarly to the maintenance constraint, the orientations of the compositions are tracked by accumulating orientation changes. The first constraint (9) resets the orientation change accumulation at Helsinki station. If the turn between trains (t, t') is selected and the departure station of train t' is not Helsinki station, the orientation changes before train t' equals the orientation changes o_t before train t, plus the orientation changes during train t and orientation change during the turn $O(t, t')$. This is defined in constraint (10). The number of orientation changes before a turn at Helsinki station must be an odd number, defined in constraint (11). The if-statements in the constraints are implemented with indicator constraints. See Appendix A for details on a manual implementation.

$$o_t = 0 \text{ if } dep(t) = HKI \qquad \forall\, t \in T \qquad (9)$$

$$o_{t'} = o_t + O(t) + O(t,t') \text{ if } dep(t') \neq HKI \;\&\; f_{(t',t)} = 1 \qquad \forall\, (t,t') \in F \qquad (10)$$

$$o_t + O(t) = 2n_{t'} - 1 \text{ if } dep(t') = HKI \;\&\; f_{(t',t)} = 1 \qquad \forall\, (t,t') \in F \qquad (11)$$

Trains with a distance exceeding $L_{R,km}$ (T_R) are required to have a restaurant wagon. Thus, the composition operating these trains should either be a Pendolino (S) or an IC-composition with an ERd-wagon. This is defined by the constraint

$$\sum_{\substack{c \in C \text{ s.t.} \\ ERd,\ S \notin T(c)}} a_{t,c} \cdot Km(t) \leq L_{R,km} \qquad \forall\, t \in T_R. \tag{12}$$

Finally, the number of available base-composition and additional wagons are limited using back-in-time turns. Constraints (13) and (14) limit the number of available base-compositions of type IC and S, respectively. Similarly, the number of ERd-wagons is limited by summing the back-in-time flows corresponding to turns with an ERd-wagon (15) and setting the upper limit U_{ERd}. The Ed-wagons are limited by summing over the back-in-time turns using Ed-wagons and accounting for the number of Ed-wagon in each composition with $Ed(c)$ for each composition (16).

$$\sum_{(t,t') \in F^*} \sum_{\substack{c \in C \text{ s.t.,} \\ IC \in T(c)}} f_{(t,t'),c} \leq U_{IC} \tag{13}$$

$$\sum_{(t,t') \in F^*} \sum_{\substack{c \in C \text{ s.t.,} \\ S \in T(c)}} f_{(t,t'),c} \leq U_{S} \tag{14}$$

$$\sum_{(t,t') \in F^*} \sum_{\substack{c \in C \text{ s.t.,} \\ ERd \in T(c)}} f_{(t,t'),c} \leq U_{ERd} \tag{15}$$

$$\sum_{(t,t') \in F^*} \sum_{\substack{c \in C \text{ s.t.,} \\ Ed \in T(c)}} f_{(t,t'),c} \cdot Ed(c) \leq U_{Ed} \tag{16}$$

The domains of the variables for the constraints are defined as

$$\begin{aligned}
a_{t,c} &\in \{0,1\} && \forall\, t \in T, c \in C \\
r_t &\in \{0,1\} && \forall\, t \in T \\
f_{(t,t'),c} &\in \{0,1\} && \forall\, (t,t') \in F, c \in C \\
f_{(t,t')} &\in \{0,1\} && \forall\, (t,t') \in F \\
o_t, n_t &\in \mathbb{N} && \forall\, t \in T \\
m_{t,c} &\in \mathbb{N} && \forall\, t \in T, c \in C \\
d_t &\geq 0 && \forall\, t \in T.
\end{aligned}$$

Finally, lazy constraints are added dynamically during the solving process to exclude infeasible rotations R_i w.r.t. the Helsinki station turn constraint or exceeding the maximum rotation length requirement. The cuts

$$\sum_{(t,t') \in R_i} f_{(t,t')} \leq |R_i| - 1$$

are created dynamically during the solving process, ensuring the infeasible rotation R_i cannot be included for any composition c.

4.2.2 Exception week model

The exception week model is based on the source-destination multicommodity flow formulation, as the prior and subsequent week of the exception week must be accounted for. These surrounding weeks typically follow the RSR of a standard week. The prior and subsequent weeks determine the location of each composition at the beginning and end of the exception week. This can be modelled with the source and destination trains. A source train (s,v) is defined to be the prior week's last train of each rotation thus determining the composition's location and type. Similarly, the destination trains (d,v) are defined to be the subsequent week's first trains in each rotation which determine where each composition type must terminate at.

As each composition has accumulated distance travelled during the prior weeks, this must be taken into consideration in the formulation of maintenance constraints. Furthermore, this also applies to the subsequent week, as each composition has a preset plan with a given distance to be travelled before the next maintenance opportunity. In addition, the composition orientation must be tracked w.r.t. the plans of the prior and subsequent weeks to ensure correct orientation at Helsinki station.

The objective (17) for the exception week model is the same as for the standard week formulation minimizing total operational costs, lost revenue due to missed demand and the total turn penalty.

$$\begin{aligned} \min \quad & \sum_{t \in T} \sum_{v \in V} a_{t,v} \cdot OC(t,v) \\ & + \sum_{t \in T} \sum_{v \in V} m_{t,v} \cdot A(t) \\ & + \sum_{(t,t') \in F} f_{(t,t')} \cdot TP(t,t') \end{aligned} \tag{17}$$

The exception week problem formulation is similar to the standard week for the most part. Each train (timetabled trains, source trains and destination trains) must be operated by one composition (18). In addition, specific composition type requirements for trains T_v^* must be satisfied (19). These specific trains are the source trains $s \in S$ and destination trains $d \in D$ for which the standard week poses the composition type requirement. However, the same constraint could be used for any train during the week.

$$\sum_{v \in V} a_{t,v} = 1 \qquad \forall\, t \in \{T \cup S \cup D\} \tag{18}$$

$$a_{t,v^*} = 1 \qquad \forall\, (t, v^*) \in T_v^* \tag{19}$$

As in the standard multicommodity flow formulation, flow conservation constraints are defined for source trains (20), destination trains (21) and timetabled trains (22, 23)

for each composition v ∈ V.

$$a_{s,v} = \sum_{t:(s,t)\in F_S} f_{(s,t),v} \qquad \forall\, s \in S, v \in V \tag{20}$$

$$a_{d,v} = \sum_{t:(t,d)\in F_D} f_{(t,d),v} \qquad \forall\, d \in D, v \in V \tag{21}$$

$$a_{t,v} = \sum_{t':(t',t)\in F_S\cup F} f_{(t',t),v} \qquad \forall\, t \in T, v \in V \tag{22}$$

$$a_{t,v} = \sum_{t':(t,t')\in F\cup F_D} f_{(t,t'),v} \qquad \forall\, t \in T, v \in V \tag{23}$$

Again, auxiliary variables for the turns are defined for the objective function and other constraints.

$$f_{(t,t')} = \sum_{v\in V} f_{(t,t'),v} \qquad \forall\, (t,t') \in F \cup F_S \cup F_D$$

Similar to the standard week formulation, a constraint is defined for determining the missed demand of each train per composition type. The missed demand for a composition type not used for a train will minimize to zero.

$$m_{t,v} \geq a_{t,v} \cdot D(t) - C(v) \qquad \forall\, t \in T, v \in V$$

In contrast to the standard week, where lazy constraints were used to exclude rotations without the required weekly Helsinki visit, a constraint directly prohibiting such rotations is included. This is possible as each composition is modelled separately in the formulation and thus composition specific constraints can be used.

$$1 \leq \sum_{(t,t')\in F} f_{(t,t'),v} \cdot H(t,t') \qquad \forall\, v \in V$$

Maintenance constraints of the same form as in the standard week model are added (24, 26, 27, 28). The accumulated distance since the previous maintenance opportunity of source trains is defined with constraint (25) and the source trains are taken into account in the distance accumulation constraint (27). The distance before the next maintenance opportunity during the subsequent week is considered with constraint (29) for each destination train.

$$r_t = \sum_{t':(t',t)\in F\cup F_S} \sum_{v\in V} f_{(t',t),v} \cdot M(t',t) \qquad \forall\, t \in T \cup D \tag{24}$$

$$d_s = P_d(s) \qquad \forall\, s \in S \tag{25}$$

$$d_t \leq M \cdot (1 - r_t) \qquad \forall\, t \in T \cup D \tag{26}$$

$$d_{t'} \geq d_t + Km(t) - M \cdot (1 - \sum_{v\in V} f_{(t,t'),v}) - M \cdot r_{t'} \qquad \forall\, (t,t') \in F \cup F_S \cup F_D \tag{27}$$

$$d_t + Km(t) \leq U_m \qquad \forall\, t \in T \tag{28}$$

$$d_d + S_d(d) \leq U_m \qquad \forall\, d \in D \tag{29}$$

The composition orientation at Helsinki station (HKI) is defined with constraints similar to the standard week model (31, 32, 33). The orientation of each composition at the end of the prior week must be taken into consideration (30). The required orientation for the subsequent week is ensured with constraint (34). As previously, the if-statements are implemented with indicator constraints. See Appendix A for details on a manual implementation.

$$o_s = P_o(s) \qquad \forall\, s \in S \tag{30}$$
$$o_t = 0 \text{ if } dep(t) = HKI \qquad \forall\, t \in T \tag{31}$$
$$o_{t'} = o_t + O(t) + O(t,t') \text{ if } dep(t') \neq HKI \;\&\; f_{(t,t')} = 1 \qquad \forall (t,t') \in F \cup F_S \tag{32}$$
$$o_t + O(t) = 2n_{t'} - 1 \text{ if } dep(t') = HKI \;\&\; f_{(t,t')} = 1 \qquad \forall\, (t,t') \in F \tag{33}$$
$$o_t + O(t) + O(t,d) = 2n_d - S_o(d) \text{ if } f_{(t,d)} = 1 \qquad \forall\, (t,d) \in F_D \tag{34}$$

The final constraint (35) forces trains with a distance exceeding $L_{R,km}$ to have a composition with a restaurant wagon.

$$\sum_{\substack{v \in V \text{ s.t.} \\ ERd,\, S \notin T(v)}} a_{t,v} \cdot Km(t) \leq L_{R,km} \qquad \forall\, t \in T_R \tag{35}$$

The domains of the variables for the constraints are defined as

$$a_{t,v} \in \{0,1\} \qquad \forall\, t \in \{T \cup S \cup D\}, v \in V$$
$$r_t \in \{0,1\} \qquad \forall\, t \in \{T \cup D\}$$
$$f_{(t,t'),v} \in \{0,1\} \qquad \forall\, (t,t') \in F \cup F_S \cup F_D, v \in V$$
$$f_{(t,t')} \in \{0,1\} \qquad \forall\, (t,t') \in F \cup F_S \cup F_D$$
$$m_{t,v} \in \mathbb{N} \qquad \forall\, t \in T, v \in V$$
$$o_t, n_t \in \mathbb{N} \qquad \forall\, t \in \{T \cup D\}$$
$$d_t \geq 0 \qquad \forall\, t \in \{T \cup D\}$$

4.3 Solution methods

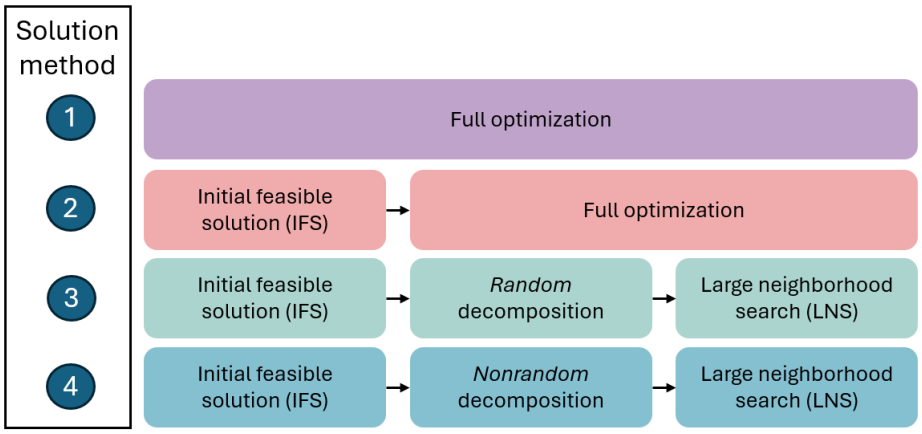


Figure 9: Visualization of the solution methods evaluated in this thesis.

The developed MILP models must be solved within a reasonable time limit of 2 hours (120 minutes) for the planners to be able to utilize the optimization in their workflow. Due to the size and complexity of the model, solving the problem directly with a commercial MILP solver may be computationally challenging. Therefore, four alternative solution methods are developed and evaluated. The alternative solution methods are illustrated in Figure 9. Method 1 consists of simply solving the full model directly. However, given the size and nature of the RSRP in this thesis, this approach might not yield high-quality or even feasible solutions within the time limit. To improve the solving process, the alternative solution methods utilize an initial feasible solution (IFS) to warm start the solver. The second method utilizes the IFS and solves the full problem with a warm start from the IFS. Furthermore, decomposition and large neighborhood search methods are implemented in solution methods 3 and 4, respectively, to accelerate the solving process of finding high-quality feasible solutions.

4.3.1 Initial feasible solution generation

Providing an initial feasible solution (IFS) as a warm start for a solver can significantly improve the solving time of a MILP model [44]. In this thesis, the approach used to generate an initial feasible solution is to solve a reduced version of the RSRP MILP formulation in additional wagons, maintenance constraints and restaurant wagon requirements are ignored. These omitted hard constraints are formulated as soft constraints in the full model using penalty functions, described in the following Section 4.3.2.

In more detail, the goal of generating an IFS is to find a feasible RSR for the given timetable while ignoring the details on compositions and maintenance constraints. The generated IFS is feasible w.r.t. the train assignment, flow conservation and base-composition availability, but not necessarily w.r.t. the maintenance, orientation

or composition type requirements. This aligns with an approach of optimizing RSRPs from a coarse, higher level solution to a finer solution with more details [5].

To generate an IFS to the RSRPs considered in this thesis, a reduced IP formulation is formulated for both standard week and exception week models. The key feasibility requirements for a RSR for the standard week are that each train is operated by exactly one composition, the selected turns between train are feasible and the number of available compositions is not exceeded. All of these requirements can be fulfilled by only utilizing the turn variables ($f_{(t,t')}$ and $f_{(t,t'),c}$) of the full model for the standard week RSRP. Formally, this is expressed as the formulation (36).

