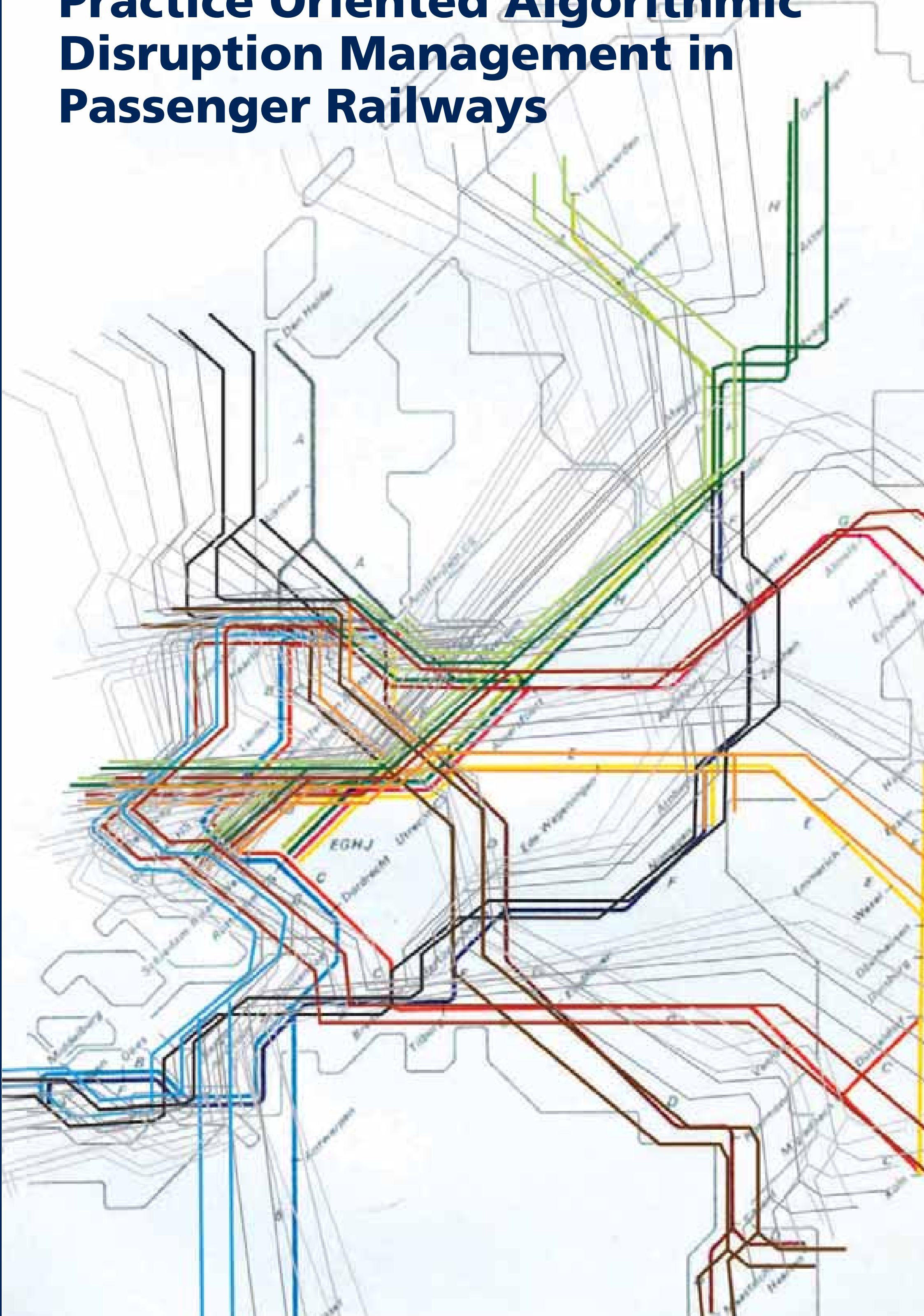
JORIS CAMIEL WAGENAAR

Practice Oriented Algorithmic Disruption Management in Passenger Railways



PRACTICE ORIENTED ALGORITHMIC DISRUPTION MANAGEMENT IN PASSENGER RAILWAYS

PRACTICE ORIENTED ALGORITHMIC DISRUPTION MANAGEMENT IN PASSENGER RAILWAYS

Praktijk georiënteerde bijsturing van reizigerstreinen

——— Thesis ———

to obtain the degree of Doctor from the
Erasmus University Rotterdam
by command of the

rector magnificus
Prof.dr. H.A.P. Pols

and in accordance with the decision of the Doctorate Board

The public defense shall be held on 08-09-2016 at 11.30h

by

JORIS CAMIEL WAGENAAR
born in Rotterdam, the Netherlands.

Erasmus University Rotterdam

Doctoral Committee

Promotor
Prof. dr. L.G. Kroon
Prof. dr. A.P.M. Wagelmans

Other members
Prof. dr. D. Huisman
Prof. dr. R. Borndörfer
Dr. R.M. Lusby

Erasmus Research Institute of Management – ERIM
The joint research institute of the Rotterdam School of Management (RSM)
and the Erasmus School of Economics (ESE) at the Erasmus University Rotterdam
Internet: http://www.erim.eur.nl

ERIM Electronic Series Portal: http://hdl.handle.net/1765/1

ERIM PhD Series in Research in Management, Nummer 390
ERIM reference number – EPS-2016-390-LIS
ISBN 978-90-5892-452-0

Design cover – PanArt, www.panart.nl
Illustration cover – Renssen ontwerpbureau, www.renssen.nl

This publication (cover and interior) is printed by Tuijtel on recycled paper, BalanceSilk®
The ink used is produced from renewable resources and alcohol free fountain solution.
Certifications for the paper and the printing production process: Recycle, EU Ecolabel, FSC®, ISO14001.
More info: www.tuijtel.com

This thesis is dedicated to my father who passed away on 25 Februari 2007.
——— *I know he would have been very proud.*

Acknowledgments

I have been using the train as mode of transportation for many years in my life. It usually was a pleasant expierence. However, I still remember some frustrating journeys with long delays. Therefor, it was a nice surprise to have been given the opportunity to do a PhD on disruption management within passenger railway transportation. The result of this research now lies before you and I would like to thank all the people that were involved and supported me.

First of all I would like to thank Leo for providing me this opportunity and for his valuable feedback. I enjoyed our many meetings and detailled discussions from both a practical and theoretical point of view. I could not have wished for a better supervisor. You were always there to provide me with very detailled comments on every version of a paper I sent to you. Thank you for all your time and effort!

Furthermore, I would like to thank Dennis, Twan, Luuk, Jørgen, Richard, and Marie for working together on the papers in this thesis and on other collaborations. I have learned a lot from all of you and I am very grateful to have met each and every one of you.

I would like to thank Richard and Jørgen for hosting me at the DTU for three months. It was a pleasure to have collaborated with you during those months. I really enjoyed the environment at the DTU and the nice discussions we had over there.

I have been a PhD student at the Department of Technology & Operations Manage- ment of RSM. I would like to thank the department and all colleagues within it for the talks and the fun. Evelien thanks for being an awesome roommate for three years, it was a lot of fun! Luuk, Paul, Clint, and Evelien thanks for the laughs and during the conferences. It was always a pleasure spending time with you guys. Tim, thank you for all the coffee breaks and for being my paranymph. Wouter, Panos, Nick, Gert-Jaap, and all other PhD students: it was a pleasure meeting you and I wish you all the best!

Besides my department TOM at RSM, the research schools ERIM and TRAIL helped me during my PhD as well. They organised conferences, courses, social activities, and aided me with financial support for my development. IN this way I want to express my gratitude to them. Furthermore, I had close contact with the Econometric Institute (Albert, Dennis, Twan, Remy, Judith, Rommert, Wilco, Willem, Ilse, Charlie, and Harwin). Thanks to them I always felt very welcome in the H-building.

Judith, Rommert, Wilco, Willem, Ilse, Charlie, and Harwin). Thanks to them I always felt very welcome in the H-building.

My PhD thesis is now finished and I would like to thank my inner commitee members Dennis Huisman, Richard Lusby, and Ralf Borndörfer for taking their time to read my thesis and provide my with comments that have improved this thesis.

Finally, but most importantly, I would like to thank my family and friends for their support. My mother who has always been very proud and who has supported me throughout all times. She also designed the titlepage, for which I am very grateful. My brother Nils for his support and for being my paranymph. Most importantly, I want to thank Nathalie for helping and supporting me during the whole PhD trajectory. Besides making me always very happy, she has also read all my papers and given valuable comments to them. Thank you very much for everything!

Rotterdam, April 2016
Joris Wagenaar

Table of Contents

1 – Introduction

1.1 – MOTIVATION

The most commonly used passenger transportation mode is currently road transportation. This holds true for both leisure activities and for commuter traffic. However, due to the negative effects on the environment and due to the large number of traffic jams in the morning and afternoon rush, governments aim to reduce the number of cars on the road. One of the key ingredients to reduce road transportation is to increase the utilization of public transportation. In order to do so, the public transport sector should, among other factors, increase the quality of their services. The quality of a public transport system is measured by, among other factors, its ability to transport passengers with a seat from their origin to their destination on time.

In a perfect world, the public transportation services are always on time and provide enough capacity for every passenger to sit. As a consequence, the usage of the public transportation service gives a high passenger satisfaction and the passengers are willing to use the public transport the next time again. However, in practice the world is not perfect. Passengers might face large delays and/or seat shortages. As a result, the passengers who own a car may switch to their car for their next journey.

Unfortunately, unexpected incidents are unavoidable in public transportation services. If these unexpected incidents have a large influence on the system, they are called disruptions. Examples of disruptions are accidents, malfunctioning infrastructure or rolling stock, and extreme weather conditions. It is of high importance to be able to react on these unforeseen events in such a way that as much of the passenger service is upheld as possible. This is called *Disruption Management*. In this thesis the focus is on disruption management for passenger railway transportation. Note, however, that all presented models can be translated to other public transport modes (e.g. subway) on rail as well. The models have to slightly adjusted, according to the corresponding rules, in order to use them on public transportation modes that are not on rail. There are three major resource schedules essential in rail transportation: The timetable, the rolling stock circulation, and the crew schedule.

The timetable consists of arrival and departure times of trains at stations. The rolling stock circulation contains the rolling stock compositions appointed to the trips to meet the expected passenger demand. This appointment of compositions to trips can be translated to duties for every physical train unit. The train drivers and conductors are assigned to the trains on the trips in the crew schedule. This assignment can also be translated to crew duties for every crew member. These three resource schedules are created long before the operations take place. The timetable is communicated to the passengers such that they can schedule their journey. However, if a disruption occurs during the daily operations, it means that (some of) the tracks between two stations are blocked. As a consequence, the planned resource schedules become infeasible and they need to be rescheduled. The objective is to find new schedules where the passengers are faced with the least amount of inconvenience. This is mainly achieved by minimizing the amount of cancelled train services and by aiming to appoint the rolling stock to trips such that the number of seat shortages are minimal.