The IFS IP model minimizes the turn penalties (36a). Minimizing turn penalties biases the IFS to realistic rotations which utilize the preferred turns w.r.t. turn penalties. This solution is a good-quality MIP start to the full model. In the IFS model, it is required that each train is operated by one composition, that is, each train has exactly one turn-composition pair flowing into the train (36b). For flow conservation, each train must have an equal number of compositions flowing in and out of the train (36c). In practice, only the base-types of compositions are considered (IC and S) and all additional wagons are ignored. This makes sure the rotation lengths are compatible with the available compositions. Constraint (36d) defines variables for the objective function. The number of each type composition available is limited through the back-in-time turns (36e), similar to the full model. All of the variables used in the IFS generation model are binary.

$$\min \quad \sum_{(t,t')\in F} f_{(t,t')} \cdot TP(t,t') \tag{36a}$$

$$\text{s.t.} \quad \sum_{t':(t,t')\in F} \sum_{c\in C} f_{(t,t'),c} = 1 \qquad \forall\, t \in T, \tag{36b}$$

$$\sum_{t':(t',t)\in F} f_{(t',t),c} = \sum_{t':(t,t')\in F} f_{(t,t'),c} \qquad \forall\, t \in T, c \in C, \tag{36c}$$

$$\sum_{c\in C} f_{(t,t'),c} = f_{(t,t')} \qquad \forall\, (t,t') \in F, \tag{36d}$$

$$\sum_{(t,t')\in F^*} \sum_{c\in C} f_{(t,t'),c} \leq U_c \qquad \forall\, c \in C, \tag{36e}$$

$$f_{(t,t')} \in \{0,1\} \qquad \forall\, (t,t') \in F, \tag{36f}$$

$$f_{(t,t'),c} \in \{0,1\} \qquad \forall\, (t,t') \in F, c \in C \tag{36g}$$

The IP formulation for exception weeks (37) is more concise as the number of each composition type used is not explicitly limited, as the composition types are defined by the surrounding standard weeks. It is sufficient to require one turn from a source or timetable train to each timetabled or destination train (37b), such that each train has exactly one incoming turn and similarly exactly one outgoing turn (37c). This ensures each train is operated by one composition. As with the standard week IFS model, all the variables used for the exception week IFS generation IP are binary.

The objective of the exception week IFS model (37a) is formed of two parts. Firstly, the turn penalties are minimized as with the standard week IFS model. In addition, a

penalty is introduced for deviating from the set of turns F_P used by the surrounding standard week. This encourages the exception week solution to be structurally close to the standard week RSR solution, while also allowing for necessary deviations. It is assumed that the exception week is for the most part similar to the standard week and thus the solution for the standard week provides a good starting point for the solution of the exception week. Therefore let w_P denote the penalty associated with the IFS for the exception week not using a flow in the standard week.

$$\min \quad \sum_{(t,t')\in F} f_{(t,t')} \cdot TP(t,t') + w_p \sum_{(t,t')\in F_P} (1 - f_{(t,t')}) \tag{37a}$$

$$\text{s.t.} \quad \sum_{t':(t',t)\in F\cup F_S} f_{(t',t)} = 1 \qquad \forall\, t \in \{T \cup D\}, \tag{37b}$$

$$\sum_{t':(t,t')\in F\cup F_D} f_{(t,t')} = 1 \qquad \forall\, t \in \{S \cup T\}, \tag{37c}$$

$$f_{(t',t)} \in \{0,1\} \quad \forall\, (t',t) \in F \tag{37d}$$

Rotations without a turn at Helsinki are eliminated by dynamically adding the lazy constraints

$$\sum_{(t',t)\in R_i} f_{(t',t)} = |R_i| - 1$$

during the solving process, as with the previous models for infeasible rotations. The same cuts are used for both standard and exception week IFS models. These lazy constraints are also utilized in eliminating rotations exceeding the maximum rotation length in the standard week IFS model. Preliminary testing indicated that this can result in generating a large number of lazy constraints, which negatively impacts the performance. Therefore, the maximum rotation length parameters should be adjusted accordingly.

Through testing, the orientation constraint was found to be a heavy constraint for the solver when finding an initial feasible solution. However, in most cases, an initial feasible solution w.r.t. the orientation constraint can be found artificially by banning certain turns by which unwanted cycles can be excluded. In the processing pipeline, this can be interpreted as a preprocessing step. For example, this can be done by prohibiting all turns at Turku which have a train from Helsinki to Turku and a train from Turku to Tampere. The prohibited turns can still be present in a feasible solution to the full model and thus they should be added back to the available turns in the next stage of optimization along with the proper orientation constraints.

It is sufficient to provide a partial MIP start for the solver, as the Gurobi's MIP solver will try to construct a complete solution by inferring the remaining variables from the given information [44]. In case of the RSRP, providing the selected turns $f_{(t,t')}$ (or $f_{(t,t'),c}$ for the standard week) are enough for Gurobi MIP solver to construct the initial solution. In addition, the train variables $a_{t,c}$ can be inferred from the $f_{(t,t'),c}$ variables and also provided to the solver.

4.3.2 Penalty function method

The initial feasible solution (IFS) does not necessarily satisfy all hard constraints in either model. Specifically, the maintenance constraints are not guaranteed to be satisfied and as ERd-wagons are not included in the IFS generation for standard weeks, the constraint requiring all trains with over $\mathrm{L_{R,km}}$ kilometres to have a restaurant wagon is not satisfied. Furthermore, if one is unable to provide a list of prohibited turns to make the IFS feasible w.r.t. the orientation constraint, that constraint is also not guaranteed to be satisfied. However, by transforming the remaining unsatisfied hard constraints into soft constraint via a penalty function [45, p. 470], the initial solution will be feasible and allow to warm start the optimization. Formally, this can be defined as follows. Given an optimization problem P with hard constraints

$$\begin{aligned} \min \quad & c^\top x \\ \text{s.t.} \quad & Ax \leq b, \\ & x \geq 0, \end{aligned} \tag{P}$$

all hard constraints are transformed into soft constraints by introducing slack variables s to allow for violating of the restriction b (P^*). These slack variables are penalized in the objective function with the function f.

$$\begin{aligned} \min \quad & c^\top x + f(s) \\ \text{s.t.} \quad & Ax - s \leq b, \\ & x, s \geq 0 \end{aligned} \tag{P*}$$

An option for the penalty function f of the slack variables is to use a linear function $f : s \to ps$, where p is a constant. Thus, the penalty increases linearly as the constraint is violated. For binary variables, the slack variables are of discrete values and each violation ($s = 1$) results in a fixed penalty p being added to the objective.

As the initial feasible solutions for the RSRP presented here do not satisfy the maintenance constraints or kilometre-based restaurant constraints, these are modified to soft constraints. Slack variables $d^*_\mathrm{t} \geq 0$ and $d^r_\mathrm{t} \geq 0$ are added to the maintenance (8, 28) and restaurant constraints (12, 35), resulting in the maintenance constraint

$$d_\mathrm{t} + Km(\mathrm{t}) - d^*_\mathrm{t} \leq \mathrm{U_m} \qquad \forall\, \mathrm{t} \in \mathrm{T}$$

and restaurant wagon constraint

$$\sum_{\substack{\mathrm{c} \in \mathrm{C} \text{ s.t.} \\ \mathrm{ERd},\, \mathrm{S} \notin T(\mathrm{c})}} a_{\mathrm{t,c}} \cdot Km(\mathrm{t}) - d^r_\mathrm{t} \leq \mathrm{L_{R,km}} \qquad \forall\, \mathrm{t} \in \mathrm{T_R}.$$

The penalty term

$$\mathrm{w}_{d^*_\mathrm{t}} \cdot \sum_{\mathrm{t} \in \mathrm{T}} d^*_\mathrm{t} + \mathrm{w}_{d^r_\mathrm{t}} \cdot \sum_{\mathrm{t} \in \mathrm{T_R}} d^r_\mathrm{t} \tag{38}$$

is added to the objective which penalizes exceeding the given upper limit for maintenance $\mathrm{U_m}$ and restaurant wagon km $\mathrm{L_{R,km}}$ for the standard week formulation. The

same penalty terms are added to the exception week model's objective function (with replacing the composition indices from c to v). Each violated kilometre is penalized by a constant $w_{d_t^*}$ or $w_{d_t^r}$.

For the exception week model's destination train's maintenance constraint, the slack variable d_d^* is added and the constraint takes the form

$$d_d + S_d(d) - d_d^* \leq U_m \qquad \forall\, d \in D$$

In addition, the penalty terms

$$w_{d_d^*} \cdot \sum_{d \in D} d_d^* \tag{39}$$

and

$$w_{v^*} \cdot \sum_{(d,v^*) \in T_v^*} \sum_{\substack{v \in V \text{ s.t.} \\ v \neq v^*}} a_{d,v}$$

are added to the objective function of the exception week model to penalize exceeding the maintenance distance limitation during the subsequent week and deviations from the required composition type for the trains at the beginning of the subsequent week.

If one is unable to generate an IFS which satisfies the orientation constraint, it can be also transformed into soft constraint with a penalty function. As the orientation is only depended on parity, the added slack variable should be binary. See appendix (B) for details for the implementation.

4.3.3 Decomposition heuristics

Decomposition heuristics are widely used to reduce computational time when solving large-scale vehicle scheduling problems [8, 46]. In particular, Santini et al. [8] propose a route-based decomposition heuristic for vehicle routing problems which decomposes the full problem into smaller subproblems. They recommend utilizing an existing solution to form the subproblems, which ensures each subproblem will have at least one feasible solution. In route-based decomposition [8], these subproblems are formed by grouping the routes in the existing solution into smaller sets. Furthermore, the subproblems are recommended to be homogeneous and independent. Homogeneous subproblems retain the same structure and definition as the full problem, allowing to use the same model and solver as for the full problem. Independent subproblems have objectives and feasibility that do not depend on each other. This independence enables parallel solving of the subproblems, reducing the overall solving time. Finally, the solutions of the subproblems can be combined for a solution to the full problem.

Following this general principle, the route-based decomposition heuristic is adapted for the RSRP. Here, the RSRP is divided into subproblems based on the existing rotations $R_1, R_2, ..., R_n$ in a feasible solution. The resulting subproblems are homogeneous as they are of the same structure as the full problem. In a feasible solution, each rotation has a composition allocated to it. The allocated compositions of the rotations in a subproblem form the available compositions and ERd- and Ed-wagons to be used when solving the subproblem, making the subproblems independent. Notably,

while the number of base IC-compositions and wagons are limited, the number of each IC-composition type can vary during the subproblem solve, as that is not fixed.

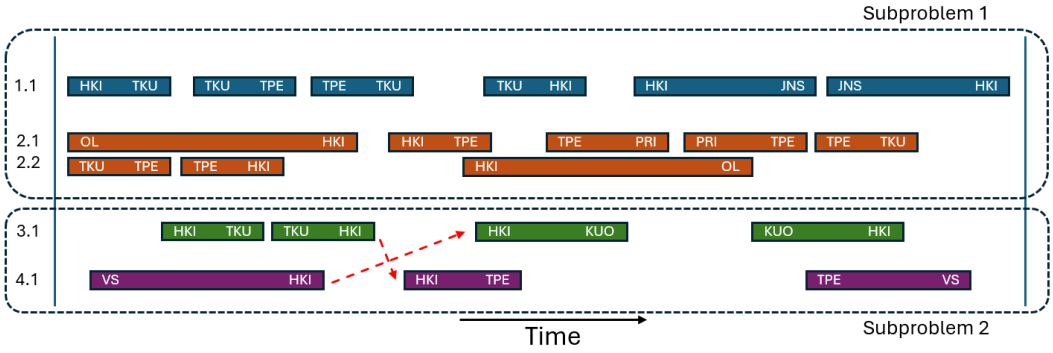


Figure 10: Decomposition of a feasible RSR solution to two subproblems that are reoptimized in parallel. In subproblem 2, the red arrows indicate potential turns (TKU-HKI → HKI-TPE and VS-HKI → HKI-KUO) which may replace the ones (TKU-HKI → HKI-KUO and VS-HKI → HKI-TPE) in the current solution. If compositions 3.1 and 3.2 are of the same type, these potential turns could be selected during the reoptimization, resulting in a new solution for subproblem 2 which would have a single rotation spanning two weeks.

The rotation-based decomposition is visualized in Figure 10, where the rotations in an existing feasible solution are grouped into two sets. As recommended in [8], the subproblems should be solvable to near-optimality in the given time limit, which can be determined experimentally. In the example, the both subproblems contain only two rotations, but in practice a subproblem of the RSRP would typically contain between 3 and 15 rotations.

When the decomposition method is used for standard week optimization and initialized with an IFS, none of the rotations in the IFS contain any ERd- or Ed-wagons. As a result, the subproblems formed solely on the IFS would not have access to use these wagons. To address this, a wagon-distribution step is added to the decomposition heuristic, in which the available wagons not used by any rotation are allocated to the subproblems, giving priority to subproblems containing IC-compositions. In addition, some rotations in the IFS or the current best solution may exceed the desired maximum rotation length. To accommodate this, the decomposition heuristic sets the maximum rotation length in each subproblem as the greater of the desired limit and the longest rotation present in the existing feasible solution of the subproblem.

When the decomposition heuristic is applied to exception weeks, the destination and source trains are included when forming decompositions of the exception week timetable. Their allocation is also based on the IFS.

Algorithm 1 Decomposition heuristic for RSRP

Input: Timetable and other data, composition type limits, max iterations I, time limit per iteration q_i

Output: RSR plan

1: Generate IFS $x^0 = \{\mathrm{R}_1, \mathrm{R}_2, ..., \mathrm{R}_n\}$ with n rotations for the RSRP
2: Set current solution $x^* \leftarrow x^0$
3: Set desired maximum rotation length y^*
4: Set iteration counter $i = 0$
5: **while** $i < I$ **do**
6: Divide n rotations in the current solution x^* into m sets $S = \{S_1, S_2, \ldots, S_m\}$
7: **for** $S_j \in S$ **do**
8: Set $y_j \leftarrow \max\left(\max_{R \in S_j} |R|, y^*\right)$
9: Set $U_{S_j,a} = \sum_{\mathrm{R} \in S_j} C_{\mathrm{R},a} \quad \forall\, a \in \{\mathrm{IC,S,ERd,Ed}\}$
10: **end for**
11: **for** $a \in \{\mathrm{ERd,Ed}\}$ **do**
12: **if** $\sum_{\mathrm{R} \in x^*} C_{\mathrm{R},a} < U_a$ **then**
13: Allocate U_a -$\sum_{R \in x^*} C_{\mathrm{R},a}$ to sets S
14: **end if**
15: **end for**
16: Solve subproblems in sets S in parallel with time limit q_i and upper limits y_j and $U_{S_j,a} \quad \forall\, a \in \{\mathrm{IC,S,ERd,Ed}\}$
17: Combine solutions $x^* = \sum_{j=1}^{j=m} x_j$
18: $i = i + 1$
19: **end while**
20: **return** x^*

The algorithm used for the decomposition heuristic is described in Algorithm 1. $C_{S_j,a}$ denotes the number of compositions or wagons of type a in the rotations of set S_j. First, an IFS solution $x^0 = \{\mathrm{R}_1, \mathrm{R}_2, ..., \mathrm{R}_n\}$ is generated (step 1) and set as the current solution x^* in step 2. The desired maximum rotation length y^* is also defined on line 3. The decomposition is repeated for I iterations or alternatively until a time limit is reached. In each iteration, the rotations in the current solution are divided into m homogeneous and independent subproblems $S = \{S_1, S_2, \ldots, S_m\}$ (step 6). For each subproblem S_j (step 7), the maximum rotation length y_j, and the upper limits on compositions and wagons ($U_{S_j,\mathrm{IC}}, U_{S_j,\mathrm{S}}, U_{S_j,\mathrm{ERd}}, U_{S_j,\mathrm{Ed}}$) are computed based on the rotations in the set (steps 8 and 9). Unused ERd- and Ed-wagons are distributed to the sets (step 13), prioritizing sets containing IC-compositions. The subproblems are solved in parallel (step 16), each with a time limit q_i and the corresponding upper limits. Finally, the solutions x_j of the subproblems are combined to form the new current solution x^* (step 17). When applying the Algorithm to exception weeks steps 7-15 are skipped, as the Ed- and ERd-wagons are fixed to compositions already.

The partitioning of the rotations into the subproblems can be performed randomly (referred to *random route decomposition* in [8]) or based on specific characteristics of the rotations, such as the average revenue per passenger, average demand or maximum

train distance. Furthermore, each iteration may use different parameter settings, including the time limit, termination MIP-gap value and varying number of sets m. These two approaches are defined as *random* and *nonrandom* decompositions.

4.3.4 Large neighborhood search

In addition to the decomposition heuristic, a local search method is deployed to help solve large instances of the RSRP. This method is known as the Large Neighborhood Search (LNS), first introduced by [9], also known as *ruin-and-recreate* method [8], as part of solution is relaxed (destroyed), while the rest is kept fixed. Then the relaxed part of the solution is reoptimized to obtain a better solution. The LNS has previously been applied to for example multidepot vehicle scheduling problem [47].

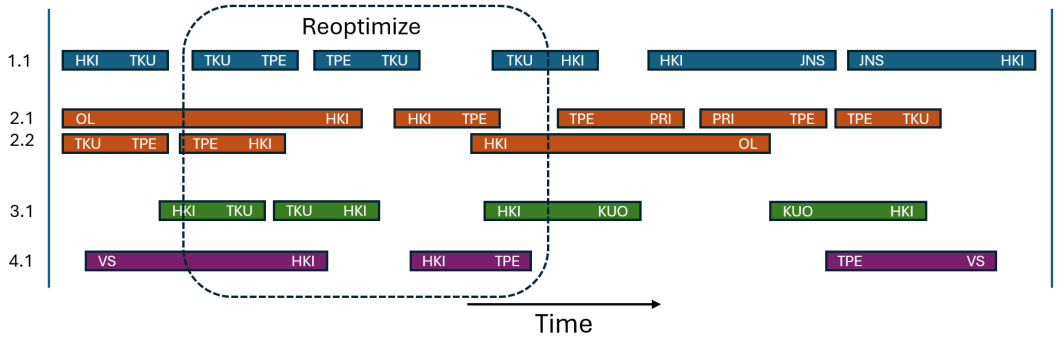


Figure 11: The train and turns included in a iteration of the LNS for an existing RSR plan, drawn as a box around the search area.

In the context of RSRP in this thesis, the LNS is applied by fixing the rotation for a part of the week while the rest of the week is reoptimized. In Figure 11, a box drawn around the duration for which the turns would be relaxed and reoptimized. In practice, all rotations for 2 consecutive days are relaxed and reoptimized. This is repeated for all consecutive pairs of days for the whole week. While the decomposition heuristic allows trains to be swapped from one rotation to any other rotation within the subproblem, LNS allows swapping between all rotations within these 2 days.

Algorithm 2 LNS algorithm for RSRP

Input: Timetable and other data, composition type limits, max iterations I, time limit per iteration q_i

Output: RSR plan

1: Generate or provide an initial feasible solution $x^0 = \{R_1, R_2, ..., R_n\}$ with n rotations for the RSRP
2: Set $x^* \leftarrow x^0$
3: Set iteration counter $i = 0$
4: **while** $i < I$ **do**
5: Fix the turns of 5 consecutive days of the week
6: Provide the turns for the remaining 2 days as warm start values
7: Solve the semi-fixed RSRP with a time limit q_i to obtain new solution x^i
8: Set $x^* \leftarrow x^i$
9: i = i + 1
10: **end while**
11: **return** x^*

The LNS algorithm for RSRP is summarized in Algorithm 2. An IFS is x^0 is generated or provided (step 1) and set as the current solution x^* (step 2). During each iteration, the turns of 5 consecutive days of the week are fixed (step 5) and the remaining two days are relaxed and provided as warm start values (step 6). The semi-fixed RSRP is then solved with a time limit of q_i (step 7), producing a new solution x^i, which is set as the current solution x^* (step 8). Seven iterations are required to fix all consecutive pairs of days for the standard week model and six iterations for the exception week model.

4.4 Test sets and other data

The functionality and applicability of the models developed in this thesis are evaluated using real timetables for long-distance trains in Finland provided by VR. Both the standard week and exception week model are evaluated with 3 different timetables: S, M and L. Each timetable spans over one week.

Test set	Trains	Stations	Trains affected
S	177	17	-
S-exc	163	17	16
M	666	10	-
M-exc	640	10	26
L	1051	18	-
L-exc	1044	18	66

Table 7: Overview of the test sets used for validating the models. Exception week test sets are denoted by the suffix "-exc". For each test set, the number of trains and turns included is reported. Column "Trains affected" refers to the number of trains affected in the corresponding standard week test set during the exception week.

The timetable test sets are described in Table 7. For the standard week, the smallest S test set has 177 trains, the medium sized M test set has 666 trains and the full timetable for a week 1051 trains. The trains in the S and L test sets span over a greater geographical area than the trains included in the M test set. This is evident from the number of stations. The M test set represents a situation in which the compositions are planned to operate between a subset of stations only. This can improve the robustness of the RSR plans, as in the case of a possible disruption, only a subset of rotations would be affected.

Exception week plans capture deviations from the standard week plan due to, for example, holidays or track work. In the S-exc test set, 16 trains are affected (modified or removed) in the timetable, in the M-exc test set, 26 trains are affected and in the largest test set L-exc 66 trains are affected. For the L test set, the number of trains to operate in the exception week compared to the standard week only differs by 7 trains, but in total 66 trains are affected due to deviations in the departure and arrival stations.

This optimization framework assumes the provided timetables are balanced, i.e. each station has an equal number of trains arriving and departing over the week. Without a balanced timetable, compositions would accumulate at certain stations when the rotations are repeated. This imbalance could be resolved with deadheading. However, the models developed in this thesis require deadheading movements to be predefined i.e. be included in the timetable. With exception weeks, the source and destination trains contribute to maintaining this balance.

In addition to the timetable, the optimization models require other data. This is the data of the orientation changes during trains and turns, limitations and information of the composition types, operational costs, passenger demand and average revenue estimates per passenger. Passenger demand is estimated for each train in the timetable. In addition, turn penalty values for all turns are required.

4.4.1 Generation of turns

A selective generation of turns is used, as not all turns will be feasible to use in practice. As it is assumed that deadheading is not allowed, all turns must take place at the same station. Furthermore, it is assumed that each turn has a maximum duration of x hours. That is the duration between the arrival of the first train and the departure of the next train of the turn. This avoids creating unnecessary turns, which would make the solution search space of the model larger. The generation of turns can be further tuned to exclude turns which are not feasible, for example, due to the track layout at a station or other operational reasons.

As previously discussed, the timetable for standard weeks must be balanced. That is, each station has as many trains arriving as departing during the week. The cyclic nature of the standard weeks allows to calculate a station balance of compositions at each time point. At the time points when a train arrives to the station, the number of compositions at the station increases by one. Similarly, when a train departs from the station, the number of compositions decreases by one. It is assumed that this is the only way compositions may be at stations. For many stations during the week, there are time points when there are zero compositions at the station. These time points can

be used to eliminate turns. For example, if the station balance is zero from 10:00 to 14:00 at a station on a Monday, all of the compositions that have arrived at the station prior to 10:00 have turned to another train before 10:00. Thus no turn can take place over 10:00 to 14:00, as all of the compositions were used already. This elimination assumes the minimum number of possible compositions is used and no additional compositions are available at the stations.

4.5 Implementation

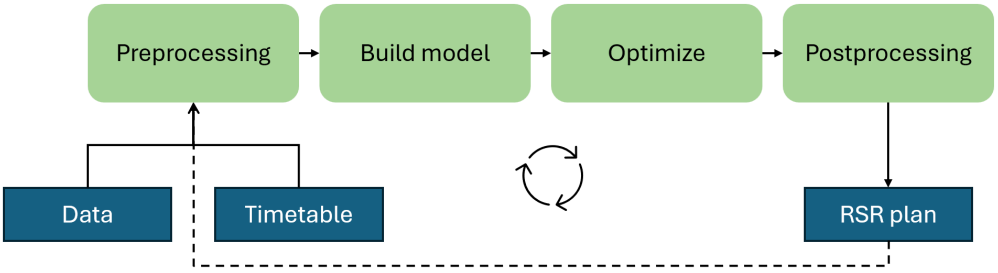


Figure 12: Visualization of the implementation framework.

The implementation of the optimization framework described in this thesis was done in Python, utilizing libraries such as Pandas and NumPy for pre- and postprocessing of the data. The implementation is composed of four parts: preprocessing, model building, optimization and postprocessing, as visualized in Figure 12. The resulting RSR plan can be provided as an input (warm start for the solver) for the next iteration of the optimization. In total, over 10 000 lines of code were written with accordance to best practices to ensure it would be applicable for practical application.

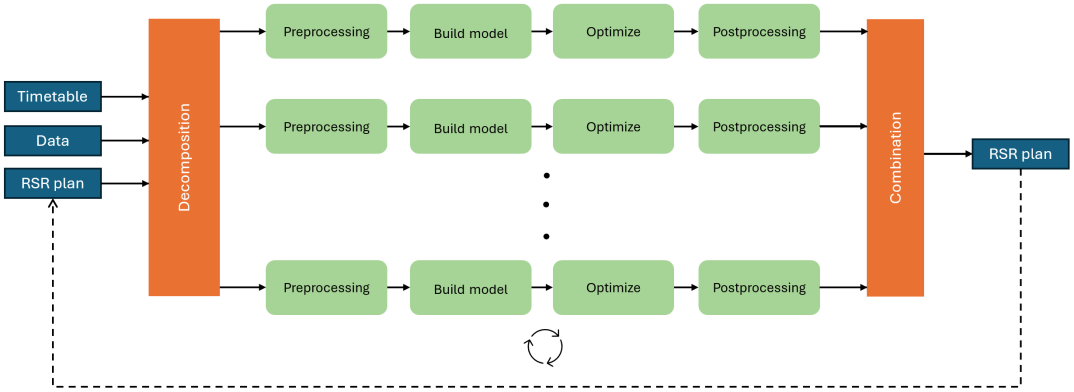


Figure 13: Framework visualization for the implementation of the decomposition algorithm.

In Figure 13, the code implementation framework for the decomposition algorithm (Algorithm 1) is presented. The existing RSR plan along with the timetable and other data is provided as input. Based on the RSR plan, the rotations are decomposed into multiple subproblems. The processing of each subproblem (preprocessing, model building, optimization and postprocessing) is done in parallel. After all subproblems have been solved and processed, the resulting RSR plans of the subproblems are combined to form a new RSR plan, which is then used as the input for the next iteration.

5 Computational results

The performance of the RSRP optimization models and the four solution methods developed in this thesis are evaluated using six real-world timetable test sets. Each test set is subject to a two-hour time limit for optimization. This section is structured as follows. The model sizes are first summarized, followed by the initial feasible solution (IFS) generation results for each test set. Next, the performance of the solution methods for standard and exception weeks are compared. Finally, the results are interpreted.

For each dataset, all composition types are available; however, the number of IC and Pendolino composition units varies according to the size of the test set. Similarly, the number of ERd- and Ed-wagons is limited and scaled according to the test set size. As discussed in Section 2, the RSRP instances considered in this thesis are capacity-limited and thus the maximum number of available composition units of each type are assumed to be known. All models are solved using Gurobi [48] version 13.0.1 on Amazon AWS with a virtual c7i.8xlarge machine (16 CPU cores, 64 GB of RAM).

5.1 Model sizes

Test set	Trains	Turns	Turns after balancing	Reduction
S	177	1992	740	63 %
M	666	27403	20757	24 %
L	1051	54670	44211	19 %

Table 8: Number of trains and turns in each standard week test set. "Turns after balancing" denotes the number of turns remaining after eliminating infeasible turns due to the balance of the timetable as discussed in Section 4.4.1.

In Table 8, the original number of potential turns as well as the number of remaining turns after utilizing the balance of the timetable to remove impossible turns for the standard week models are presented. Exploiting the balance of the timetable leads to a large reduction in number of turns across all test sets. The reduction is largest in the smallest test set S, where the number of turns decreases by 63 %, compared to 24 % for the M test set and 19 % for the largest test set L. As the smallest test set is comparable in size to the subproblems solved in decompositions of the largest test set, this reduction property decreases the number of turns considered in each subproblems.

Test set	Trains	Full model			IFS model			Reduction %	
		Turns	Var	Constr	Turns	Var	Constr	Var	Constr
S	177	740	10495	6986	738	2219	1273	79	82
M	666	20757	201021	62717	20321	60968	22323	70	64
L	1051	44211	420359	121014	43655	130970	46812	69	61

Table 9: Model sizes for each standard week test set. "Turns" denote the potential turns, "Var" refers to the number of variables and "Constr" to the number of constraints in the model. The IFS model contains fewer turns due to removing a subset of turns to ensure orientation compatibility. The reduction percentages are calculated as the relative decrease from the full model to the IFS model.

In Table 9, the model sizes of each test set for the standard week are presented. The final columns report the percentage reduction in the number of variables and constraints from the full model to the IFS model. The reduction in the number of variables varies between 69 and 79 %, with a larger reduction observed for smaller test sets. A similar pattern is observed with constraints, where reductions vary between 61 and 82 %. These results indicate that the IFS model is significantly smaller than the full model, which is expected to decrease complexity and solution time.

Test set	Trains	Full model			IFS			Reduction %	
		Turns	Var	Constr	Turns	Var	Constr	Var	Constr
S (exc)	163	1558	21172	8869	1556	1708	344	91	96
M (exc)	640	21161	581576	99333	20262	21851	1326	96	99
L (exc)	1044	47787	2742951	277513	46318	49901	2188	98	99

Table 10: Model sizes for each exception week test set, similar to Table 9.

In Table 10, the size of each model for exception week are presented. As with the standard week models, the IFS models are smaller than the full models, with a 96-99 % reduction in constraints and 91-98 % reduction in variables relative to the full model. In contrast to the reductions observed for the standard week models, the reductions are greater for the larger test sets. Overall, the omitted constraints contribute substantially to the model size.