In current practice there is a lack of decision support systems during disruptions. Here, decision support systems are mathematical models programmed into computerized tools to support the railway operators. Consequently, the three major resource schedules are usually rescheduled manually with very little automated support. As a consequence, the first concern of the operators is to make the timetable, rolling stock circulation, and crew schedule feasible for the earliest minutes to come. In this way they just want to survive the coming minutes. By focusing on the minutes to come, the operators might postpone the problems during a disruption. Furthermore, by only focussing on local decisions for the minutes to come, the resulting resource schedules may be of bad quality. It might be that more trains are cancelled than necessary. Decisions in the first phase of the disruption have a large influence on the consequences of the disruption on the railway network. The probability of not being able to find an overall feasible solution without cancelling additional trips later in time is substantial. Furthermore, the passenger demand is not fully taken into account during disruptions when rescheduling only the minutes to come. The current way of reasoning is as follows: Any train is better than no train. However, a small train can, for example, still lead to angry passengers who do not fit in the train. Therefore, railway operators are looking for decision support systems to aid their dispatchers in finding new resource schedules during disruptions. It is expected that the throughput time and quality of disruption management can be improved by using computerized support.

In academic literature, disruption management is handled with a three stage procedure. In the first stage a new timetable is developed, whereafter in the second stage, with the new timetable as input, the rolling stock circulation is rescheduled, and in the final stage, with both the new timetable and the new rolling stock circulation as input, the crew schedule is altered. Existing literature focusses on one (or at most two) step(s) of disruption management. For instance, Veelenturf et al. (2015) and Zhan et al. (2015) are two papers focussing on timetable rescheduling. Papers on rolling stock rescheduling are for example: Nielsen et al. (2012), Haahr et al. (2014), and Sato and Fukumura (2012). Papers on rescheduling the crew also exist, see for instance Rezanova and Ryan (2009), Potthoff et al. (2010), and Abbink et al. (2009). Some papers investigate an integrated model for rescheduling two resource schedules at the same time: Walker et al. (2005) integrate timetable and crew rescheduling, and Cadarso et al. (2013, 2015) integrate timetable and rolling stock rescheduling. An overview of the existing literature on disruption management can be found in Cacchiani et al. (2014).

Unfortunately, there are still practical aspects lacking in the currently available models. As a consequence, those models are not fully applicable on their own. Therefore, the focus in this thesis is on developing algorithmic support for disruption management while including many practical aspects, such that the models can better support the dispatchers during a disruption.

1.2 – RESEARCH QUESTION(S)

The central research question of this thesis is:

"Which relevant operational details have not yet been included in the models for real-time disruption management processes, and how can these white spots in the existing models be filled?"

Real-time disruption management are the processes taking place from the time of the occurrence of the disruption until the moment the new resource schedules are operational again. In this time the three resource schedules need to be rescheduled in order to secure their feasibility. This thesis develops models and algorithms to support this process. These models and algorithms can then be turned into decision support systems.

Existing models and algorithms from literature cannot yet be turned into decision support models for usage in practice due to the lack of certain real-life aspects. In this thesis five important practical aspects are included in the disruption management models:

1. Creating a macroscopic globally feasible solution for all three resource schedules, instead of focussing on one individual resource schedule. Here, macroscopic means that the infrastructure is taken into account on a high level of abstraction.
2. Taking maintenance appointments required by certain rolling stock units into account while rescheduling.
3. Dead-heading trips to transfer rolling stock units from stations with a surplus of inventory to stations with a shortage of inventory are allowed to reduce the number of additional cancelled trips.
4. Adjusted passenger demand is taken into account, because the passenger demand is not static, but depends on the capacity appointed to the previous trips.

5. Checking whether a rolling stock circulation is feasible with respect to the available depot tracks (the shunting yard) within a station.

1.3 – OVERVIEW OF THE CONDUCTED RESEARCH

In this section an overview of the conducted research is presented by means of a short summary for the Chapters 2 to 6.

Chapter 2: An iterative framework for real-time railway rescheduling. T.A.B. Dollevoet, D. Huisman, L.G. Kroon, L.P. Veelenturf, and J.C. Wagenaar. Under review at Computers & Operations Research.

In this chapter the first aspect from the list in the previous section that is currently lacking in disruption management models is identified and included. It has always been assumed that combining the results of the individual steps of the disruption management process will lead to an overall feasible solution during a disruption. This has, however, never been tested. Chapter 2 describes an iterative framework in which all three resource schedules are considered. This framework is extensively tested on instances from Netherlands Railways. Results demonstrate that an overall feasible solution can be obtained by solving the three stages individually in an acceptable amount of time. As a result, the individual modules for rescheduling during a disruption can indeed be applied in practice by means of an iterative framework.

Chapter 3: A comparison of two exact methods for passenger railway rolling stock (re)scheduling. J.T. Haahr, J.C. Wagenaar, L.P. Veelenturf and L.G. Kroon. Under review at Transportation Research Part E: Logistics and Transportation Review.

In Chapter 3 two different approaches to solve the rolling stock (re)scheduling problem are compared with each other. The first approach is based on a Mixed Integer Linear Program (MILP) solved by using Cplex. The second approach is based on a different MILP, and now solved by an extension of an existing column generation approach. The approaches are compared and bench marked on various instances from railway operators in different countries. The approaches are used to create the daily schedules and to reschedule the schedules during a disruption. The results demonstrate that both approaches can be used on networks of different operators in two different countries. Furthermore, the results demonstrate that both models are fast enough to be used in a real-time setting.

Chapter 4: Maintenance appointments in railway rolling stock rescheduling. J.C. Wagenaar, L.G. Kroon, and M. Schmidt. Revision is under review at Transportation Science.

Chapter 4 addresses the Rolling Stock Rescheduling Problem (RSRP), while taking maintenance appointments into account. After a disruption, the rolling stock of passenger trains has to be rescheduled in order to restore a feasible rolling stock circulation. A limited number of rolling stock units have a scheduled maintenance appointment during the day: These appointments need to be taken into account while rescheduling. In this paper we propose three different models for this. The Extra Unit Type model extends the Composition model, which aims at rescheduling the rolling stock without maintenance appointments, by adding an additional rolling stock type for every rolling stock unit that requires maintenance. The Shadow-Account model keeps track of a shadow account for all units that require maintenance. The Job-Composition model creates paths for train units such that maintenance units are on time for their maintenance appointment. All models are tested on various instances of Netherlands Railways. The results show that the Shadow Account and Job Composition model are able to efficiently take maintenance appointments into account during rescheduling. Depending on the characteristics of an instance, either the Shadow-Account or the Job-Composition model performs best.

Chapter 5: Rolling stock rescheduling in passenger railway transportation using dead-heading trips and adjusted passenger demand. J.C. Wagenaar and L.G. Kroon. Under review at Transportation Research part B: Methodological.

Chapter 5 discusses two other practical aspects. First of all, unscheduled dead-heading trips are currently excluded from the rolling stock rescheduling models in current literature. A dead-heading trip is an empty train sent from one station to another to increase the local inventory during a disruption. The rolling stock resides at different stations/ tracks throughout the whole country during the day. As a consequence, some stations may have a surplus of units in inventory, while other stations have a shortage during a disruption. Dead-heading trips have already been used in practice to decrease the number of cancelled trains. Chapter 5 includes these dead-heading trips in the RSRP. Furthermore, this chapter introduces the concept of adjusted passenger demand. The major objective of the RSRP models is to uphold as much of the passenger service as possible. This is done by cancelling as few trains as possible and by appointing rolling stock compositions with enough capacity to meet the passenger demand on the trips. However, the passenger demand depends on which compositions are actually appointed to the trips (e.g. cancelling a trip leads to a larger passenger demand on the next trip). This is taken into account in the proposed model as well. Results demonstrate that the model is able to take both dead-heading trips and adjusted passenger demand into account, while still being able to solve the problem in a reasonable computation time.

Chapter 6: A comparison of optimization methods for solving the depot matching and parking problem. J.T. Haahr, R.M. Lusby, and J.C. Wagenaar. Under review at Transportation Science.

The final practical aspect that is treated in this thesis relates to the Train Unit Shunting Problem (TUSP). Up to this point it is assumed that all scheduled shunting movements are feasible with respect to the available depot tracks at every station. In Chapter 6 different methods are developed to assign physical train units to scheduled train services in such a way that the resulting shunting yard operations are feasible. It involves matching train units to arriving and departing train services at a station, as well as assigning the selected matching to a specific depot track. The chapter presents a benchmark of multiple solution approaches. To this end, an approach based on constraint programming, an approach based on column generation, and a randomized greedy construction heuristic are newly developed. These approaches are compared with slightly adjusted existing methods based on a mixed integer program and a two-stage heuristic. The models are tested on multiple real-life instances provided by either Netherlands Railways and Danish State Railways. There is no model that outperforms all other models. It depends on the characteristics of the instance which model can be used best. Therefor, we demonstrate the strengths and weaknesses of each of the approaches.

1.4 – DECLARATION OF CONTRIBUTION

This section states the contribution of the author.

Chapter 1: The work in this chapter has been done by the author of this dissertation.

Chapter 2: The author of this dissertation is one of the authors of the paper included in Chapter 2. The contribution to this paper was the following: developing and programming the main part of the iterative framework, writing the main part of the framework section, interpreting the results, writing the results section, and finally receiving and including feedback from the other authors.

Chapter 3: The author of this dissertation is one of the main authors of the paper included in Chapter 3. The contribution of the author to this paper was the following: collecting the data set from Netherlands Railways, analyzing the data set from Netherlands Railways, implementing the model solved by Cplex, verifying the solution methods by inspecting the results from both methods, performing benchmarks and experiments, and being main author of the sections: composition model and computational experiments.

Chapter 4: The author of this dissertation is the main author of this paper.

Chapter 5: The author of this dissertation is the main author of this paper.

Chapter 6: The author of this dissertation is one of the three main authors of this paper. The contribution of the author was the following: collecting data and analyzing the data sets from Netherlands Railways, developing and programming the constraint programming formulation, verifying the solution methods by inspecting results from all methods, performing benchmarks and experiments, and being main author of the sections: literature overview, problem description, and constraint programming method.

Chapter 7: The work in this chapter has been carried out by the author. The feedback from the promotor has been incorporated in this chapter and in all other chapters as well.

1.5 – OUTLINE

The coming chapters consist of the papers as described before. All these chapters contain the original version of the papers. As a consequence, all chapters are autonomous and some of the notation, definitions, and introductions within chapters can be overlapping. Finally, Chapter 7 presents conclusions and final remarks of this thesis.