The comparison of IFS and full models show that substantial reductions occur across both models and test sets when forming the IFS. The formed IFSs are expected to be easier to solve than the full model. These solved IFSs are provided as warm starts for the solver.

5.2 Initial feasible solutions

Initial feasible solutions (IFS) are generated for each dataset to be used in the optimization process by the solution methods 2, 3 and 4 (Figure 9). The same IFS is used for these three methods of a given test set to ensure accurate performance comparison between the methods.

To generate IFSs which enable the use of the decomposition method, the maximum length of rotations must be limited. Without this limitation, the IFS may not contain a sufficient number of rotations to allow decomposition into groups. However, restricting the maximum rotation length affects the computational time required to generate the IFS. Preliminary testing indicated that setting the maximum rotation length to the desired value of 3 weeks for test sets M and L significantly increased the time to find an IFS. Therefore, the maximum rotation length is set to 2 weeks for the S test set, 6 weeks for the M test set and 10 weeks for the L test set. These limits were determined through iterative testing. This approach allowed to get a sufficient number of rotations to perform the decomposition method, while avoiding generating an excessive number of lazy constraints during the IFS generation, which slowed down the solving process.

Optimization parameter	Value
MIPFocus	1
Heuristics	0.5
MIPGap	0.01
PreCrush	1
CliqueCuts	2
NoRelHeurTime	30/50/600
Threads	-/-/16

Table 11: Non-default optimization parameters for generating the IFS with Gurobi.

To accelerate the solving process, a couple of the solver's optimization parameters are adjusted. Gurobi's "no-relaxation-heuristic" (NoRelHeur) parameter is utilized when solving the IFS models. For the S test sets, the no-relaxation-heuristic time is set to 30, 50 and 300 seconds for the S, M and L test sets, respectively. Other non-default optimization parameters are set to the values in Table 11. The parameter "MIPFocus" is set to 1 to prioritize feasible solutions over proving optimality. Heuristics is set to 0.5, as through preliminary testing it was discovered that most solutions were found by heuristics in the Branch-and-Bound stage. MIPGap is set to 0.01, since that tolerance is enough in the IFS generation stage. PreCrush is set to 1, as suggested by Gurobi when using lazy constraints [48]. CliqueCuts are set to 2, as the problem is similar to set partitioning problems and clique cuts can be beneficial for those [48].

Preliminary experimentation indicated that finding an optimal IFS for the L test set is faster when using a smaller machine, likely because fewer threads can sometimes lead to better performance, as reported in [49]. Therefore, the number of threads is limited to 16 (from the original 32) during the IFS solution generation for the L test set. Otherwise, the same machine and optimization parameters are used for all IFS generations.

Test set	Obj. val	Gap (%)	FFS (s)	Run time (s)	LC	μ(R)	max R
S	3400.3	0.01	<5	9	237	1.29	2
M	7882.0	0.00	36	87	1257	2.30	5
L	10606.7	0.37	51	511	5567	3.85	10
S (exc)	3578.4	0.00	<1	<1	0*	-	-
M (exc)	8683.3	0.00	<1	<1	0*	-	-
L (exc)	13751.4	0.00	<1	<1	0*	-	-

Table 12: IFS generation results for all test sets. Gap refers to the MIPgap reported by the solver, FFS refers to the time to find the first feasible solution, Run time to the time to solve the IFS model to optimality and LC to lazy constraints generated during the solving process. μ(R) denotes the average rotation length in weeks and max R the maximum rotation length in weeks. Lazy constraints were not reported in the logging provided by the solver for exception weeks (denoted by *).

The results of generating the IFS for each test set are presented in Table 12. An optimal solution (using a 1 % MIPgap) is found for each test set. For the standard week test sets, all solutions are obtained within 10 minutes. For exception weeks, the optimal IFSs are found in less than a second for all test sets. While the average week length μ(R) and maximum rotation lengths are large for the M and L test sets, the number of lazy constraints generated during the solve time remained acceptable and did not slow the solving process significantly.

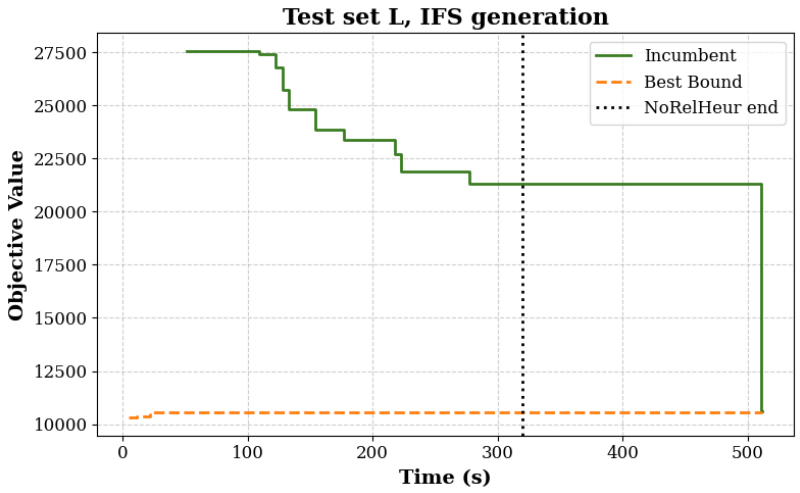


Figure 14: Solving process of the IFS model for test set L. The no-relaxation-heuristic is run for 300 seconds as indicated by vertical dashed line.

In Figure 14, the optimization progress for generating the IFS for the L test set is shown. The first feasible solution is found after 50 seconds during the no-relaxation-heuristic. The incumbent solution improves throughout this phase, which is run for 300 seconds as indicated by a vertical dashed line in the figure. After this, the root relaxation is solved and the Branch-and-Bound stage starts. The incumbent does not

improve before the solver finds the optimal solution after around 500 seconds. As the first found solution could already be used to warm start the full model, the remaining time spent optimizing may not be necessary. Utilization of the first feasible solution could reduce the total optimization time and thus improve computational efficiency. However, it is also possible that a non-optimal IFS would slow the full optimization. The solving process for the other standard week IFS models follows a similar pattern.

5.3 Performance comparison

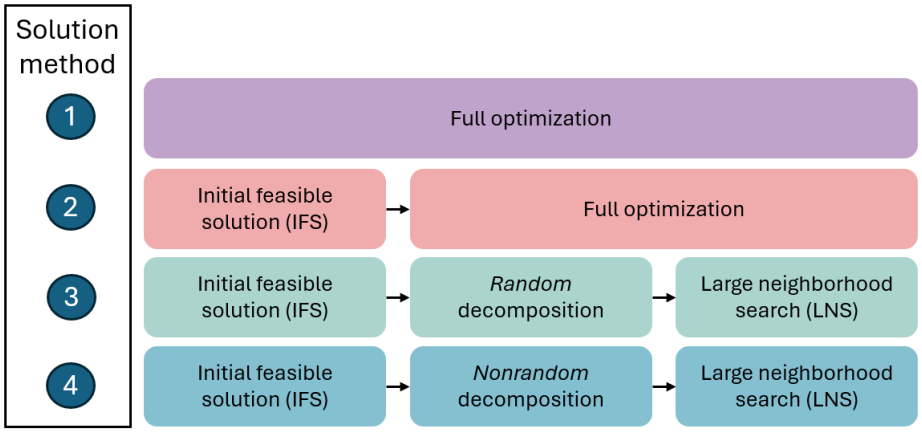


Figure 15: Visualization of the solution methods evaluated in this thesis.

The performance of the solution methods (Figure 15) are evaluated for both models across the six test sets. Method 1 is referred to as "Full" and method 2 as "IFS+Full" in the following tables and figures. The performance of the model and the solution methods for exception weeks are analysed separately from the standard week model. For both models, the S test set is solved using only the first two solution methods, while the M and L test set are used to evaluate all solution methods. In all runs, the LNS stage is run only with the no-relaxation-heuristic, as solving the root relaxation is computationally consuming and not efficient within the given time limit. To compare solution methods, each method is set a maximum runtime of 7200 seconds (two hours).

The *random* decomposition is composed of 12 iterations, with each iteration set to a maximum solving time of 300 seconds, 100 seconds for the no-relaxation-heuristic and a MIP gap of 0.01 %. The total maximum runtime is thus 3600 (one hour) seconds for the *random* decomposition. The decompositions used in the *nonrandom* decomposition method were iteratively preselected by adjusting the MIP gap, no-relaxation-heuristic time, max runtime limit, the number of subproblems formed and the conditions based on which the subproblems are formed on each iteration. The iterations are composed such that the total maximum runtime is equal to 3600 seconds, as with the *random* decomposition method. The *nonrandom* decomposition is composed of 17 iterations. The same *nonrandom* decomposition is used for both M

and L test sets and for both models. Since the decomposition methods are unable to obtain a global bound, the MIP gaps reported in the results are calculated based on the bound obtained by the full optimization with IFS.

In both *random* and *nonrandom* solution methods, the decomposition method is run first and the LNS second. This order was chosen as during preliminary testing the objective value improved significantly faster with the decomposition method than with the LNS method, when both were started with the same IFS. In addition, with the LNS method, the maximum rotation length does not decrease as fast as with the decomposition methods.

5.3.1 Standard week model

The performance of the four solutions methods for the standard week model are evaluated in this section. The solution quality, computational performance and effect of decomposition methods are evaluated. The desired maximum rotation length is set to 3 weeks for the decomposition methods and for the full optimization without IFS. For the full optimization with IFS, the maximum rotation length is determined by the maximum rotation length in the IFS, as setting a tighter limit would make the IFS infeasible for the full model.

Method	IFS value	Obj. val	BB	Gap (%)	Time (s)	LC	μ(R)	max R
Full	-	51930.2	51930.2	0.00	**21.3**	**2**	1.5	2
Full+IFS	425386	51930.2	51930.2	0.00	30.9	52	1.5	2

Table 13: Comparison of solution methods on the standard week model with test set S. BB refers to best bound, Gap to the solver-determined MIP gap, LC to lazy constraints, μ(R) to the mean rotation week length in the solution and max R to the maximum rotation length in weeks. The solution time reported does not include the time to generate the IFSs. Bolded value indicates the best value across the methods.

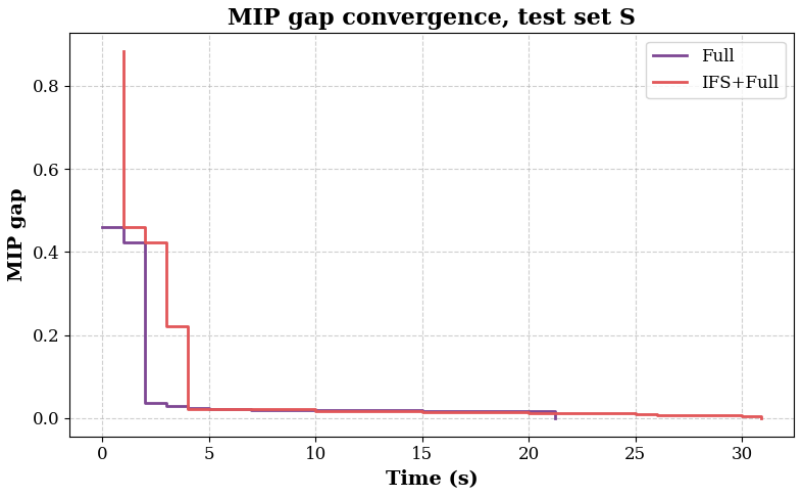


Figure 16: MIP gap convergence when solving the test set S with full optimization without IFS (Full) and full optimization with IFS (IFS+Full). The first value for the IFS+Full is obtained later than for Full, as warm starting the solver with the IFS takes a couple of seconds.

The test set S for standard week is considered first. Two solution methods are evaluated: full optimization and full optimization with IFS (IFS+Full). For both approaches, the maximum rotation length is set to two weeks. The computational results are summarized in Table 13 and the solution process is illustrated in Figure 16. Both methods reached optimality in less than a minute. Given the small problem size and reaching optimality rapidly, the more advanced decomposition methods with LNS are not required. Inspection of the resulting rotations shows that all the required operational constraints are satisfied, indicating that the standard week model works as intended.

The full optimization method without an IFS took 21.3 seconds to reach proven optimality for test set S. During the optimization, only two lazy constraints are generated to eliminate rotations violating the maximum rotation length or Helsinki station visit constraint. In contrast, when the IFS is provided as a warm start, optimality is reached in 30.9 seconds and 52 lazy constraints are added. Including the time to generate the IFS, the total run time is 39.9 seconds. Therefore, for this test set, the direct full optimization without IFS outperforms the full optimization with IFS. Despite the differences in runtime and number of cuts generated, the methods produced identical solutions with exactly the same composition types assigned and selected turns.

These results also provide guidance for determining the size of the subproblems used in the decomposition methods with test sets M and L. Specifically, the S set shows that a set of around 200 trains could be solved to optimality in under a minute.

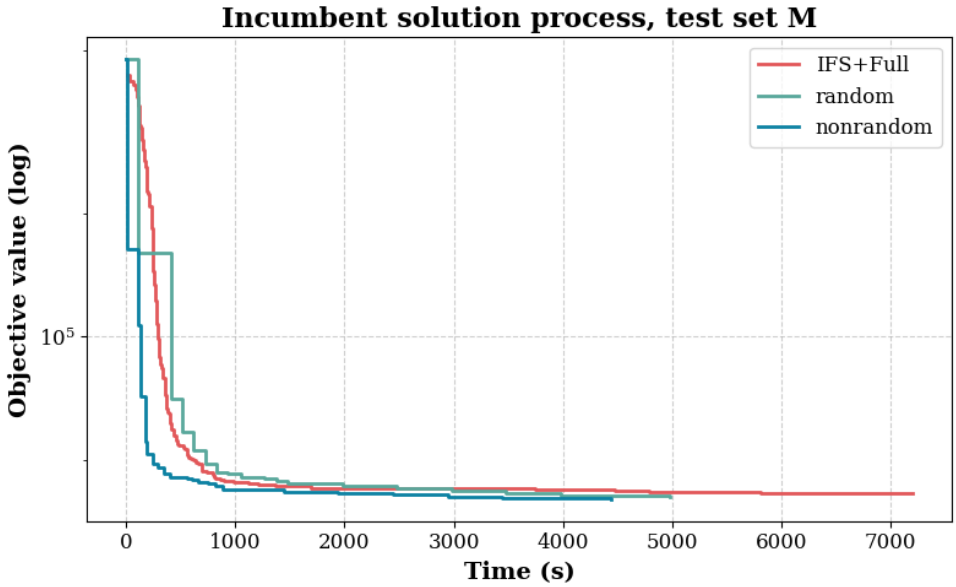


Figure 17: Comparison of performance of the three different solution methods for standard week test set M. The stalling of the solving process for *random* method around 300-400 seconds is due to waiting for a subproblem to reach optimality during an iteration.

Method	IFS value	Obj. val	BB	Gap (%)	Time (s)	LC	μ(R)	max R
Full	-	-	38052	-	7200	0	-	-
IFS+Full	474068	41551	**38065**	8.39	7200	477	**1.53**	5
random	474068	40783	(38065)	6.67*	4978	-	1.64	**3**
nonrandom	474068	**40173**	(38065)	**5.25***	**4383**	-	1.77	**3**

Table 14: Comparison of solution methods for the M standard week test set. The best value obtained across the solution methods is bolded. LC is not calculated for decomposition methods. See Table 13 for description of columns. When denoted by (*), the gaps are calculated from the best bound found with the IFS+Full solution method.

In contrast to test set S, the more complex test set M is used to evaluate the model with increased problem size and complexity. The computational results of solving test set M using the different methods are summarized in Table 14 and the solution process in Figure 17. The full optimization method without an IFS does not produce a feasible solution within the 7200 second time limit. When an IFS is provided, the solver produces a solution with a 8.39 % gap within the time limit.

The decomposition methods produce similar solutions, with the best solution achieved with the *nonrandom* decomposition method in 4383 seconds with a gap of 5.25 %. However, the *random* decomposition method produces a similar quality solution. Even though it had fewer iterations, it took slightly longer (4978 seconds). The full time limit of 7200 seconds for either decomposition method is not utilized, as

each iteration of the decomposition is terminated once all subproblems are solved. The convergence behaviour illustrated in Figure 17 indicates all three solution methods converge similarly over time for test set M.

With respect to the the RSR plan structure, the full optimization results in the smallest mean rotation length. However, because the IFS sets the maximum rotation length, the best solution obtained by the full optimization with IFS also includes a rotation with 5 weeks, exceeding the goal of 3 weeks or less. In contrast, both decomposition methods produced solutions where the maximum rotation length satisfies the 3 week limit.

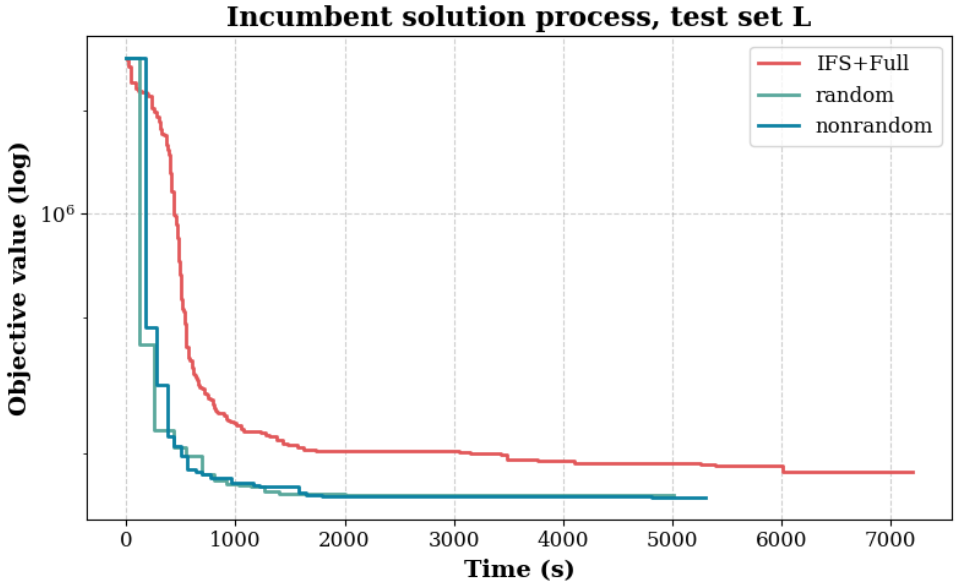


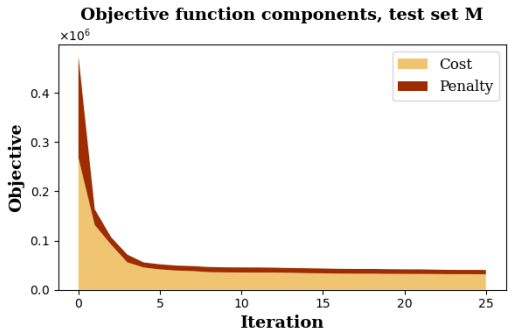
Figure 18: Comparison of performance of three different solution methods for standard week test set L.

Method	IFS value	Obj. val	BB	Gap (%)	Time (s)	LC	μ(R)	max R
Full	-	-	134774	-	7200	-	-	-
Full+IFS	2846420	176377	**134792**	23.58	7200	1424	2.94	8
random	2846420	150953	(134792)	10.71*	**5115**	-	2.00	**3**
nonrandom	2846420	**148993**	(134792)	**9.53***	5348	-	**1.92**	**3**

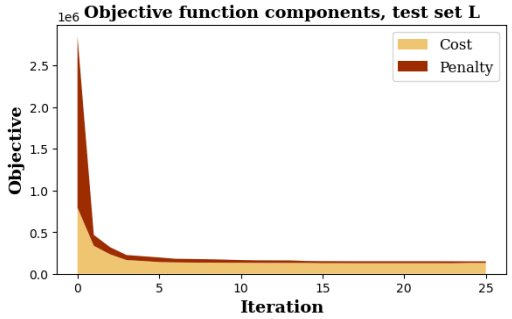
Table 15: Comparison of solution methods for the L standard week test set, given the time limit of 7200 seconds. The best value obtained across the solution methods is bolded. See Table 13 for description of columns. When denoted by (*), the gaps are calculated from the best bound found with the IFS+Full solution method.

The computational results for the different solution methods for test set L are summarized in Table 15 and the solution process in Figure 18. Test set L is the largest and representative of a real-life problem size. As with test set M, the full optimization without an IFS fails to produce a feasible solution for test set L in the

given time limit. Comparing the other three methods, the best objective value is obtained with the *nonrandom* decomposition method. Both decomposition methods clearly outperform the full optimization with IFS when comparing the objective value, computation time and the mean and maximum rotation lengths. Both decomposition methods return a solution with the maximum rotation length below the given threshold of 3 weeks, while the full optimization with IFS method produces a solution with an 8-week rotation. Inspecting the solution process in Figure 18, the decomposition methods converge similarly, outperforming the full optimization method with IFS. The *nonrandom* method produces a slightly better solution that *random* method. Therefore, the solution process of *nonrandom* is further inspected.



(a) Objective components during each *nonrandom* decomposition iteration, test set M. At the end, penalties account for 21 % of the total objective.



(b) Objective components during each *nonrandom* decomposition iteration, test set L. At the end penalties account for 12 % of the total objective.

Figure 19: Share of cost and penalty in each solution during the *nonrandom* decomposition method for test sets M and L.

The objective function consists of two parts: costs and penalties. Penalties are from the restaurant-wagon allocation constraint, exceeding maintenance limits and turn time penalties. Except for the turn time penalty, the other penalties are included to the model to ensure the IFS is feasible w.r.t. full optimization model. The full optimization model minimizes the sum of costs and penalties. Therefore, it is of interest to analyse the proportion of each in the objective function throughout the solving process of the best performing method. The share of costs and penalties in the objective function during each iteration of the *nonrandom* decomposition method are illustrated in Figure 19a (test set M) and Figure 19b (test set L). In the beginning of the optimization process (i.e., in the IFS solution), the costs and penalties account for an approximately equal amount to the objective function of test set M. For test set L, the share of penalty component is initially larger than the cost component. For both test sets, most of the penalty is eliminated during the first 5 iterations of the decomposition method. At the end of the solution process, the share of penalty in the objective is 21 % and 12 %, for the test sets M and L, respectively.

For test set M, the minimum cost obtained during the optimization process was 31594, which corresponds to the value obtained in the last iteration. For the L test set,

the minimum value obtained during the iterations for the costs was 126589, whereas the cost in the best objective value was 130541. This indicates the penalties resulting from not having restaurant wagons for trains with a distance over $L_{R,km}$, composition going over the maintenance limit and turn duration penalties are a trade-off from achieving a more efficient RSR plan in terms of capacity utilization and demand matching.

Method	IFS	After decomp.	Time (s)	After LNS	Total time (s)
random	474068	43922.4	1478.3	40782.9	4978.3
nonrandom	474068	**42270.4**	**943.6**	**40173.3**	**4443.6**

Table 16: Comparison of performance of *random* and *nonrandom* decomposition methods for the test set M. In column IFS, the value of the IFS solution evaluated by the full model is reported. "After decomp." refers to the objective value obtained after the decomposition method and "After LNS" to the objective value obtained after the LNS stage.

Method	IFS	After decomp.	Time (s)	After LNS	Total time (s)
random	2846420	151856.7	**1518**	150953.1	**5018**
nonrandom	2846420	**150329.8**	1802	**148993.2**	5302

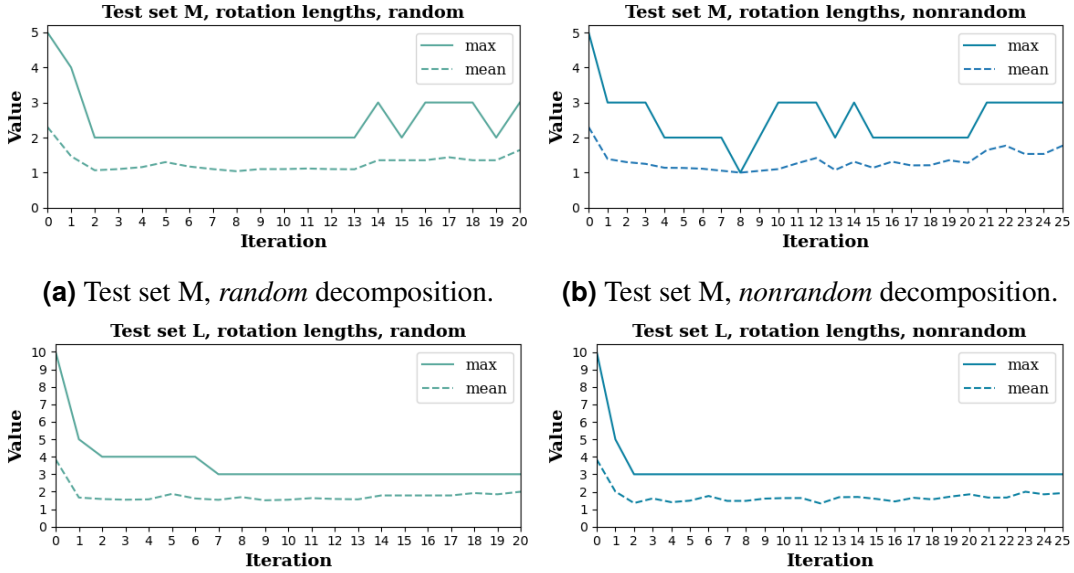
Table 17: Comparison of performance of *random* and *nonrandom* decomposition methods for the test set L, similarly to Table 16.

A comparison of the *random* and *nonrandom* decomposition methods for datasets M and L is presented in Tables 16 and 17. For both test sets, the *nonrandom* method produces a better objective value. However, while having the same maximum solving time limit, the *nonrandom* method has more iterations than the *random* decomposition. The subproblems are solved in the given time limit to optimality and thus the time limit could have been decreased and more iterations included in the *random* decomposition methods. The difference in the number of decomposition might bias the *nonrandom* decomposition to perform better. Nevertheless, with the both test sets, the LNS is able to improve the solutions. However, the proportion of time spent on the LNS compared to the decomposition is large in comparison to the improvement in the objective function value. This may indicate that the LNS is less efficient in improving the solution than decomposition methods or that the objective value is already close to optimality and thus larger improvements are impossible.

	random (60 subproblems)		*nonrandom* (89 subproblems)	
Test set	Optimal (%)	Not optimal (%)	Optimal (%)	Not optimal (%)
M	98	2	99	1
L	100	0	93	7

Table 18: Share of optimal and not optimal solutions to subproblems found during the *random* and *nonrandom* decomposition methods for test sets M and L.

To further compare the two decomposition methods, the number of subproblems solved to optimality for each test set with each decomposition method are presented in Table 18. Most of the subproblems were solved to optimality (w.r.t. the given MIP gap of the iteration) during each iteration with both decomposition methods. When the subproblems are solved in parallel, the duration of an iteration is determined by the slowest subproblem to solve to optimality. If all subproblems are solved to optimality quickly, the algorithm moves onto the next iteration faster. This stands out in Figure 17 with the *random* decomposition solution, where around 100 to 400 seconds, the incumbent solution does not improve. This is likely due to one subproblem taking the whole allocated 300 seconds to terminate. This clearly sets the *random* decomposition solution method at a disadvantage in the performance comparison of methods with test set M.



(a) Test set M, *random* decomposition.

(b) Test set M, *nonrandom* decomposition.

(c) Test set L, *random* decomposition.

(d) Test set L, *nonrandom* decomposition.

Figure 20: Maximum and mean rotation length (in weeks) after each iteration of the decomposition methods for test sets M and L.

In Figure 20, the maximum rotation length in each iteration of the decomposition methods for test sets M and L are presented. For all decomposition methods, the desired maximum rotation length of 3 weeks is reached. Only test set L with *random* decomposition requires more than two iterations of the decomposition heuristic to reach the desired value of 3. For test set M with the *nonrandom* decompositions in Figure 20b, the maximum rotation length decreases to one week during iteration 8, which is the minimum. However, during subsequent iterations, the maximum rotation length increases to the maximum 3. This behaviour suggests that either an optimal solution includes rotations longer than one week or that the solver is able to more easily find solutions with longer rotations. For all test sets, the mean rotation length increases during the last iterations. The last 7 iterations were from the LNS method, suggesting that while the LNS method improves the objective, it also increases the

average rotation length.

Overall, these results demonstrate that the standard week model for RSRP is solvable and high-quality solutions are found within a reasonable time for all test sets. For smaller test set (or decomposition subproblems), optimality can be proven, whereas for larger datasets, solutions of sufficient quality can be obtained within the two hour time limit.

5.3.2 Exception week model

The performance of the solution methods is evaluated for exception weeks following the same approach as with the standard week test sets. The test sets S-exc, M-exc and L-exc introduced in Section 4.4 are used for the evaluation and hereafter referred to as the S, M and L test sets.

Method	IFS value	Obj. val	BB	Gap (%)	Runtime (s)
Full	-	52683.4	52683.4	0.0000	52
IFS+Full	123560	52683.4	52683.4	0.0000	47

Table 19: Comparison of solution methods on the test set S, exception week. BB refers to the best bound and Gap to the MIP gap determined by the solver.

In Table 19 the full optimization and full optimization with IFS are compared for the test set S for exception weeks. An optimal solution is obtained with both methods within a minute. The full optimization with IFS is slightly quicker, even when considering the time spent generating the IFS, which took under a second. Given the problem size and rapid convergence, the decomposition methods are thus not required for test set S.