2 – Application of an Iterative Framework for Real-time Railway Rescheduling

This chapter considers the paper (Dollevoet et al. (2015)) which is under review at Computers & Operations Research. The research leading to this paper has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) in the ON-TIME project under Grant Agreement SCP1-GA-2011-285243.

Co-authors: T.A.B. Dollevoet, D. Huisman, L.G. Kroon and L.P. Veelenturf

2.1 – INTRODUCTION

Railway transportation plays an important role in the lifes of many people. They travel by train to their work or school, or for leisure purposes. One of the most important criteria for passenger satisfaction is the reliability of the journeys. However, disruptions like accidents, malfunctioning infrastructure or rolling stock, or crew unavailability are inevitable in a railway system. As a consequence, passengers face cancelled, delayed or overcrowded train services. It is very important for railway operators to reduce the nuisance caused by these disruptions for the passengers as much as possible.

As stated in the overview paper by Cacchiani et al. (2014): "*the development of algorithmic real-time railway rescheduling methods is currently still mainly an academic field, where the research is still far ahead of what has been implemented in practice.*" The models and algorithms from literature mainly deal with rescheduling either the timetable, or the rolling stock, or the crew. It is currently unknown whether it is possible to combine the algorithms for individual resources and come up with an overall feasible solution that is satisfactory for the passengers. A solution is overall feasible if the three resource schedules are mutually compatible. This means that both rolling stock and crew are available for each trip in the timetable. This might be one of the reasons why the models from literature have not been implemented in practice yet.

In this paper, we make a first step in bridging this gap between theory and practice, by introducing an iterative framework for timetable, rolling stock, and crew rescheduling. We show that a satisfactory, overall feasible solution can usually be found in only a few iterations. This suggests that the approaches for rescheduling individual resources can be combined and applied in practice during a disruption.

In the iterative framework, we use earlier published models and algorithms on (macroscopically) adjusting the timetable, rescheduling the rolling stock, and rescheduling crew schedules. The framework first computes a new timetable. Then, it reschedules the rolling stock, covering as many trips in the timetable as possible. Trips that cannot be covered by rolling stock are then cancelled in the timetable. Finally, the crew duties are rescheduled. Again, the objective is to cover as many trips from the timetable as possible. If some trips cannot be covered by crew, these trips are cancelled, and another iteration of the framework is necessary. Otherwise, if all trips are covered by crew, the algorithm terminates. We emphasize that our framework is very generic: Instead of the particular models and algorithms we use, other methods can be used in the framework as well, as long as they solve a similar problem.

We demonstrate the effectiveness of the iterative approach on real-world instances from Netherlands Railways (Nederlandse Spoorwegen, or NS). We consider 976 instances in total. In half of them, one of the tracks between two stations is blocked for a certain period of time. Then, only limited train traffic is possible between these stations. In the other half, all tracks between two stations are blocked and no train traffic is possible at all. The most important objective is to minimize the total duration of the cancelled train services.

The contribution of this paper is threefold. Firstly, we introduce an iterative framework to reschedule the timetable, rolling stock, and crew. This all-in-one framework leads to an overall feasible solution for all resources. Secondly, we show that the algorithm converges to a satisfactory solution for all considered real-world instances in a few iterations. This shows that an integrated approach is not required to obtain solutions that are overall feasible. Thirdly, we show that the framework and the underlying algorithms that we use are able to solve practical problems and can be of great benefit to railway operators. In this way, we hope to reduce the earlier mentioned gap between theory and practice.

The remainder of the paper is structured as follows. Section 2.2 reviews the relevant literature. Section 2.3 contains a description of the iterative framework. This section includes a short description of the algorithms we use to reschedule the individual resources. In Section 2.4, we present results on 976 disruptions on the railway network in the Netherlands. Finally, we finish the paper with some concluding remarks in Section 2.5.

2.2 – LITERATURE REVIEW

A disruption usually causes the timetable, the rolling stock schedule, and the crew schedule to be infeasible. The timetable and rolling stock schedule might contain trips that make use of infrastructure that is temporarily unavailable. These trips cannot be operated, which might prevent some crew members to perform all tasks in their duties. As a result, the resource schedules need to be adjusted. In current practice, this is mostly done manually. First, often with the help of contingency plans, the timetable is rescheduled. Then, with the new timetable as input, the rolling stock and crew tasks are rescheduled manually, one by one. This is a time consuming process, so decision support tools are most welcome.

Most of the scientific literature on railway disruption management focuses on rescheduling only one of the three resources. In this section, we will briefly review the literature on rescheduling the timetable, the rolling stock, and the crew. Fore a more in depth review we refer to Cacchiani et al. (2014).

The literature on timetable rescheduling can be classified in two parts: Macroscopic and microscopic timetable rescheduling. Macroscopic approaches to timetabling model the infrastructure on a high level of abstraction and usually deal with larger disruptions. For example, certain tracks might be unavailable for a couple of hours. Amongst others, Louwerse and Huisman (2014), and Veelenturf et al. (2015) have recently developed a macroscopic model for timetable rescheduling and have performed tests on the Dutch railway network. Zhan et al. (2015) developed a different macroscopic model and tested it on the Chinese railway network.

In contrast, microscopic models consider the railway infrastructure with a high level of detail. By doing so, the propagation of delays can be modelled with high accuracy. These models are usually applied to resolve smaller disturbances, e.g., few delays of up to half an hour. We refer to D'Ariano et al. (2007) and Corman et al. (2011) for examples of microscopic approaches to timetable rescheduling tested on the Dutch railway system and to Lamorgese and Mannino (2015) for microscopic rescheduling cases tested and implemented on the Italian and Norwegian railway network, respectively.

There are multiple papers with a focus on rescheduling the rolling stock. For instance, Nielsen et al. (2012) adjusted the Composition Model from Fioole et al. (2006) and applied it in a disruption management setting. In this model, the rolling stock rescheduling problem is formulated as a multi-commodity flow model. Here, the nodes correspond to stations and the arcs represent the trips between stations, or waiting inside stations. Furthermore, there is also a composition graph describing the feasible transitions of compositions in the stations. In the composition graph we have nodes representing trips for which rolling stock is required and the arcs represent possible changes in the rolling stock composition between these trips. Haahr et al. (2014) developed a unit based model for a similar problem, where a specific path is created for each unit separately. The model is then solved by means of column generation. The performance of these models is compared in Haahr et al. (2015c) (Chapter 3) on both the Dutch and the Danish railway network.

The third resource is the crew. Multiple researchers have investigated crew rescheduling. Rezanova and Ryan (2009) model the crew rescheduling as a Set Partitioning Problem and solve it by column generation. In a similar fashion, Potthoff et al. (2010) solve a Set Covering Problem by column generation and Lagrangian relaxation. This latter approach is extended by Veelenturf et al. (2012) with the possibility of retiming some of the tasks. Using a completely different method, Abbink et al. (2009) solve the crew rescheduling problem by means of an agent based system. Here, agents correspond to crew members and can swap parts of their duties.

All these papers show that models and algorithms can be used as decision support tools for rescheduling one resource individually. However, it has never been tested whether these individual rescheduling algorithms can be combined and lead to a solution that is overall feasible. If, for instance, no train driver can be found for a particular trip, it means that this specific trip cannot be executed. As a result, the timetable and rolling stock schedule become infeasible, and need to be rescheduled again. In the next section we propose an iterative framework that copes with these interactions.

There are few papers that investigate the integration of all or at least two of the rescheduling steps. However, these papers focus mainly on small or less complex railway networks. Examples are Walker et al. (2005), who integrate timetable and crew rescheduling, and Cadarso et al. (2013) and Cadarso et al. (2015), who integrate timetable and rolling stock rescheduling. Cadarso et al. (2013) also explicitly consider the effect of the rescheduling measures on the passenger demand and on the required seat capacity.

2.3 – FRAMEWORK

In this section, the iterative framework for real-time railway rescheduling is introduced. Furthermore, we describe the interactions between the different modules in the framework and we discuss the modules individually. Note that the modules that we use have been developed with a sequential approach in mind. Our framework performs this sequential approach iteratively.

As a consequence, the models and algorithms do not need to be adjusted or reconfigured, but they can be used without any modifications.

2.3.1 – Framework

The real-time railway rescheduling framework can be used for disruptions which lead to a temporary blockage of one or more tracks. In this paper, we assume that the duration of the disruption is known and fixed. The process can, however, be repeated as soon as there is new information available about the duration. In other words, the process can be embedded in a rolling horizon algorithm to handle the uncertainty regarding disruptions. A similar approach has been suggested by Nielsen et al. (2012) for rolling stock rescheduling. Alternatively, the uncertainty about the duration can be incorporated in the individual modules comprising the framework. As an example, Veelenturf et al. (2014) develop a quasi-robust crew rescheduling algorithm that takes into account the uncertainty about the duration of the disruption.

Due to the blockage of (some of) the tracks, the timetable becomes infeasible and needs to be rescheduled. Furthermore, the crew and rolling stock schedules need to be adjusted as well. A schematic overview of the framework can be found in Figure 2.1. The framework starts by

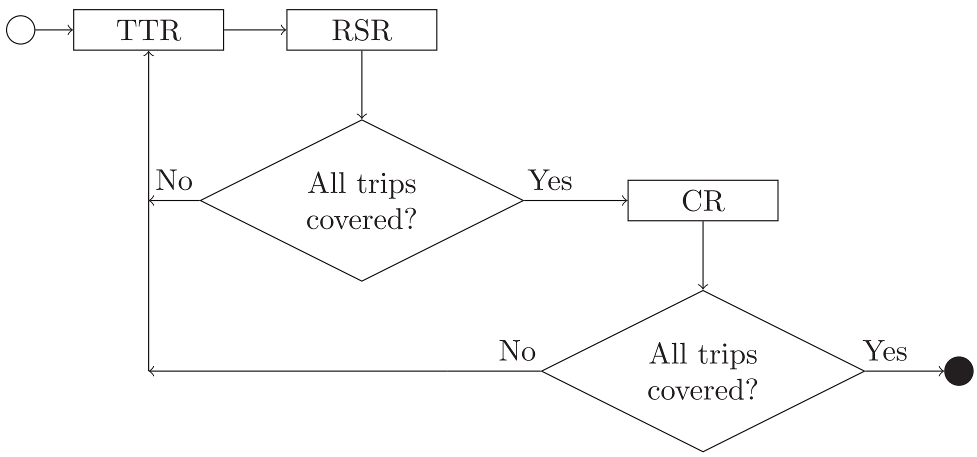


Figure 2.1: Overview of the iterative framework.

rescheduling the timetable macroscopically (TTR). The timetable should be adjusted by means of delaying or cancelling train services. The objective when rescheduling the timetable is to find a balance between cancelling as little train services as possible and minimizing the delay introduced in the timetable. Output of this module is a disposition timetable that is feasible with respect to the reduced infrastructure capacity.