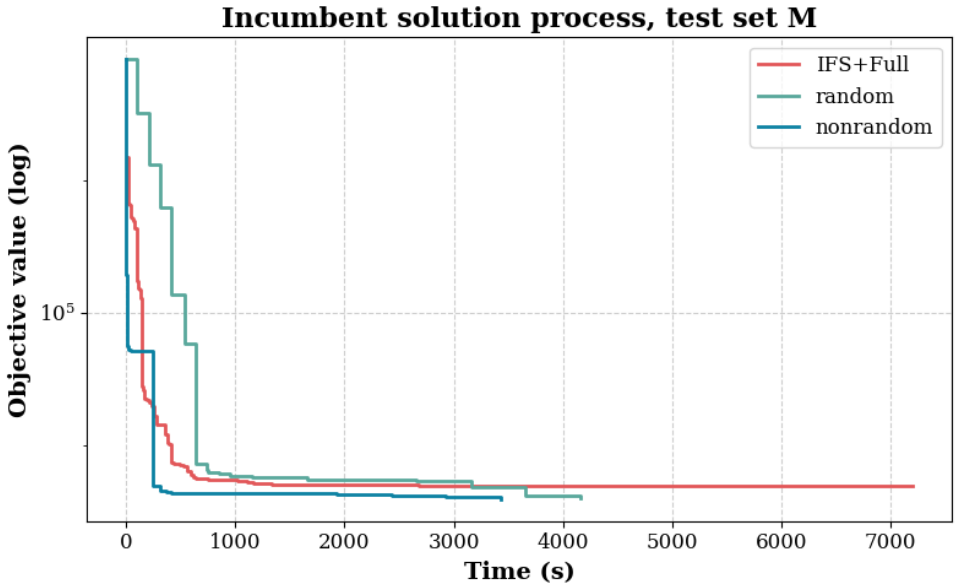


Figure 21: Solving process of the three methods for test set M (exception week). The IFS+Full method is able to improve the IFS before starting the main solving process, therefore the first point is lower compared to the other methods. This is due to the turns to the subsequent week selected by the IFS not being included in the list of warm start variable values.

Method	IFS value	Obj. val	Best bound	Gap (%)	Time (s)
Full	-	-	36601	-	7200
Full+IFS	228875	40363	**36618**	9.28	7200
random	228875	37912	(36618)	3.41*	4155
nonrandom	228875	**37724**	(36618)	**2.93***	**3416**

Table 20: Comparison of solution methods for the M exception week test set. When denoted by (*), the gaps are calculated from the best bound found with the IFS+Full solution method.

The performance of the solution methods for the test set M are presented in Table 20. The *nonrandom* decomposition method achieved the best objective value in the shortest time. Figure 21 presents the solving process for the different methods for test set M. The *nonrandom* decomposition visually performs the best, however, the difference is not significant.

Method	IFS	After decomp.	Time (s)	After LNS	Total time (s)
random	228875	42213	1155	37912	4155
**random*	228875	40743	**46**	-	-
nonrandom	228875	**39008**	416	37724	**3416**

Table 21: Comparison of decomposition method performance for test set M. Decomposition algorithm was run without the no-relaxation-heuristic in method **random*. The LNS step was not run for **random* method.

The decomposition methods run for test set M are compared in Table 21. All subproblems in the decompositions are solved to optimality. However, when inspecting the solver's logging from the decompositions subproblems, it seems that the progress stalls during the no-relaxation-heuristic. In contrast, the root relaxation and the Branch-and-Bound stage are solved to optimality quickly in each subproblem. Therefore, as an experimental adjustment, the *random* decomposition is rerun without the no-relaxation-heuristic. When running *random* decomposition without the no-relaxation-heuristic (**random*), the total runtime decreases by around 90 % across the 12 iterations. Although the **random* clearly performs the best in terms of time, the objective value is similar to those of *random* and *nonrandom*. Therefore, LNS is not run for **random*. For both *random* and *nonrandom* decomposition methods, the LNS is able to improve the solutions, resulting in comparable objective values.

Next, the solution methods on the test set L are evaluated. Unlike for the other test sets, the IFS solution obtained for the test set L is not feasible w.r.t. the orientation constraint. This is due to a failure in creating a comprehensible list of unallowed turns to artificially ban incorrect orientations in the IFS generation process. Therefore, in the full optimization and decomposition methods, orientation constraints are implemented via a penalty function as described in the Appendix B to make the IFS feasible for the full optimization. This increased the size of the model by an additional 433 binary variables. This scenario reflects a real-life operational context, as it may not always be possible nor time efficient for planners to create the list of turns to ban for the IFS to ensure fulfilment of the orientation constraint.

Method	IFS value	Obj. val	Best bound	Gap (%)	Time (s)
Full	-	-	-1.8e+07	-	7200
Full+IFS	908271	501075	-1.9e+07	3909	7200
random	908271	**127236**	-	-	**5436**
nonrandom	908271	127472	-	-	6050

Table 22: Comparison of solution methods for the L exception week test set.

In Table 22, the solution methods' performance on the test set L are summarized and the solving process visualized in Figure 22. No feasible solution is obtained with the full optimization and both decomposition methods clearly outperform the full optimization with IFS. In the full optimization with IFS, all improvement in the objective function occurs during the no-relaxation-heuristic, while the remainder of the optimization is spent solving the root relaxation, specifically the barrier method.

The Branch-and-Bound stage is not reached. As a result, the best bounds obtained are not meaningful and therefore no MIP gaps are calculated for the decomposition methods. Nevertheless, when comparing the best obtained objective value (127472) to the best objective value obtained for the standard week test set L (148993), the exception week solution is better. Since both models optimize the same key costs, and the exception week test set is majorly the same as the standard week test set, the objective values are comparable. Therefore, it can be inferred that the best objective value obtained for L test set in exception weeks is near-optimal, despite the absence of a meaningful best global bound.

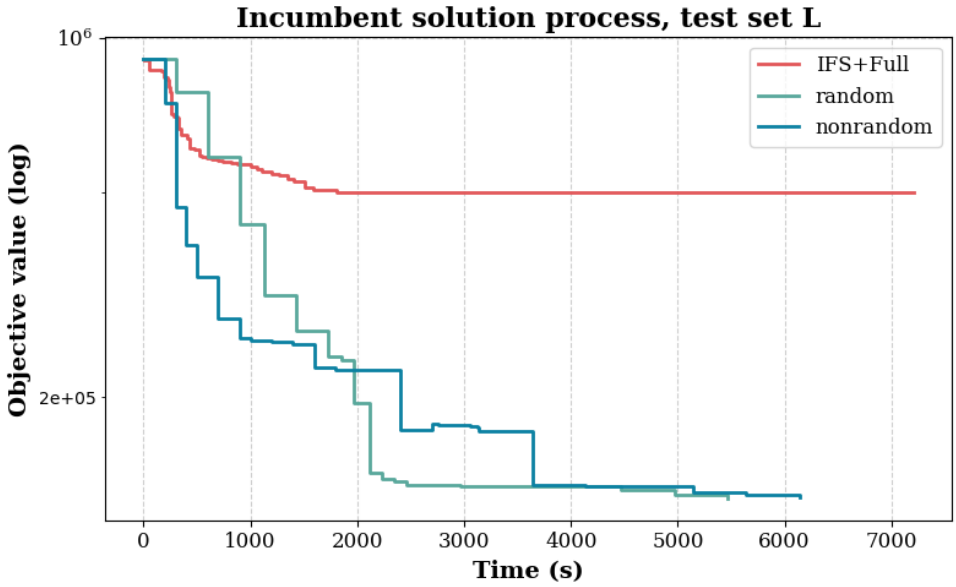


Figure 22: Solving process of the three methods for test set L (exception week).

Method	IFS	After decomp.	Time (s)	After LNS	Total time (s)
random	908271	**134706**	**2436**	**127236**	**5436**
**random*	908271	201004	3648	-	-
nonrandom	908271	172075	3050	127472	6050

Table 23: Comparison of decomposition methods for test set L. For method **random*, only the decomposition method without no-relaxation-heuristic was run.

Further comparison of the decomposition methods on test set L are presented in Table 23. The same post-hoc experiment is performed for test set L as for test set M, by running the *random* decomposition without the no-relaxation-heuristic (**random*). In contrast to test set M, this adjustment does not bring clear improvement in solution time, and in fact it performs worse than either other decomposition method. Overall, the *random* decomposition performs the best, with a clearly better objective value reached at the end of the decomposition algorithm. However, as Figure 22 illustrates, the *nonrandom* decomposition method initially outperforms the *random*

decomposition. In addition, Figure 22 illustrates the poor performance of the full optimization with IFS method for test set L.

These results indicate that the RSRP for exception week is solvable for different test sets sizes representing real-life use cases. The best performance is achieved with full optimization for the S test set, whereas decomposition methods perform better with the larger test sets M and L.

5.4 Interpretation of results

Careful examination of the resulting RSR plans indicates that the model formulations are valid and all constraints function as intended. Overall, all test sets for both models are solvable using one or more of the solution methods. High-quality solutions are obtained within the given time limit, demonstrating the practical applicability for real-life use cases.

The initial feasible solution (IFS) generation provides significant performance gains. For the S test sets, full optimization without IFS is sufficient for both standard and exception weeks. However, for larger test sets M and L resembling real-world use cases, no feasible solution is found for either model without warm starting the solver. This highlights the importance of generating an IFS for the RSRP. The generation of IFS is sufficiently fast in all cases, particularly with exception weeks, where the IFS for all test sets are obtained in under a second.

Optimization with orientation constraints formulated with penalty functions is feasible, however, it appears to slow down the solving process. When comparing the M and L test sets of exception weeks, the observed performance difference is larger than expected. This may be attributed to the addition of the orientation constraint as a penalty or due to the exception week problem not scaling well. Nevertheless, even with a slower performance, high-quality solutions were found when comparing to the objective value of the corresponding model week solution. This emphasizes the importance of preprocessing the potential turns when generating the IFS to ensure the IFS is feasible w.r.t. the orientation constraints.

Other observations include that defining and calibrating appropriate penalty weight values is crucial, as there is a trade-off between costs and penalized values. The penalty values should be carefully calibrated to reflect the monetary value from violations. In addition, the decomposition Algorithm 1 performs efficiently and when combined with the cut separation method is able to decrease the overall maximum rotation length to the required value for standard weeks.

The *nonrandom* decomposition method was tested experimentally, however, across all comparisons of *random* and *nonrandom* decomposition methods on all test sets and both models, no consistent performance advantage is observed. For example with the exception week test set L, the *random* decomposition clearly outperforms the *nonrandom* decomposition method before the LNS stage. Nevertheless, further refinement of the selected decompositions could lead to potential performance gains. The refinement requires careful analysis on each iteration of the current RSRP solution to determine the optimal decomposition.

Interestingly, for the M test set for exception weeks, the post-hoc experiment of running the *random* decomposition without the no-relaxation-heuristic proved to be exceptionally fast, producing a high-quality solution in just a couple of minutes. This illustrates the impact of the problem formulation, highlighting the difference between the back-in-time turn structure and the more traditional source-destination MCFP formulation and how these differences influence the solution efficiency.

6 Conclusions

In this thesis, two MILP models were developed to optimize the rolling stock rotations for VR's long-distance trains in Finland. The models address two types of rolling stock rotation (RSR) plans: standard week plans and exception week plans. Both formulations minimize operational costs, lost revenue caused by unmet demand and undesirable turn durations while accounting for constraints related to composition availability, orientation of compositions at Helsinki station, distance-based maintenance constraints and allocation of restaurant wagons. In addition, the maximum length of multiweek rotations is constrained in the standard week model. Four alternative solution methods were developed and evaluated using real-life timetables.

The proposed models were successfully implemented and produced feasible high-quality RSR plans that satisfy all considered constraints. The developed solution methods were able to obtain high-quality solutions within reasonable computation times, demonstrating their applicability for real-life use. Based on the computational results, the combination of generating an initial feasible solution (IFS) for warm starting the solver and applying rotation-based decomposition methods is recommended for both the standard week and exception week models due to their computational efficiency.

Optimized, well-functioning RSR plans are important in practice as they help railway operators to align capacity with demand, reduce operational costs and improve service quality for passengers. Furthermore, the use of optimization methods enables planners to work more efficiently. When high-quality solutions for RSR plans can be generated within reasonable computation times, planners are able to test and compare alternative RSR plan scenarios. The results indicate that the developed models and solution methods are adaptable for use in planning at VR, replacing the current optimization algorithm for standard week planning and introducing optimization for the exception week planning, which is currently performed manually [4].

This thesis contributes to the RSR literature by evaluating alternative solution methods for RSRP, formulating orientation constraints and implementing maximum multiweek rotation length constraints via cut separation. In addition, a clear distinction is made between standard week and exception week planning and application of mathematical optimization to each is presented.

6.1 Limitations

There are several limitations to the work in this thesis related to modelling assumptions, practical implementation, and applicability. First, the rolling stock rotation optimization models developed are not integrated with timetabling or crew scheduling, but instead assume a sequential planning order. While railway operations planning is typically hierarchical [1], integration with other planning processes could improve overall efficiency [50].

Second, several practical limitations affect the applicability of the models and solution methods. The standard week optimization model requires a balanced timetable as an input. Obtaining such timetables may require substantial manual effort from

the planner, as modifications to train services affect many stages across the planning process. In addition, deadheading movements must be predefined in these models and included in the timetable, as the models only consider the timetabled train services. Consequently, potential optimal solutions involving additional deadheading may not be identified. This limitation could be addressed by including turns with deadheading in the models. However, deadheading movements affect the orientation and kilometre-based maintenance and thus constraints related to these should be also adjusted accordingly. The selection of a set of allowed turns such that the initial feasible solution (IFS) is feasible w.r.t. the orientation constraints is not covered in this thesis. As observed with the L test set for exception weeks, without obtaining such an IFS, an increase in the computational time is expected.

Third, the models depend strongly on the estimated demand for each train. Demand is assumed to remain the same for each train over the planning horizon of the standard week, which does not capture real-world variability between weeks. This limitation can be mitigated by the use of shorter planning horizons. Alternatively, the exception week optimization could be used to reoptimize rotations for each week based on week-specific demand estimates.

Fourth, while the computational results show that the models and decomposition solution methods are able to produce high-quality solutions in reasonable computation times, more in-depth sensitivity analysis is required. In particular, further analysis is required to determine the appropriate weights for the penalty functions and to assess the impact of different decomposition strategies on solution quality and performance. The weights of the penalty function should be calibrated to reflect the true trade-off between competing constraints in monetary terms.

Finally, on a broader level, the models developed in this thesis are highly tailored for the operating environment of the considered railway operator, VR, as is common in rolling stock rotation planning [5]. In particular, constraints such as limiting the maximum length of multiweek rotations, allocation of restaurant wagons and composition orientation at Helsinki station are highly operator-specific requirements. However, the underlying modelling techniques and methods, such as cut separation and IFS generation can be adapted to other RSR models.

6.2 Future work

Based on the results and identified limitations in this thesis, several suggestions for future work are proposed. These relate to model extensions, further development of solving methods and broader applicability.