The new disposition timetable is given as input to the module responsible for rescheduling the rolling stock (RSR). Cancelled train services can result in the original rolling stock circulation being infeasible. The goal when rescheduling the rolling stock is to assign a rolling stock composition (a set of combined rolling stock units) to as many trips in the disposition timetable as possible. Here, a trip is a part of a train service between two stations where the rolling stock composition can be changed.

It might be impossible to cover all trips from the disposition timetable with rolling stock. In that case, the timetable should be rescheduled a second time, preferably in such a way that rolling stock is available for all trips in the new timetable. One obvious solution is to cancel all trips in the timetable without a rolling stock composition assigned to it, but one could also consider more elaborate approaches. This process is repeated until a timetable is obtained for which rolling stock can be assigned to all trips.

Due to cancelled or delayed train services, the original crew schedule might be infeasible as well. The third module of the framework is responsible for rescheduling the crew (CR). This is done by appointing a new duty to every crew member. A duty is a list of tasks to be performed by a single crew member. A task corresponds to performing work (e.g., as a driver or as a conductor) on a certain trip. The most important constraints in the crew rescheduling part are the crew regulation rules (e.g., the presence of a meal break and a maximal working duration).

There might be tasks that cannot be assigned to any crew member. As a consequence, the corresponding trip cannot be performed. In that case, the timetable should be adapted in such a way that all trips can be covered with crew. After such an adjustment of the timetable, also the rolling stock might have to be rescheduled again. To prevent several iterations in the loop being necessary, the objective of the crew rescheduling approach is to assign crew to as many tasks as possible. A summary of the iterative framework is shown in Algorithm 1. It needs a

Algorithm 1: Iterative framework

1: **Input**: Characteristics of the disruption
2: Reschedule the macroscopic timetable (TTR).
3: Use output of TTR to reschedule the rolling stock (RSR).
4: **if** Not all trips are covered by rolling stock **then**
5:     Timetable needs adjustments; go back to Line 2.
6: **end if**
7: Use output of TTR and RSR to reschedule the crew (CR).
8: **if** Not all trips are covered by crew **then**
9:     Timetable needs adjustments; go back to Line 2.
10: **else**
11:     Overall feasible schedule found.
12: **end if**

timetable, a rolling stock and a crew rescheduling approach. In Sections 2.3.3-2.3.5 we briefly describe the timetable, rolling stock and crew rescheduling algorithms we apply in this research.

However, note that one can replace the particular approach we have chosen for a resource by any other rescheduling algorithm for that resource.

As is common in practice, we reschedule first the rolling stock and then the crew. One reason is that drivers cannot run all rolling stock types and that the number of conductors required for a trip depends on the length of the rolling stock composition. However, if these dependencies are discarded, it might be possible to reschedule the crew first, and then the rolling stock. In that case, Lines 3-5 and Lines 7-9 are interchanged in the algorithm above. We also test this variant of the framework. The results are presented in Section 2.4.3.2.

2.3.2 – Interactions

There are several interactions between the different modules comprising the iterative framework. These interactions, indicated by arrows in Figure 2.1, are discussed in this section in more detail.

First, at the time a disruption occurs, all modules require information on the current state of the railway system. This includes all events that have taken place up to this time. These events can no longer be changed. For the rolling stock and crew rescheduling modules, this information furthermore includes the original rolling stock and crew schedule. The rolling stock rescheduling module requires the passenger demand for each trip, such that enough capacity can be provided.

The timetable rescheduling module requires more information on the current state. First, it requires the timetable as it has ran up to the current time, in order to determine the location of all trains. Second, it needs the time the disruption takes place (the current time), because trains that have already departed cannot be cancelled any more. The timetable rescheduling approach we use also requires the duration of the disruption as input. Finally, the location of the blockage and the specific tracks that are blocked are required as input.

Another interaction is the exchange of information between the timetable rescheduling module and the rolling stock and crew rescheduling modules. The rolling stock and crew rescheduling modules need to know the newly constructed disposition timetable. This timetable describes for all non-cancelled train services all departure and arrival times at the stations. The interaction back from the rolling stock and crew rescheduling module to the timetable rescheduling module consists of the trips for which no rolling stock or crew can be found.

The last interaction is an information stream between the rolling stock and the crew rescheduling modules. The rolling stock schedule influences the transfer time for crew members. For example, if the next trip of a crew member uses the same rolling stock, then the crew does not have to walk to another train. Furthermore, not every crew member is allowed to run all types of rolling stock. We decided to make this interaction one-way: When rescheduling the rolling stock, we do not keep track of which crew member is assigned to the trip. Therefore, the crew rescheduling module should take into account which type of rolling stock is assigned to a certain trip.

2.3.3 – Timetable Rescheduling

We use two different approaches for rescheduling the timetable in the iterative framework. The first time the timetable is rescheduled, we use a sophisticated approach, which is based on Veelenturf et al. (2015). This approach considers the reduced infrastructure capacity caused by the disruption. Thereafter, if the timetable must be adjusted because of a lack of rolling stock or crew, we use a greedy approach. This combination of the sophisticated and greedy approach ensures that the number of iterations is finite. Furthermore, it reduces computation time. We now first explain the sophisticated approach and then the greedy approach.

Sophisticated approach

If the timetable is rescheduled for the first time, we use the approach suggested by Veelenturf et al. (2015). Here, the macroscopic timetabling problem is modelled using an event activity network. The events in this network represent the departures and arrivals of train services and the activities represent the minimal times (e.g. running, dwell and headway times) which should be scheduled between two events. The infrastructure capacity is modelled as the number of available tracks, which means that the precise locations of switches and signals are neglected. An important feature of this model is that it also takes the rolling stock capacity into account by keeping track of the number of rolling stock compositions entering and leaving a station. For each train service, it checks whether a rolling stock composition is available. If no rolling stock composition is available, that particular train service is cancelled. Detailed information about rolling stock types and coupling and uncoupling options are not considered. This means that the rolling stock capacity is measured on the level of compositions instead of on the level of units. This leads to a high probability that a feasible rolling stock schedule can be found for the disposition timetable, in which all trips are covered by rolling stock.

The model is formulated as a MIP and then solved using a general purpose solver (e.g., Cplex). The aim of the approach is to cancel and delay as few train services as possible by considering the reduced capacity and by deciding which train services should be delayed and/or cancelled. The order of train services on tracks can be switched and rolling stock turnings can be adapted.

Greedy approach

The greedy approach, which is used in case the rolling stock or crew rescheduling module was unable to cover all trips with rolling stock or crew, respectively, is quite basic. It cancels all trips that are not covered by rolling stock or crew. If the rolling stock module was unable to cover certain trips, this means that all other trips are included in a feasible rolling stock circulation. If we cancel the tasks that could not be covered, then, by construction, we obtain a timetable that is feasibly covered by rolling stock. Similarly, if the timetable algorithm was executed because some trips could not be covered by the crew, and we cancel these trips, we obtain a timetable in which all tasks are covered by crew. If, thereafter, the rolling stock module is able to cover all remaining trips, we can skip the subsequent crew rescheduling approach and we are done. In this way, we might possibly save one step in the next iteration.

Our choice for the greedy approach also implies that the iterative algorithm always terminates. In every iteration, except for the first one, it holds that another iteration is only performed if at least one trip is not covered by rolling stock in the rolling stock rescheduling step and, after cancelling this trip with the greedy approach, another trip is not covered by crew in the crew rescheduling step. Otherwise, the framework terminates with a globally feasible solution. So, in every iteration, at least two trips are cancelled. As a consequence, after $\frac{\#\text{trips}}{2}$ iterations all trips are cancelled, and this is a feasible solution as well. However, in most cases a feasible solution is obtained already after a few iterations, as will be shown in our computational results.

2.3.4 – Rolling Stock Rescheduling

The rolling stock rescheduling module implements the approach of Nielsen et al. (2012). The rolling stock rescheduling problem is formulated as a multi-commodity flow model. The aim is to appoint rolling stock compositions to trips and make sure that there is enough capacity for all passenger demand. The objective is to minimize the number of non-covered trips and the deviation from the original plan. Decision variables in the model indicate which composition is assigned to each trip. Furthermore, the model contains decision variables to indicate the rolling stock composition changes taking place between two consecutive trips. Not all composition changes are allowed. For example, for a composition consisting of three different units, it is impossible to uncouple only the middle unit from the composition. The model is solved using a general purpose solver (e.g., Cplex).

The number of available rolling stock units is given as input to the module. In order to get some flexibility, the number of rolling stock units which need to be parked at each station during the night is not fixed. However, it is heavily penalized if at a station less units are available at the end of the day in the new schedule. Such end-of-day unbalances require additional deadheading trips during the night. Similarly, additional shunting movements are penalized. Every new shunting movement must be communicated to local shunting crew and must then be scheduled in between other shunting work. We want to minimize the additional work for local shunting crew, and therefore penalize any new shunting movements.

2.3.5 – Crew Rescheduling

The approach of Veelenturf et al. (2012) is used for rescheduling the crew. This approach is based on Potthoff et al. (2010) and combines column generation with Lagrangian relaxation. It assigns new duties to crew members such that as many tasks as possible are covered by the duties of the crew members. Furthermore, this approach allows to delay certain tasks by a few minutes in order to reduce the number of non-covered tasks. However, in this paper we make no use of the possibility to delay certain tasks.

The crew rescheduling problem is formulated as a Set Covering Problem. Here, the duty of each crew member must be replaced by a new duty. The model contains decision variables that indicate which replacement duty is assigned to each crew member. Because the original duties can be replaced by many new duties, column generation is applied to generate promising replacement duties. Besides the main objective of covering as many tasks as possible, another objective is to have that the new duties deviate as little as possible from the original duties. Every deviation in the duties should be communicated to the crew members. This takes time and could lead to errors in practice.

In order to speed up the solution process, only a subset of the crew members is considered. In particular, the crew members whose duty became infeasible due to the disruption and the crew members in the neighbourhood of the disruption are included in the problem. The duties of the other crew members are fixed. In an iterative way, other neighbourhoods are explored as long as non-covered tasks are left.

2.4 – COMPUTATIONAL EXPERIMENTS

In this section, we describe the computational experiments that we have executed to assess the performance of the iterative framework. We first describe the cases that we have considered and then discuss the results we have obtained. The iterative framework has been implemented in Java using Eclipse Kepler. All computational tests are performed on a desktop with an Intel Quad Core i7 processor and 4GB of RAM. We used Cplex 12.6 as solver for the MIP models.

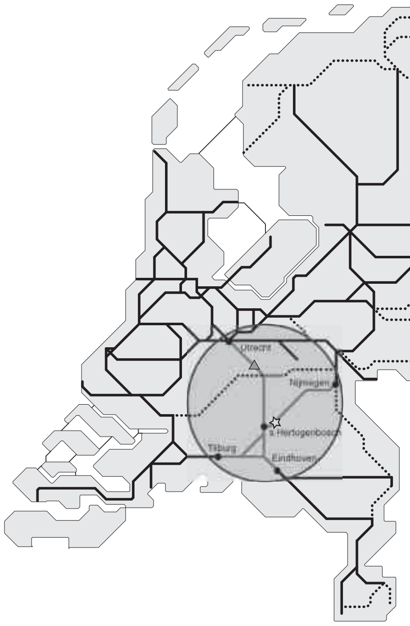


Figure 2.2: Map of the railway network in the Netherlands (2012).

2.4.1 – Case description

We have used the Dutch railway system to test our iterative framework. We have obtained the timetable, rolling stock data, and the crew schedule for a specific day in June 2012 from Netherlands Railways.

In Figure 2.2, a picture of the Dutch railway network is shown. The solid lines represent the railway network that is operated by Netherlands Railways. The dotted lines are operated by other railway operators. In general, the timetable is half-hourly periodic, with some exceptions in the rural areas. This means that the majority of long distance and regional train services run once every thirty minutes.

For the timetabling stage, we use only part of the railway network and require that all train services run as planned outside this region. This approach is in line with current practice and follows the literature on this topic, see, e.g., Veelenturf et al. (2015) and Louwerse and Huisman (2014). The region that we consider in the timetabling phase is depicted by the circle in the figure and in more detail in Figure 2.3. In the latter figure also the number of tracks within a station and the number of tracks between stations are displayed. In this region, 116 train departures from stations depicted in Figure 2.3 are scheduled per hour in the timetable.

We consider disruptions at two different locations: either between 's-Hertogenbosch (Ht) and Nijmegen (Nm) (indicated by the star in Figure 2.2) or between 's-Hertogenbosch and Utrecht (Ut) (indicated by the triangle). The dashed edges in Figure 2.3 also indicate the locations of the two disruptions. On both locations, there is a double track between the stations. We consider both a complete and a partial blockage. In a complete blockage, both

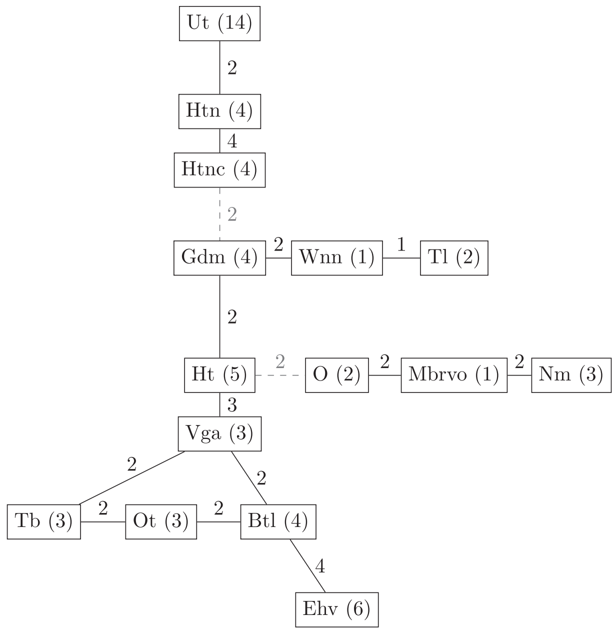


Figure 2.3: Graph of the railway network taken into account when rescheduling the timetable. Rectangles are the nodes and correspond to stations or important junctions. For each station, the number of tracks within the station is specified. Numbers at the edges indicate how many tracks connect two stations. The dashed edges represent the locations of the disruptions.

tracks are blocked and it is impossible to operate any train services between these stations. In a partial blockage, only one of the tracks is blocked, which allows train services to cross the disrupted region.

In the railway network considered when rescheduling the timetable, there are in total 15 nodes (stations or important junctions), with the number of tracks in the nodes varying from 1 until 14. Only stations or junctions are considered where ordering decisions between trains travelling in different directions are necessary. With ordering decisions between trains we mean the decision which train utilizes the specific track before the other. As a consequence, the considered stations and junctions are the ones to influence the macroscopic routing options of the train services. There are in total 5 stations (Ut, Gdm, Ht, Ehv and Nm) that have a shunting yard. We assume that all shunting yards have a sufficient number of tracks to store rolling stock units.

To analyse how the performance of our approach depends on the characteristics of the disruption, we consider different start times and different durations of the disruption. The start time of the duration varies between 7:00 AM and 5:00 PM in steps of ten minutes, giving us 61 different start times. The duration of the disruption is either 60, 80, 100, or 120 minutes. Recall that we assume that the duration of the disruption is known directly at the start of the disruption.

For the rolling stock and crew rescheduling phase, we reschedule the complete railway network. We no longer focus on a specific part of the network alone. Furthermore, for the rolling stock rescheduling phase, note that in the Netherlands we distinguish between long distance train services and regional train services. A regional train service is scheduled to stop at every station, while a long distance train service only dwells at the larger stations. The rolling stock units available for long distance train services are different from those available for regional train services. For example, rolling stock units for regional train services accelerate faster. In our experiments, it is not allowed to use rolling stock meant for long distance train services for a regional train service or vice versa. This allows us to decompose the rolling stock rescheduling problem into two independent problems: one rolling stock rescheduling problem for the long distance train services and one for the regional train services.

In the crew rescheduling step we assume that drivers and conductors work in pairs. As a consequence, by rescheduling the drivers we have immediately rescheduled the conductors.

2.4.2 – Implementation

The models discussed in Sections 2.3.3-2.3.5 require certain settings and input data. In this section, we discuss the details of the actual implementation of the iterative framework. The main objective in our experiments is to minimize the total duration of cancelled train services. Therefore, in each of the three approaches, this will be the individual main objective. For the rolling stock and crew rescheduling this is done indirectly by focusing on covering as many trips as possible with rolling stock and crew. If no rolling stock or crew is assigned to a certain trip, this trip will be cancelled by the timetable rescheduling. The penalties for not assigning rolling stock or crew to a certain trip depend on the duration of the trip. The values of the other penalties and parameters are set at the values proposed in the original publications on these algorithms. Note that the algorithms have been developed and tested in close collaboration with NS.

2.4.2.1 – Settings timetable rescheduling

The first time the timetable is rescheduled, we apply the approach developed by Veelenturf et al. (2015). Each next iteration, the timetable rescheduling module is called because either the rolling stock or the crew rescheduling was not able to cover all trips. For these cases we implemented the greedy timetable rescheduling approach which cancels all trips for which no rolling stock or crew was found.

The timetable rescheduling approach of Veelenturf et al. (2015) requires the original scheduled timetable as input. Every scheduled trip is represented by a corresponding departure and arrival event. Such an event contains the time it takes place, the corresponding station where it takes place, and the scheduled track on which it takes place.

Next to the timetable, a description of the railway network is necessary. Furthermore, the model requires parameters specifying the regulations between train services. Within the stations a headway time of 2 minutes is considered in between two consecutive train services assigned to the same track. This headway time of 2 minutes also applies to two consecutive train services running in the same direction assigned to the same track in between stations. The headway time of train services using a single track in opposite directions is 0 minutes.

Detailed settings for the rolling stock part of the timetable rescheduling module are necessary as input as well. When a train service ends, the rolling stock of that train service may be used by a starting train service at the same location. This is called a turning pattern. The minimum time between these two train services is set to 1 minute. If a rolling stock turning is chosen with a turn around time longer than 10 minutes and there is a shunting yard available, the rolling stock is shunted away. In such a situation, 5 minutes after the train service has ended, the rolling stock is shunted away, and 5 minutes before the new train service starts, the rolling stock is considered to be back at a station track.

In the objective function, a single cancelled minute is penalized by 50,000 since minimizing cancellations is the main objective. For each minute of arrival delay, a penalty of 1 applies. Furthermore, an event may be delayed by at most 8 minutes. This ensures that train services are not endlessly delayed in order to prevent cancellations. Note that we only take direct train delays into account. Longer passenger delays, due to missed connections for example, are not considered.

Another penalty is set upon deviating from a preferred turning pattern. For example, turning patterns of the same series are preferred. A list of preferred turning patterns is given as input and using a different turning pattern is penalized by 10.

Finally, the model requires as input a duration that specifies from what time onwards the timetable should be equal to the original timetable. We set this duration to 60 minutes. This means that from 60 minutes after the duration has ended onwards, all trains must be operated as planned again.

All penalty values for the sophisticated timetable rescheduling approach are summarized in Table 2.1.

Description	Penalty
Cancelled minute	50,000
One minute arrival delay	1
Turning pattern deviation	10

Table 2.1: Penalties timetable rescheduling approach.

2.4.2.2 – Settings Rolling Stock Rescheduling

The rolling stock rescheduling module based on Nielsen et al. (2012) uses many settings. In our experiments, there are four different rolling stock types available. There are two types for the long distance train services, namely one with 3 carriages and one with 4 carriages, and two types for the regional train services, also consisting of 3 and 4 carriages. Rolling stock units can be coupled to each other to form a rolling stock composition. In this way more capacity can be appointed to a trip. Only units of the same type are allowed to be coupled into a composition. The maximum length of a composition is 15 carriages.

At the start of the day, each station with a shunting yard contains a starting inventory of rolling stock units. This starting inventory denotes the number of available rolling stock units per rolling stock type at that station. Next to the starting inventory, the desired end-of-day inventory is needed per station. This is the amount of rolling stock units of a certain type that preferably is present at the end of the planning period at the corresponding station. We set a penalty of 100 per unit deviation from the desired end-of-day inventory.

Furthermore, an original rolling stock circulation is required for both the long distance and the regional train services. This original circulation contains a list of trips, where a trip is defined as a part of the train service between two stations where the composition may be changed. Every trip has the following characteristics: departure station, arrival station, departure time, arrival time, successor of the trip (turning pattern), and the originally appointed composition. The last information which the model requires is the start time of the disruption.

The largest penalty is set upon not covering a trip. A trip is not covered in the rolling stock rescheduling phase if no composition can be appointed to the trip. The penalty for not covering a trip equals 100,000 plus 1,000 times the duration of the trip in minutes. In this way we minimize the number of cancelled train services, and if we have to cancel a train service, then we prefer to cancel the train service with the shortest duration.