The thesis considered the essential constraints required to produce optimized RSR plans for the long-distance trains of VR. However, more detailed operational constraints exist, which were not considered in this thesis. These are, for example, constraints related to cleaning [4], which have been suggested to be considered in the RSR planning. These constraints could be implemented using the techniques presented in this thesis, for example through cut separation [42] or by modelling them in a similar manner to the distance-based maintenance constraints. Furthermore, the maintenance constraints were assumed to be independent of the type of composition, while in

practice maintenance requirements typically depend on the type of composition or wagon. Future work should also consider incorporating other aspects typically addressed in RSR optimization, such as regularity and robustness.

Cut separation was used in this thesis to eliminate solutions with rotations without weekly Helsinki station visits and rotations exceeding the maximum rotation length. Other constraints, such as maintenance constraints, could also be implemented via cut separation, as demonstrated in [42]. Cut separation is particularly useful when constraints are difficult to directly formulate in the MILP model or would lead to a large number of constraints and auxiliary variables, while checking feasibility w.r.t. the constraint is simple and efficient. However, as discovered during model development, using cut separation with some constraints, such as orientation constraints, can also result in generating a large number of cuts, slowing the computational performance. Therefore, future work should investigate which constraints are better handled through cut separation and which should be explicitly included in the model formulation. In addition, the IFS generation process could be further developed to include more constraints via cut separation which may lead to higher-quality IFSs and further performance improvements of the full models.

While the *nonrandom* decomposition method performed similarly to the *random* decomposition method across the test sets, there is potential to further improve the *nonrandom* decomposition approach. Specifically, the decomposition and optimization parameters selected for each iteration could, for example, be determined based on the quality and properties of the solution of the previous iteration. By selecting the optimal decomposition for each iteration, the performance of the method could improve.

The largest gap between the developed models and real-life concerns the allocation of Ed-wagons. In this thesis, it is assumed that an IC-composition is allocated a fixed number of Ed-wagons that remain attached for the entirety of the planning horizon. In practice, however, Ed-wagons can be coupled and decoupled during sufficiently long turns at Helsinki station. Coupling allows to maintain single wagons during a rotation and to adjust the capacity of a composition during a rotation, leading to more accurate matching of capacity and passenger demand. Therefore, future work should explore modelling the coupling and decoupling of Ed-wagons within both standard and exception week planning.

The computational performance of the models in this thesis may be improved through solver-specific techniques such as concurrent MIP solving [51] or by using barrier methods instead of penalty function methods. More generally, similar multicommodity flow formulations and application of the solution methods presented in this thesis could be applied to other railway planning problems, for example crew scheduling, as demonstrated in [52]. In addition, alternative RSRP modelling techniques, such as utilization of time-space networks [25, 13] or hypergraphs [30], could be explored and their performance compared to the formulations and methods used in this thesis.

Finally, the developed models could be expanded for optimizing rotations in cases of disruption, as demonstrated in [38]. Disruptions typically last only a few days, which makes the exception week model applicable. However, disruption management requires solutions in a short time frame and the methods used in this thesis may not be

computationally efficient enough. Nevertheless, IFSs for all of the exception week test sets were obtained in under a second, suggesting that the reduced model could be further developed to optimize rotations in cases of disruptions. The IFS model for exception weeks could therefore be applicable in operational contexts, when a fast, feasible solution for the disturbance is required, even if optimality is not guaranteed.

References

[1] M. R. Bussieck, T. Winter, and U. T. Zimmermann, "Discrete optimization in public rail transport", *Mathematical Programming*, vol. 79, no. 1, pp. 415–444, Oct. 1997. DOI: 10.1007/BF02614327.

[2] R. M. Lusby, J. Larsen, M. Ehrgott, and D. Ryan, "Railway track allocation: Models and methods", *OR Spectrum*, vol. 33, no. 4, pp. 843–883, Oct. 2011. DOI: 10.1007/s00291-009-0189-0.

[3] B. Grimm, R. Hoogervorst, and R. Borndörfer, "A comparison of two models for rolling stock scheduling", *Transportation Science*, vol. 59, no. 5, Sep. 2025. DOI: 10.1287/trsc.2024.0505.

[4] E. Kere, "Improving the traffic planning process for long-distance passenger train services: A case study", M.S. thesis, Aalto University, Espoo, Aug. 2023. [Online]. Available: https://urn.fi/URN:NBN:fi:aalto-202308275138.

[5] M. Reuther and T. Schlechte, "Optimization of rolling stock rotations", in *Handbook of Optimization in the Railway Industry*, R. Borndörfer, T. Klug, L. Lamorgese, C. Mannino, M. Reuther, and T. Schlechte, Eds., Cham: Springer International Publishing, 2018, pp. 213–241, ISBN: 978-3-319-72153-8. DOI: 10.1007/978-3-319-72153-8_10.

[6] W. Dai, J. Zhang, and X. Sun, "On solving multi-commodity flow problems: An experimental evaluation", *Chinese Journal of Aeronautics*, vol. 30, no. 4, pp. 1481–1492, Aug. 2017. DOI: 10.1016/j.cja.2017.05.012.

[7] P.-J. Fioole, L. Kroon, G. Maróti, and A. Schrijver, "A rolling stock circulation model for combining and splitting of passenger trains", *European Journal of Operational Research*, vol. 174, no. 2, pp. 1281–1297, Oct. 2006. DOI: 10.1016/j.ejor.2005.03.032.

[8] A. Santini, M. Schneider, T. Vidal, and D. Vigo, "Decomposition strategies for vehicle routing heuristics", *INFORMS Journal on Computing*, vol. 35, no. 3, pp. 543–559, May 2023. DOI: 10.1287/ijoc.2023.1288.

[9] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems", in *Principles and Practice of Constraint Programming — CP98*, M. Maher and J.-F. Puget, Eds., Berlin, Heidelberg: Springer, 1998, pp. 417–431. DOI: 10.1007/3-540-49481-2_30.

[10] VR. "Rautatieasemat ja reitit - VR". (in Finnish), (Accessed on Mar. 4, 2026). [Online]. Available: https://www.vr.fi/rautatieasemat-ja-reitit.

[11] VR. "VR teki jälleen kaikkien aikojen ennätyksen: Vuonna 2025 ostettiin 16,1 miljoonaa kaukojunamatkaa". (in Finnish), (Accessed on Jan. 20, 2026). [Online]. Available: https://www.vrgroup.fi/fi/vrgroup/uutiset/vr-teki-jalleen-kaikkien-aikojen-ennatyksen-vuonna-2025-ostettiin-161-miljoonaa-kaukojunamatkaa/.

[12] VR-Yhtymä Oyj, "VR Vuosikertomus 2025", (in Finnish), VR Group, Annual report, Mar. 2025, p. 256. [Online]. Available: https://2025.vrgroupraportti.fi/pdf/VR-vuosikertomus-2025.pdf.

[13] J. Eskola, "Developing an optimization model for VR's long-term locomotive allocation planning", M.S. thesis, Aalto University, Espoo, Mar. 2019. [Online]. Available: https://urn.fi/URN:NBN:fi:aalto-201903172297.

[14] J. Abara, "Applying integer linear programming to the fleet assignment problem", *Interfaces*, vol. 19, no. 4, pp. 20–28, Aug. 1989. DOI: 10.1287/inte.19.4.20.

[15] S. S. Perumal, R. M. Lusby, and J. Larsen, "Electric bus planning & scheduling: A review of related problems and methodologies", *European Journal of Operational Research*, vol. 301, no. 2, pp. 395–413, Sep. 2022. DOI: 10.1016/j.ejor.2021.10.058.

[16] P. Carraresi and G. Gallo, "Network models for vehicle and crew scheduling", *European Journal of Operational Research*, vol. 16, no. 2, pp. 139–151, May 1984. DOI: 10.1016/0377-2217(84)90068-7.

[17] S. Bunte and N. Kliewer, "An overview on vehicle scheduling models", *Public Transport*, vol. 1, no. 4, pp. 299–317, Nov. 2009. DOI: 10.1007/s12469-010-0018-5.

[18] S. Raff, "Routing and scheduling of vehicles and crews: The state of the art", *Computers & Operations Research*, vol. 10, no. 2, pp. 63–211, Jan. 1983. DOI: 10.1016/0305-0548(83)90030-8.

[19] M. Benkirane, F. Clautiaux, J. Damay, and B. Detienne, "A hypergraph model for the rolling stock rotation planning and train selection", 2019. [Online]. Available: https://inria.hal.science/hal-02402447.

[20] M. Peeters and L. Kroon, "Circulation of railway rolling stock: A branch-and-price approach", *Computers & Operations Research*, vol. 35, no. 2, pp. 538–556, Feb. 2008. DOI: 10.1016/j.cor.2006.03.019.

[21] V. Cacchiani, A. Caprara, and P. Toth, "Solving a real-world train-unit assignment problem", *Mathematical Programming*, vol. 124, no. 1, pp. 207–231, Jul. 2010. DOI: 10.1007/s10107-010-0361-y.

[22] Y. Gao, J. Xia, A. D'Ariano, and L. Yang, "Weekly rolling stock planning in Chinese high-speed rail networks", *Transportation Research Part B: Methodological*, vol. 158, pp. 295–322, Apr. 2022. DOI: 10.1016/j.trb.2022.02.005.

[23] S.-P. Hong, K. M. Kim, K. Lee, and B. Hwan Park, "A pragmatic algorithm for the train-set routing: The case of Korea high-speed railway", *Omega*, vol. 37, no. 3, pp. 637–645, Jun. 2009. DOI: 10.1016/j.omega.2008.03.003.

[24] R. Borndörfer et al., "Deutsche Bahn schedules train rotations using hypergraph optimization", *INFORMS Journal on Applied Analytics*, vol. 51, no. 1, pp. 42–62, Feb. 2021. DOI: 10.1287/inte.2020.1069.

[25] P. Thorlacius, J. Larsen, and M. Laumanns, "An integrated rolling stock planning model for the Copenhagen suburban passenger railway", *Journal of Rail Transport Planning & Management*, vol. 5, no. 4, pp. 240–262, Dec. 2015. DOI: 10.1016/j.jrtpm.2015.11.001.

[26] M. A. D. Silva, "Railway planning: Optimization approaches for the rolling stock rotation problem", M.S. thesis, Universidade de Lisboa, Nov. 2023. [Online]. Available: https://scholar.tecnico.ulisboa.pt/records/IBmkLdQpiUe9gL3X5fSDqapXwPc8mDD-UWm8.

[27] M. Reuther, "Mathematical optimization of rolling stock rotations", Ph.D. dissertation, Technischen Universität Berlin, 2017. [Online]. Available: https://depositonce.tu-berlin.de/handle/11303/6309.

[28] F. Prause, "A multi-swap heuristic for rolling stock rotation planning with predictive maintenance", OpenProceedings.org, 2024. DOI: 10.48786/INOC.2024.11.

[29] B. Grimm, R. Borndörfer, and J. Bushe, "Assignment based resource constrained path generation for railway rolling stock optimization", *OASIcs, Volume 115, ATMOS 2023*, vol. 115, D. Frigioni and P. Schiewe, Eds., pp. 1–15, 2023. DOI: 10.4230/OASICS.ATMOS.2023.13.

[30] R. Borndörfer, M. Reuther, T. Schlechte, and S. Weider, "A hypergraph model for railway vehicle rotation planning", *OASIcs, Volume 20, ATMOS 2011*, vol. 20, A. Caprara and S. Kontogiannis, Eds., pp. 146–155, 2012. DOI: 10.4230/OASICS.ATMOS.2011.146.

[31] P. Päprer et al., "The rolling stock circulation planning problem: A tutorial with case study", *Public Transport*, Oct. 2025. DOI: 10.1007/s12469-025-00408-8.

[32] Z. Lin and R. S. K. Kwan, "An integer fixed-charge multicommodity flow (FCMF) model for train unit scheduling", *Electronic Notes in Discrete Mathematics*, vol. 41, pp. 165–172, Jun. 2013. DOI: 10.1016/j.endm.2013.05.089.

[33] K. Salimifard and S. Bigharaz, "The multicommodity network flow problem: State of the art classification, applications, and solution methods", *Operational Research*, vol. 22, no. 1, pp. 1–47, Mar. 2022. DOI: 10.1007/s12351-020-00564-8.

[34] A. Alfieri, R. Groot, L. Kroon, and A. Schrijver, "Efficient circulation of railway rolling stock", *Transportation Science*, vol. 40, no. 3, pp. 378–391, 2006. DOI: 10.1287/trsc.1060.0155.

[35] VR. "Pendolino trains services and coach map", (Accessed on Jan. 6, 2026). [Online]. Available: https://www.vr.fi/en/trains/pendolino.

[36] VR. "InterCity trains coach map and facilities", (Accessed on Jan. 6, 2026). [Online]. Available: https://www.vr.fi/en/trains/intercity.

[37] G. L. Giacco, A. D'Ariano, and D. Pacciarelli, "Rolling stock rostering optimization under maintenance constraints", *Journal of Intelligent Transportation Systems*, vol. 18, no. 1, pp. 95–105, Jan. 2014. DOI: 10.1080/15472450.2013.801712.

[38] J. C. Wagenaar, L. G. Kroon, and M. Schmidt, "Maintenance appointments in railway rolling stock rescheduling", *Transportation Science*, vol. 51, no. 4, pp. 1138–1160, Nov. 2017. DOI: 10.1287/trsc.2016.0701.

[39] R. Borndörfer, M. Reuther, T. Schlechte, K. Waas, and S. Weider, "Integrated optimization of rolling stock rotations for intercity railways", *Transportation Science*, vol. 50, no. 3, pp. 863–877, Aug. 2016. DOI: 10.1287/trsc.2015.0633.

[40] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network flows: theory, algorithms, and applications*. Upper Saddle River, NJ: Prentice Hall, 1993, pp. 649–654, ISBN: 978-0-13-617549-0.

[41] L. R. Ford, *Flows in networks* (Princeton Landmarks in Mathematics and Physics Series), 1st ed. Princeton: Princeton University Press, 2010, pp. 3–5, ISBN: 978-0-691-27343-3.

[42] B. Grimm, R. Borndörfer, M. Reuther, and T. Schlechte, "A cut separation approach for the rolling stock rotation problem with vehicle maintenance", *OASIcs, Volume 75, ATMOS 2019*, Open Access Series in Informatics (OASIcs), vol. 75, V. Cacchiani and A. Marchetti-Spaccamela, Eds., pp. 1–12, 2019. DOI: 10.4230/OASICS.ATMOS.2019.1.

[43] Gurobi Optimization, LLC. "Parameter Reference - LazyConstraints", (Accessed on Apr. 26, 2026). [Online]. Available: https://docs.gurobi.com/projects/optimizer/en/current/reference/parameters.html#lazyconstraints.

[44] Gurobi Optimization, LLC. "MIP starts - Gurobi Example Tour", (Accessed on Jan. 20, 2026). [Online]. Available: https://docs.gurobi.com/projects/examples/en/current/overview/starts.html.