Next, we want to minimize the differences between the rescheduled rolling stock circulation and the original rolling stock circulation. First of all, the deviation in the number of carriages between the two circulations is penalized. A penalty of 1,000 is given per additional carriage on a trip and a penalty of 10,000 is given for every missing carriage on a trip in the rescheduled rolling stock circulation in comparison with the original rolling stock circulation. The penalty for missing a carriage is larger than the penalty for an additional carriage, because one missing carriage means a capacity reduction of approximately 100 seats for passengers on that trip. By doing so, we implicitly assume that the passenger demand used for the original circulation is still accurate. Another option would be to incorporate how passengers react when a disruption occurs. This issue is discussed, for example, by Van der Hurk et al. (2015). In that case, one can penalize seat capacity shortages directly when rescheduling the rolling stock circulations.

The final penalty is set upon deviating from the original shunting plan. If an originally planned shunting movement (either coupling or uncoupling) is cancelled, then there is a penalty of 100 for not performing that shunting movement. A penalty of 1000 is set upon newly added shunting movements. For these shunting movements, new shunting crew must be arranged, which could be a lot of work, so the penalty is larger than the penalty for cancelling a shunting movement.

All penalties of the rolling stock rescheduling approach are summarized in Table 2.2.

Description	Penalty
Cancelled trip	100,000 + 1,000 $\cdot$ duration of the trip
End-of-day balance deviation	100
Positive carriage deviation	1,000
Negative carriage deviation	10,000
Cancelled shunting movement	100
New shunting movement	1,000

Table 2.2: Penalties rolling stock rescheduling approach.

In order to be able to solve the rolling stock rescheduling problem in short time, we use a rolling horizon. Note that all trips before the start of the disruption are fixed, so our complete planning horizon is from the start of the disruption (t^s) up to the end of the day (t^∞). Solving the rolling stock rescheduling problem with the complete planning horizon can take long, so we split it by means of a dynamic planning horizon. The first part is from the start of the disruption up to time $t^s + \frac{t^\infty - t^s}{2} = \frac{t^s + t^\infty}{2}$, this is exactly halfway the complete planning horizon. The second part is from $\frac{t^s+t^\infty}{2}$ up to the end of the complete planning horizon. The solution of the first part is given as input to the second part, such that a feasible rolling stock circulation is found for the complete planning horizon.

Furthermore, we use a fixed computation time limit of 5 minutes per horizon. The optimal rolling stock circulation, with respect to our penalties, is found relatively fast. However, proving that this solution is optimal might take time. That is why we use a computation time limit of 5 minutes per horizon. In this time limit the optimal solution is almost always found, but not yet proven to be optimal.

2.4.2.3 – Settings Crew Rescheduling

The crew rescheduling approach of Veelenturf et al. (2012) uses penalties for not covering a task and for deviations in comparison to the original schedule. In the schedule, we distinguish between three types of tasks: tasks corresponding to operating a train service with a different start and end location, tasks corresponding to operating a train service with the same start and end location (operating the same rolling stock back and forth), and tasks which do not have to do anything with operating train services (e.g., training tasks). Since the aim is to operate as many train services as possible, the first two types of tasks get heavily penalized if they are not covered, while the third one is penalized less.

The penalty for not covering a task which is not related to operating train services, called local task, equals 250. For not covering a task related to operating train services, we test two settings. In the first setting, all tasks are treated equally and the penalty for not covering a task is equal to 20,000 plus 100 times the duration of the task. In the second setting we make a distinction between tasks in which the start and end location are the same (so called *AA-tasks*) and tasks in which the end location is different from the start location (so called *AB-tasks*). It can be argued that the rolling stock schedule remains feasible if an AA-task is cancelled. However, if an AB-task is cancelled, rolling stock rescheduling is definitely necessary. Therefore, in the second setting, we prefer not covering an AA-task over not covering an AB-task. For not covering an AA-task the penalty will be 3,000 plus 100 times the duration of the task. For not covering an AB-task the penalty remains 20,000 plus 100 times the duration of the task. It is expected that this second option leads to more cancellations in the crew rescheduling step but to less cancellations in the rolling stock rescheduling step.

The fixed costs for changing a duty equals 400 and for each new task in a duty a penalty of 50 applies. Each new transfer between tasks which was not present in any duty in the original crew schedule is penalized by 1. If the driver is directly sent home by a taxi (since no replacement duty is available which complies with the rules), a penalty of 3,000 is used.

The new duties may not take longer than the original duties and at maximum 5.5 hours may pass without a break of at least 30 minutes. The transfer time in between tasks on different rolling stock compositions equals 10 minutes.

The penalty values used for crew rescheduling are different than the ones used for rescheduling the rolling stock. The penalty values are all commonly used in literature. All three modules have the same overall objective: cancelling as little train services as possible.

The penalties used in the crew rescheduling approach are summarized in Table 2.3.

Description	Penalty
Cancelled task setting 1	20,000 + 100 · duration of the trip
Cancelled AA task setting 2	3,000 + 100 · duration of the trip
Cancel AB task setting 2	20,000 + 100 · duration of the trip
Cancel local task	250
Change duty	400
New task in duty	50
New transfer in duty	1
Send driver home by taxi	3,000

Table 2.3: Penalties crew rescheduling approach.

The approach of Veelenturf et al. (2012) also has an option to slightly delay tasks to have less tasks which cannot be covered. In our experiments we did not allow these delays, since then the crew rescheduling is interfering with the timetable rescheduling.

2.4.3 – Experiments

In this section, we present the results for complete and partial blockages on various settings of the framework. First, we test the general framework and discuss the associated results. Thereafter, the differences are presented between the general framework and the variant where the order of the rolling stock and crew rescheduling step are switched. Finally, we test whether having a lower penalty for not covering AA-tasks leads to less cancelled trips overall.

2.4.3.1 – Results of the general framework

First, the results of the general framework are presented. Table 2.4 gives an overview of the number of cancelled train services and the average total duration of the cancelled train services. These numbers include the train services that inevitably need to be cancelled because they are scheduled on the blocked tracks. Recall that for each location, duration, and type of disruption, we have 61 possible start times of the disruption. Each number in the table represents an average over these 61 instances. That leads to a total of 976 instances to test the framework upon, because there are two locations where a disruption occurs, four different durations, two different types (complete and partial), and 61 different start times.

Type	Duration (min)	Cancelled trips			Cancelled minutes		
		TTR	RSR	CR	TTR	RSR	CR
Ht-Ut Complete	60	14.33	0.20	0.75	284.46	5.08	19.30
	80	19.64	0.15	1.23	413.39	2.72	34.05
	100	25.36	0.13	1.72	526.71	2.28	47.38
	120	31.00	0.16	1.97	638.54	3.05	53.64
Ht-Ut Partial	60	5.33	0.03	0.89	88.51	0.56	21.39
	80	9.67	0.16	1.41	136.80	2.80	36.56
	100	9.34	0.05	0.77	167.95	0.69	22.26
	120	12.00	0.41	1.54	209.36	11.92	38.89
Ht-O Complete	60	8.00	0.03	0.84	124.80	0.43	27.44
	80	11.69	0.39	0.98	214.98	12.90	36.85
	100	13.00	0.13	1.39	226.15	2.20	48.66
	120	16.00	0.08	1.62	249.61	1.21	50.57
Ht-O Partial	60	4.00	0.02	0.61	59.51	0.21	18.54
	80	5.69	0.28	0.74	106.31	9.87	23.43
	100	6.66	0.00	0.93	120.71	0.00	32.15
	120	8.00	0.00	0.85	141.61	0.00	29.00
All cases		12.48	0.14	1.14	231.84	3.50	33.76

Table 2.4: Results of the General Framework. The first column denotes the location and the type of disruption and the second column the duration of the disruption. The third, fourth and fifth column denote the average number of cancelled train services in timetable, rolling stock, and crew rescheduling. The sixth, seventh, and eighth column show the average total duration of all the cancelled trips in timetable, rolling stock, and crew rescheduling in minutes.

First consider the number of cancelled train services. As expected, most of the train services are cancelled in the timetabling phase. Remember that in case of a complete blockage, all tracks are blocked. As a result, the module cancels most of the train services operated on those tracks. If these train services would not be cancelled, they would queue up in the railway system, causing knock-on effects over the whole country. As a result, we observe more cancellations for complete blockages than in case of partial blockages. Furthermore, as expected, the number of cancelled train services when rescheduling the timetable increases if the duration of the disruption increases. This holds for both complete and partial blockages.

Secondly, we observe that train services are cancelled in the rolling stock phase only rarely. This can be attributed to the timetabling algorithm, which increases the probability that a feasible rolling stock schedule exists that does not need to cancel any additional train services. Consequently, almost no train services are cancelled in the first iteration of the rolling stock rescheduling module.

This does not hold for the crew rescheduling stage. On average 1 to 2 train services are cancelled in that stage. In case of a complete blockage, the average number of tasks for which no crew can be found is increasing in the duration of the disruption. We do not see this pattern for the partial blockages. One explanation could be that the longer tracks are completely blocked, the more difficult it becomes to get the crew members home on time. In case of partial blockages, it is easier to get crew members at home since still some train services are operated. However, this does not mean that we can conclude that partial blockages lead to less or more cancelled train services due to lack of crew in general. On the one hand, partial blockages cause less gaps in duties by cancelled train services due to timetable rescheduling, but on the other hand due to the lack of gaps there is less buffer to adapt duties.

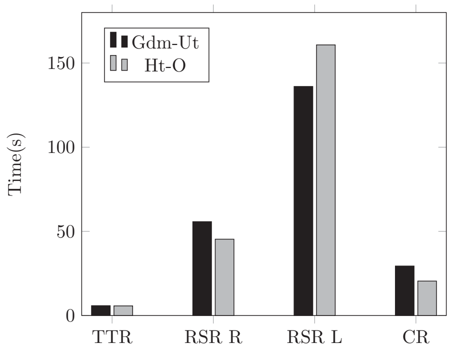
The numbers of cancelled minutes of train services follow a similar pattern as the number of cancelled train services for both complete and partial blockages.

Complete blockages		
Iteration	Cumulatively stopped after RSR	Cumulatively stopped after CR
1	-	193 (40%)
2	450 (92%)	482 (99%)
3	488 (100%)	
Partial blockages		
Iteration	Cumulatively stopped after RSR	Cumulatively stopped after CR
1	-	228 (47%)
2	475 (97%)	485 (99%)
3	486 (99%)	488 (100%)

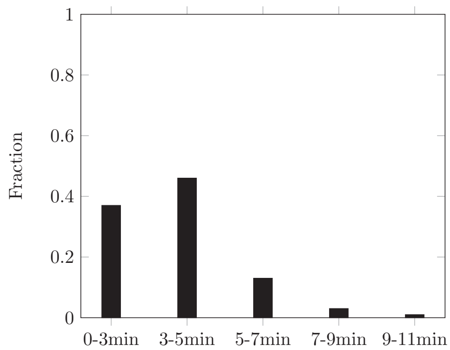
Table 2.5: Iterative behaviour of the general framework. The first column denotes the iteration. The second column gives the number of instances for which a feasible overall solution is found after rescheduling the rolling stock. The third column gives the number of instances for which a feasible overall solution is found after rescheduling the crew.