[45] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: theory and algorithms*, 3rd ed. Hoboken, N.J: Wiley-Interscience, 2006, ISBN: 978-0-471-48600-8. DOI: 10.1002/0471787779.

[46] R. Bent and P. Van Hentenryck, "Spatial, temporal, and hybrid decompositions for large-scale vehicle routing with time windows", in *Principles and Practice of Constraint Programming – CP 2010*, D. Cohen, Ed., Berlin, Heidelberg: Springer, 2010, pp. 99–113. DOI: 10.1007/978-3-642-15396-9_11.

[47] A. S. Pepin, G. Desaulniers, A. Hertz, and D. Huisman, "A comparison of five heuristics for the multiple depot vehicle scheduling problem", *Journal of Scheduling*, vol. 12, no. 1, pp. 17–30, Feb. 2009. DOI: 10.1007/s10951-008-0072-x.

[48] Gurobi Optimization, LLC. “Gurobi Optimizer Reference Manual”, (Accessed on Jan. 20, 2026). [Online]. Available: https://www.gurobi.com.

[49] LLC Gurobi Optimization. “Managing Threads and Memory Usage in Gurobi”, (Accessed on Feb. 16, 2026). [Online]. Available: https://support.gurobi.com/hc/en-us/articles/42058344459409-Managing-Threads-and-Memory-Usage-in-Gurobi.

[50] A. Ceder, “Optimal multi-vehicle type transit timetabling and vehicle scheduling”, *Procedia - Social and Behavioral Sciences*, The State of the Art in the European Quantitative Oriented Transportation and Logistics Research – 14th Euro Working Group on Transportation & 26th Mini Euro Conference & 1st European Scientific Conference on Air Transport, vol. 20, pp. 19–30, Jan. 2011. DOI: 10.1016/j.sbspro.2011.08.005.

[51] Gurobi Optimization, LLC. “Concurrent Optimization”, (Accessed on Apr. 26, 2026). [Online]. Available: https://docs.gurobi.com/projects/optimizer/en/current/features/concurrent.html.

[52] J. Heil, K. Hoffmann, and U. Buscher, “Railway crew scheduling: Models, methods and applications”, *European Journal of Operational Research*, vol. 283, no. 2, pp. 405–425, Jun. 2020. DOI: 10.1016/j.ejor.2019.06.016.

A If-statements as linear constraints

Consider a constraint with bounded variables x and y and indicator variable $z \in \{0, 1\}$.

$$x = y \qquad \text{if } z = 1 \tag{A1}$$

If $z = 0$, the constraint is not included. The constraint can be reformulated as linear constraints

$$x - y \leq M(1 - z) \tag{A2}$$

$$x - y \geq -M(1 - z), \tag{A3}$$

where M is a sufficiently large constant w.r.t. the bounds on variables x and y.

B Orientation constraint via penalty function

Modified orientation constraint with slack variable $k_{t'}$:

$$o_t + O(t) = 2n_{t'} + k_{t'} - 1 \text{ if dep(t') = HKI \& } f_{(t,t')} = 1 \qquad \forall (t, t') \in F \tag{B1}$$

$$k_{t'} \in \{0, 1\} \qquad \forall t' \in T \tag{B2}$$

Modified orientation constraint of destination trains with slack variable k_d:

$$o_t + O(t) + O(t, d) = 2n_d + k_d - S_o(d) \text{ if } f_{(t,d)} = 1 \qquad \forall (t, d) \in F_D \tag{B3}$$

$$k_d \in \{0, 1\} \qquad \forall d \in D \tag{B4}$$

Term added to the objective with constant penalty $w_{k_{t'}}$ to penalize incorrect orientations

$$w_{k_{t'}} \cdot \sum_{t' \in T} k_{t'} \tag{B5}$$

Term added to the objective with constant penalty w_{k_d} to penalize incorrect orientations of the compositions operating destination trains

$$w_{k_d} \cdot \sum_{d \in D} k_d \tag{B6}$$

C Complete models

C.1 Full standard week model

$$
\begin{aligned}
\min \quad & \sum_{t\in T}\sum_{c\in C} a_{t,c}\cdot OC(t,c) \\
& + \sum_{t\in T}\sum_{c\in C} m_{t,c}\cdot A(t) \\
& + \sum_{(t,t')\in F} f_{(t,t')}\cdot TP(t,t') \\
& + w_{d_t^*}\cdot \sum_{t\in T} d_t^* \\
& + w_{d_t^r}\cdot \sum_{t\in T_R} d_t^r \\
\text{s.t.} \quad & \sum_{c\in C} a_{t,c} = 1 && \forall t\in T, \\
& \sum_{t':(t',t)\in F} f_{(t',t),c} = a_{t,c} && \forall t\in T, c\in C, \\
& \sum_{t':(t,t')\in F} f_{(t,t'),c} = a_{t,c} && \forall t\in T, c\in C, \\
& f_{(t',t)} = \sum_{c\in C} f_{(t',t),c} && \forall (t,t')\in F, \\
& m_{t,c} \geq a_{t,c}\cdot D(t) - C(c) && \forall t\in T, c\in C, \\
& r_t = \sum_{t':(t',t)\in F}\sum_{c\in C} f_{(t',t),c}\cdot M(t',t) && \forall t\in T, \\
& d_t \leq M(1-r_t) && \forall t\in T, \\
& d_{t'} \geq d_t + Km(t) - M\cdot(1-\sum_{c\in C} f_{(t,t'),c}) - M\cdot r_{t'} && \forall (t,t')\in F, \\
& d_t + Km(t) - d_t^* \leq U_m && \forall t\in T, \\
& o_t = 0 \text{ if } dep(t) = HKI && \forall t\in T, \\
& o_{t'} = o_t + O(t) + O(t,t') \text{ if } dep(t')\neq HKI \ \&\ f_{(t,t')} = 1 && \forall (t,t')\in F, \\
& o_t + O(t) = 2n_{t'} - 1 \text{ if } dep(t') = HKI \ \&\ f_{(t,t')} = 1 && \forall (t,t')\in F, \\
& \sum_{\substack{c\in C \text{ s.t.} \\ ERd,\ S\notin T(c)}} a_{t,c}\cdot Km(t) - d_t^r \leq L_{R,km} && \forall t\in T_R, \\
& \sum_{(t,t')\in F^*}\sum_{IC\in T(c)} f_{(t,t'),c} \leq U_{IC}, \\
& \sum_{(t,t')\in F^*}\sum_{S\in T(c)} f_{(t,t'),c} \leq U_S,
\end{aligned}
$$

$$\sum_{(t,t')\in F^*} \sum_{ERd\in T(c)} f_{(t,t'),c} \leq U_{ERd},$$

$$\sum_{(t,t')\in F^*} \sum_{Ed\in T(c)} f_{(t,t'),c} \cdot Ed(c) \leq U_{Ed},$$

$$\begin{aligned}
a_{t,c} &\in \{0,1\} && \forall t \in T, c \in C,\\
r_t &\in \{0,1\} && \forall t \in T,\\
f_{(t,t'),c} &\in \{0,1\} && \forall (t,t') \in F, c \in C,\\
f_{(t,t')} &\in \{0,1\} && \forall (t,t') \in F,\\
m_{t,c} &\in \mathbb{N} && \forall t \in T, c \in C,\\
o_t, n_t &\in \mathbb{N} && \forall t \in T,\\
d_t, d_t^*, d_t^r &\geq 0 && \forall t \in T
\end{aligned}$$

C.2 Full exception week model

$$\begin{aligned}
\min \quad & \sum_{t\in T} \sum_{v\in V} a_{t,v} \cdot OC(t,v)\\
& + \sum_{t\in T} \sum_{v\in V} m_{t,v} \cdot A(t)\\
& + \sum_{(t,t')\in F} f_{(t,t')} \cdot TP(t,t')\\
& + w_{d_t^*} \cdot \sum_{t\in T\cup D} d_t^*\\
& + w_{d_t^r} \cdot \sum_{t\in T_R} d_t^r\\
& + w_{v^*} \cdot \sum_{(d,v^*)\in T_v^*} \sum_{\substack{v\in V \text{ s.t.}\\ v\neq v^*}} a_{d,v}
\end{aligned}$$

$$\begin{aligned}
\text{s.t.} \quad \sum_{v\in V} a_{t,v} &= 1 && \forall t \in T \cup S \cup D,\\
a_{t,v^*} &= 1 && \forall (t, v^*) \in T_v^*, \text{s.t. } t \in S,\\
a_{s,v} &= \sum_{t:(s,t)\in F_S} f_{(s,t),v} && \forall s \in S, v \in V,\\
a_{d,v} &= \sum_{t:(t,d)\in F_D} f_{(t,d),v} && \forall d \in D, v \in V,\\
a_{t,v} &= \sum_{t':(t',t)\in F_S\cup F} f_{(t',t),v} && \forall t \in T, v \in V,\\
a_{t,v} &= \sum_{t':(t,t')\in F\cup F_D} f_{(t,t'),v} && \forall t \in T, v \in V,\\
f_{(t,t')} &= \sum_{v\in V} f_{(t,t'),v} && \forall (t,t') \in F \cup F_S \cup F_D,
\end{aligned}$$

$$
\begin{aligned}
m_{\mathrm{t,v}} &\geq a_{\mathrm{t,v}} \cdot D(\mathrm{t}) - C(\mathrm{v}) && \forall \mathrm{t} \in \mathrm{T}, \mathrm{v} \in \mathrm{V}, \\
1 &\leq \sum_{(\mathrm{t},\mathrm{t}') \in \mathrm{F}} f_{(\mathrm{t},\mathrm{t}'),\mathrm{v}} \cdot H(\mathrm{t},\mathrm{t}') && \forall \mathrm{v} \in \mathrm{V}, \\
r_{\mathrm{t}} &= \sum_{\mathrm{t}':(\mathrm{t}',\mathrm{t}) \in \mathrm{F_S} \cup \mathrm{F}} \sum_{\mathrm{v} \in \mathrm{V}} f_{(\mathrm{t}',\mathrm{t}),\mathrm{v}} \cdot M(\mathrm{t}',\mathrm{t}) && \forall \mathrm{t} \in \mathrm{T} \cup \mathrm{D}, \\
d_{\mathrm{s}} &= P_d(\mathrm{s}) && \forall \mathrm{s} \in \mathrm{S}, \\
d_{\mathrm{t}} &\leq \mathrm{M}(1 - r_{\mathrm{t}}) && \forall \mathrm{t} \in \mathrm{T} \cup \mathrm{D}, \\
d_{\mathrm{t}'} &\geq d_{\mathrm{t}} + Km(\mathrm{t}) - \mathrm{M} \cdot (1 - \sum_{\mathrm{v} \in \mathrm{V}} f_{(\mathrm{t}',\mathrm{t}),\mathrm{v}}) - \mathrm{M} \cdot r_{\mathrm{t}'} && \forall (\mathrm{t}, \mathrm{t}') \in \mathrm{F} \cup \mathrm{F_S} \cup \mathrm{F_D}, \\
d_{\mathrm{t}} + Km(\mathrm{t}) - d^*_{\mathrm{t}} &\leq \mathrm{U_m} && \forall \mathrm{t} \in \mathrm{T}, \\
d_{\mathrm{d}} + S_d(\mathrm{d}) - d^*_{\mathrm{d}} &\leq \mathrm{U_m} && \forall \mathrm{d} \in \mathrm{D}, \\
o_{\mathrm{s}} &= P_o(\mathrm{s}) && \forall \mathrm{s} \in \mathrm{S}, \\
o_{\mathrm{t}} &= 0 \text{ if } \mathrm{dep(t)} = \mathrm{HKI} && \forall \mathrm{t} \in \mathrm{T}, \\
o_{\mathrm{t}'} &= o_{\mathrm{t}} + O(\mathrm{t}) + O(\mathrm{t},\mathrm{t}') \text{ if } \mathrm{dep}(\mathrm{t}') \neq \mathrm{HKI} \ \ \& \ \ \mathrm{f}_{(\mathrm{t},\mathrm{t}')} = 1 && \forall (\mathrm{t}, \mathrm{t}') \in \mathrm{F} \cup \mathrm{F_S}, \\
o_{\mathrm{t}} + O(\mathrm{t}) &= 2n_{\mathrm{t}'} - 1 \text{ if } \mathrm{dep}(\mathrm{t}') = \mathrm{HKI} \ \& \ f_{(\mathrm{t},\mathrm{t}')} = 1 && \forall (\mathrm{t}, \mathrm{t}') \in \mathrm{F}, \\
o_{\mathrm{t}} + O(\mathrm{t}) + O(\mathrm{t},\mathrm{d}) &= 2n_{\mathrm{d}} - S_o(\mathrm{d}) \text{ if } f_{(\mathrm{t},d)} = 1 && \forall (\mathrm{t}, \mathrm{d}) \in \mathrm{F_D}, \\
\sum_{\substack{\mathrm{v} \in \mathrm{V} \\ \mathrm{ERd},\ \mathrm{S} \notin T(\mathrm{v})}} & a_{\mathrm{t,v}} \cdot Km(\mathrm{t}) - d^r_{\mathrm{t}} \leq \mathrm{L_{R,km}} && \forall \mathrm{t} \in \mathrm{T_R}, \\
a_{\mathrm{t,v}} &\in \{0, 1\} && \forall \mathrm{t} \in \mathrm{T} \cup \mathrm{S} \cup \mathrm{D}, \mathrm{v} \in \mathrm{V}, \\
f_{(\mathrm{t},\mathrm{t}'),\mathrm{v}} &\in \{0, 1\} && \forall (\mathrm{t}, \mathrm{t}') \in \mathrm{F}, \mathrm{v} \in \mathrm{V}, \\
f_{(\mathrm{t},\mathrm{t}')} &\in \{0, 1\} && \forall (\mathrm{t}, \mathrm{t}') \in \mathrm{F}, \\
r_{\mathrm{t}} &\in \{0, 1\} && \forall \mathrm{t} \in \{\mathrm{T} \cup \mathrm{D}\}, \\
m_{\mathrm{t,v}} &\in \mathbb{N} && \forall \mathrm{t} \in \mathrm{T}, \mathrm{v} \in \mathrm{V}, \\
d_{\mathrm{t}} &\geq 0 && \forall \mathrm{t} \in \{\mathrm{S} \cup \mathrm{T}\}, \\
d^*_{\mathrm{t}} &\geq 0 && \forall \mathrm{t} \in \{\mathrm{T} \cup \mathrm{D}\}, \\
o_{\mathrm{t}} &\in \mathbb{N} && \forall \mathrm{t} \in \{\mathrm{S} \cup \mathrm{T}\}, \\
n_{\mathrm{t}} &\in \mathbb{N} && \forall \mathrm{t} \in \{\mathrm{T} \cup \mathrm{D}\}, \\
d^r_{\mathrm{t}} &\geq 0 && \forall \mathrm{t} \in \mathrm{T}
\end{aligned}
$$