Next, we consider the iterative behaviour of the framework. In Table 2.5, we indicate the amount of instances that turn out to be feasible after each module for all iterations. As can be seen, for both the complete and partial blockages, at least 40% of the instances are solved in one iteration and 99% in two iterations. Furthermore, among the instances which stop in the second iteration, most are already stopped after the rolling stock rescheduling step. In other words, for more than 90% of the instances the crew rescheduling step is performed only once. All instances are solved in at most three rolling stock and three crew rescheduling steps, thus after three iterations.

The computation times are presented in Figures 2.4 and 2.5. In both figures, the left side gives an overview of the average computation time for each of the modules in the iterative framework. The computation time of a single instance of, for example, the crew rescheduling module is the total time it takes to reschedule the crew (so all iterations combined). On the right side a histogram of the total computation time is shown. It gives an overview of the percentage of instances that are solved within 0-3, 3-5, 5-7, 7-9, and 9-11 minutes. Note that the regional and long distance rolling stock rescheduling step can be solved in parallel. However, we have solved them sequentially. As a consequence, the total time it takes to solve an instance is the sum of the computation times for timetable rescheduling (TTR), rolling stock rescheduling for regional (RSR R) and long distance train services (RSR L), and crew rescheduling (CR). For both complete and partial disruptions more than 80% of the cases are solved within 5 minutes and less than 6% of the runs take more than 7 minutes. Note that the total computation time averaged over all instances is almost similar for complete and partial blockages: 3.8 and 3.9 minutes, respectively.

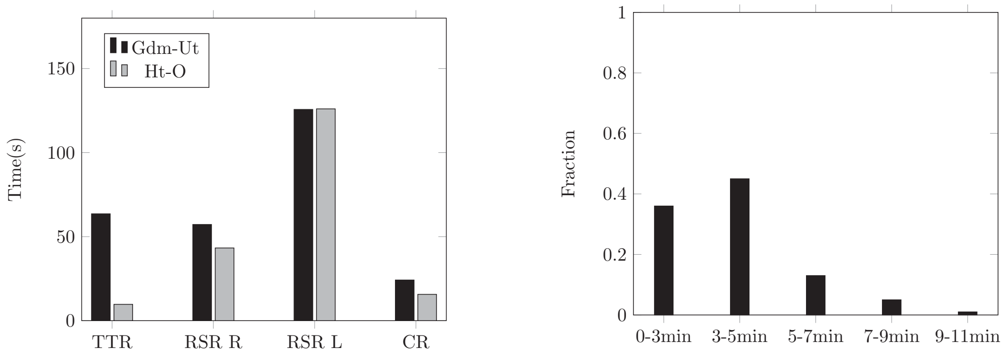


(a) Average computation times



(b) Computation time distribution

Figure 2.4: Computation times for complete blockages in the general framework. Here R abbreviates the regional train services and L the long distance train services.



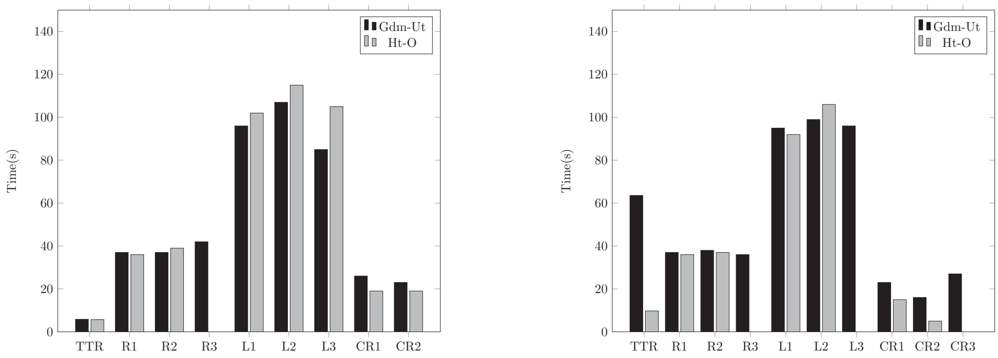
(a) Average computation times *(b) Computation time distribution*

Figure 2.5: Computation times for partial blockages in the general framework. Here R abbreviates the regional train services and L the long distance train services.

Figure 2.6 shows the average computation time per step in the framework for complete (left) and partial (right) blockages. As explained above, at most three iterations are needed within the framework. This explains why the rolling stock and crew rescheduling phases appear three times in the figure. Note that the sophisticated approach for timetabling is only applied in the first iteration. The computation time for this approach is reported in the figure. In the second and third iteration, the greedy approach for timetabling is applied. The greedy approach cancels the trips without rolling stock or crew. This can be done instantaneously. Thus, the computation times for the greedy approach all are 0 and are not reported in the figure. There is no third iteration necessary for crew rescheduling for any of the instances during a complete blockage, so we have left that one out of the figure.

Also, there was no third iteration required for any of the Ht-O instances with partial blockages and not for any of the regional train instances with a complete blockage between Ht-O. Note that for the second and third iterations, the average computation time is computed over the instances for which a second or third iteration was required, respectively. It can be seen that in case of a complete blockage, the timetable rescheduling step takes a couple of seconds, while in case of a partial blockage, it takes about half a minute. The average computation time spent in crew rescheduling is less than half a minute and is in case of a partial blockage lower than in case of a complete blockage. Rescheduling the rolling stock takes most of the time: The average time required is a couple of minutes and again less for instances with partial blockages than for instances with complete blockages.

Summarizing, the total average computation time is the same for partial and complete blockages. However, the division of the computation time over the rescheduling steps differs. Partial blockages require more time for the timetable rescheduling, but that is compensated by requiring less time for rescheduling the rolling stock and crew.



(a) Complete blockages *(b) Partial blockages*

Figure 2.6: Computation times for each of the steps in the general framework. Here R abbreviates the regional train services and L the long distance train services. $R1$ abbreviates then the first iteration for the regional train services, etc.

2.4.3.2 – Crew First Variant

As discussed in Section 2.3, the order of the rescheduling modules can be changed in the iterative framework. In this section, we consider the variant where we reschedule the crew before rescheduling the rolling stock in each iteration. In practice, this is often not possible. First, the number of conductors required for a certain trip depends on the number of carriages that are operated for the trip. This number of carriages might change in the rolling stock rescheduling step. Secondly, drivers have a license to operate a subset of all rolling stock types. Again, the rolling stock type for all trips is only known after rescheduling the rolling stock. However, in this section, we assume that the crew schedule can be generated independently of the rolling stock schedule. In this case we are able to reschedule the crew before the rolling stock. This special variant of our framework will be referred to as the Crew First Variant. The Crew First Variant is interesting, because some rolling stock constraints are taken into account in the timetable rescheduling step of Veelenturf et al. (2015). Consequently, less iterations might be required to find an overall feasible solution. Therefore, we have tested this setting. The results are presented in Tables 2.6 and 2.7 and Figures 2.7 and 2.9.

In Table 2.6 the number of cancelled train services and the average duration of the cancelled train services are presented. There is not much difference in terms of the number of cancellations and the duration of the cancelled train services in comparison with the general framework. However, if we look at the number of iterations in Table 2.7, we see that at most two rolling stock iterations are necessary now. Furthermore, the percentage of instances solved in one iteration is very large (about 90%). In the general framework, about 50 to 60 percent of the instances needed at least two rolling stock rescheduling steps (see Table 2.5). With the

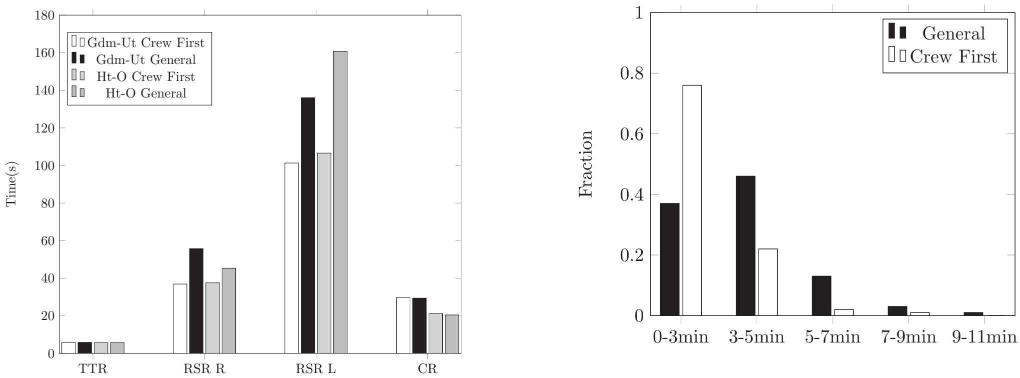
Type	Duration (min)	Cancelled trips TTR	RSR	CR	Cancelled minutes TTR	RSR	CR
Ht-Ut Complete	60	14.33	0.20	0.79	284.46	5.08	19.77
	80	19.64	0.10	1.23	413.39	1.69	34.05
	100	25.36	0.13	1.72	526.71	2.28	47.38
	120	31.00	0.12	1.97	638.54	2.02	53.64
Ht-Ut Partial	60	5.33	0.03	0.89	88.51	0.56	21.39
	80	9.67	0.16	1.41	136.80	2.80	36.56
	100	9.34	0.05	0.77	167.95	0.69	22.26
	120	12.0	0.33	1.71	209.36	9.00	43.02
Ht-O Complete	60	8.00	0.03	0.84	124.80	0.43	27.44
	80	11.69	0.39	1.00	214.98	12.89	37.25
	100	13.00	0.05	1.39	226.15	0.74	48.66
	120	16.00	0.08	1.62	249.61	1.21	50.57
Ht-O Partial	60	4.00	0.02	0.61	59.51	0.21	18.54
	80	5.69	0.26	0.79	106.31	9.31	24.85
	100	6.66	0.00	0.93	120.71	0.00	32.15
	120	8.00	0.00	0.85	141.61	0.00	29.00
All cases		12.48	0.12	1.16	231.84	3.06	34.16

Table 2.6: Results of the Crew First Variant. See Table 2.4 for the description of the columns.

Crew First Variant, only one percent of the instances need a second rolling stock rescheduling step. As the rolling stock rescheduling step is the most time consuming step, this leads to a decrease in the average computation time, as can be seen in Figures 2.7 and 2.9. Here, both the computation times of the General Framework and of the Crew First Variant are displayed. We note that the average computation time equals 2.8 and 3.2 minutes, respectively, for the instances with complete and partial blockages in the Crew First Variant. This is about one minute (about 25%) faster than in the general framework. The distribution of the computation times per step in the Crew First Variant is the same as in the general framework (see Figure 2.6), therefore we do not show such a figure again.

Complete blockages		
Iteration	Cumulatively stopped after CR	Cumulatively stopped after RSR
1	-	428 (88%)
2	481 (99%)	488 (100%)
Partial blockages		
Iteration	Cumulatively stopped after CR	Cumulatively stopped after RSR
1	-	451 (92%)
2	483 (99%)	486 (99%)
3	488 (100%)	

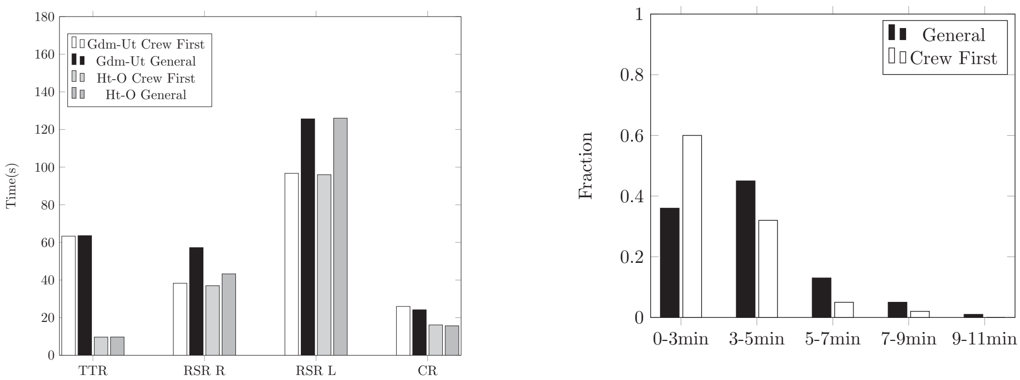
Table 2.7: Iterative behaviour of the Crew First Variant. See Table 2.5 for a description of the columns.



(a) Average computation times

(b) Computation time distribution

Figure 2.7: Computation times for complete blockages in the Crew First Variant. Here R abbreviates the regional train services and L the long distance train services.



(a) Average computation times

(b) Computation time distribution

Figure 2.9: Computation times for partial blockages in the Crew First Variant. Here R abbreviates the regional train services and L the long distance train services.

2.4.3.3 – Different Costs Variant

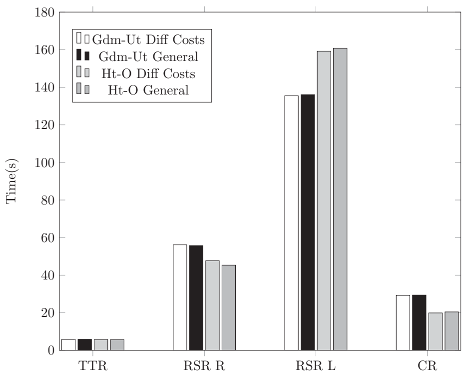
In the previous section, we demonstrated that computation times could be reduced by changing the order of the rolling stock and crew rescheduling steps. This might not be applicable in reality because it depends on assumptions which do not always hold in practice.

Another idea to save time for the rolling stock rescheduling phase is to consider rolling stock properties already in the crew rescheduling module. If a crew task with a different start and end location (AB-task) is cancelled, it causes a gap and leads to an infeasibility in a rolling stock duty. However, if a crew task with the same start and end location (AA-task) is cancelled, it is assumed that the gap in the rolling stock duty does not make the duty infeasible. During the crew rescheduling phase we could aim to prefer cancellations of AA-tasks over cancellations of AB-tasks, by having different penalties for not covering these tasks (as discussed in Section 2.4.2.3). This could result in less cancellations in the next rolling stock rescheduling step. We refer to this as the Different Costs Variant. In the Different Cost Variant, we first reschedule the rolling stock, and then the crew in each iteration. Hence, the costs are changed compared to the base case discussed in Section 2.4.3.1.

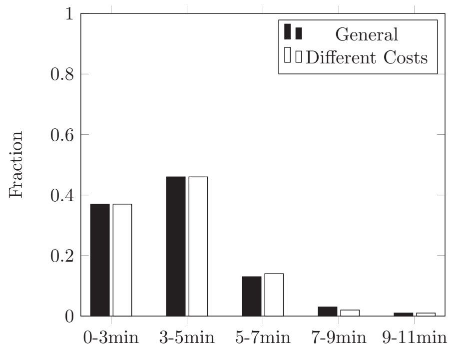
		Cancelled trips			Cancelled minutes		
Type	Duration (min)	TTR	RSR	CR	TTR	RSR	CR
Ht-Ut Complete	60	14.32	0.21	0.74	284.46	5.36	20.97
	80	19.64	0.16	1.21	413.39	3.00	35.26
	100	25.36	0.13	1.70	526.71	2.30	52.71
	120	31.00	0.20	1.97	638.54	3.62	56.84
Ht-Ut Partial	60	5.33	0.03	0.87	88.51	0.56	29.18
	80	9.67	0.16	1.36	136.80	2.80	42.82
	100	9.34	0.07	0.77	167.95	0.97	25.51
	120	12.00	0.41	1.61	209.36	11.92	49.66
Ht-O Complete	60	8.00	0.02	0.89	124.80	0.21	32.66
	80	11.69	0.39	1.02	214.98	12.90	40.12
	100	13.00	0.13	1.48	226.15	2.20	56.67
	120	16.00	0.07	1.69	249.61	0.95	57.97
Ht-O Partial	60	4.00	0.02	0.62	59.51	0.21	19.84
	80	5.69	0.28	0.80	106.31	9.87	30.79
	100	6.66	0.00	1.05	120.71	0.00	41.46
	120	8.00	0.00	0.97	141.61	0.00	39.90
All cases		12.48	0.14	1.17	231.48	3.55	39.52

Table 2.9: Results of the Different Costs Variant. See Table 2.4 for a description of the columns.

In Table 2.9, the results of the Different Costs Variant are presented. As expected, the number of cancelled train services and corresponding minutes in the crew rescheduling step

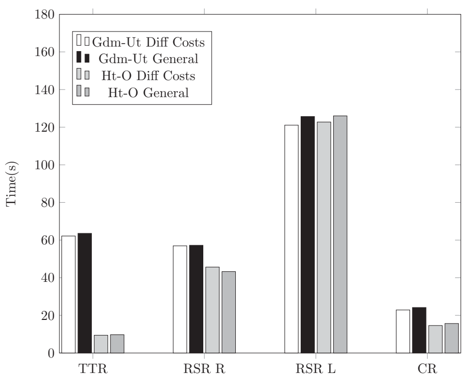


(a) *Average computation times*

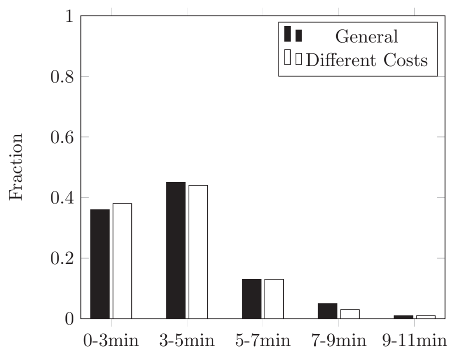


(b) *Computation time distribution*

Figure 2.10: Computation times for complete blockages in the Different Costs Variant. Here R abbreviates the regional train services and L the long distance train services.



(a) *Average computation times*



(b) *Computation time distribution*

Figure 2.11: Computation times for partial blockages in the Different Costs Variant. Here R abbreviates the regional train services and L the long distance train services.

have increased in comparison to the general framework. This is due to the fact that it is now cheaper to cancel AA-tasks. The percentage of cancelled AA-tasks with respect to all tasks cancelled due to lack of crew has now increased from 28% to 38%. However, this has not led to a decrease in the number of cancelled tasks due to lack of rolling stock. Instead, it has led to a very slight increase in the duration of the cancelled trips due to lack of rolling stock. Most likely this is caused by the fact that in total more crew tasks are cancelled.

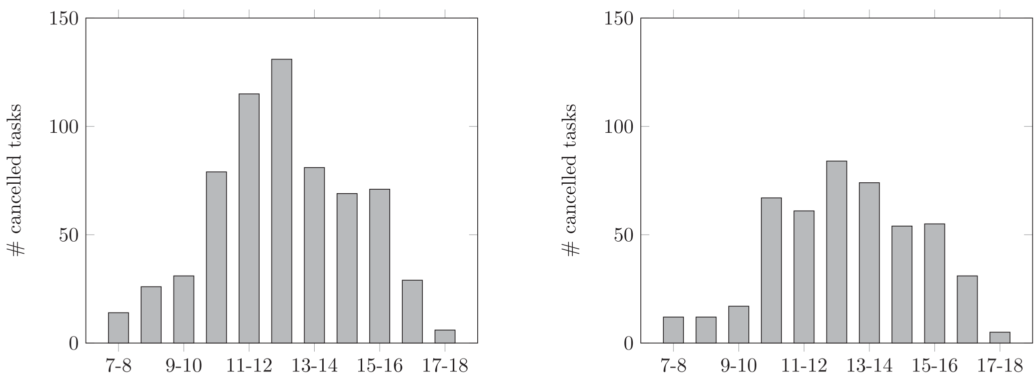
In terms of computation times, there are no significant differences as can be seen in Figure 2.10 and Figure 2.11.

2.4.4 – Practical considerations

Within Netherlands Railways, there is a lot of know-how about rescheduling during disruptions. These experiences in rescheduling result in several interesting findings, which we like to investigate now we have the relevant data.

2.4.4.1 – Influence of start time disruption on crew rescheduling

One of these interesting findings arising from practice is that if a disruption occurs between 12:00 and 14:00, many more train services get cancelled due to lack of crew than at any other time during the day. We decided to check whether this is true. Figure 2.12 shows the distribution of the number of cancelled train services due to lack of crew against the start time of the disruption. As can be seen, both for complete and partial blockages, the peak is between 11:00 and 14:00. So the start time of the disruption indeed has a large influence on the number of train services being cancelled by crew rescheduling. This is probably due to the fact that there are many duties that end around 14:00 at Netherlands Railways, which makes it difficult to get the involved crew members back on time at their end location. This finding could help with allocating and scheduling the reserve crew.



(a) Complete blockages *(b) Partial blockages*

Figure 2.12: Spread of cancelled train services due to lack of crew members

2.4.4.2 – Redundant iterative steps removed

A second finding is that if in the crew rescheduling step only AA-tasks get cancelled, we no longer have to reschedule the rolling stock again. The start and end location of an AA-task is the same, so cancelling an AA-task can probably be solved in the rolling stock circulation fairly easily. As a consequence, the rolling stock circulation should remain feasible. Our results demonstrate that this is indeed the case. If only AA-tasks get cancelled in the crew rescheduling module, then the next rolling stock rescheduling step is redundant.

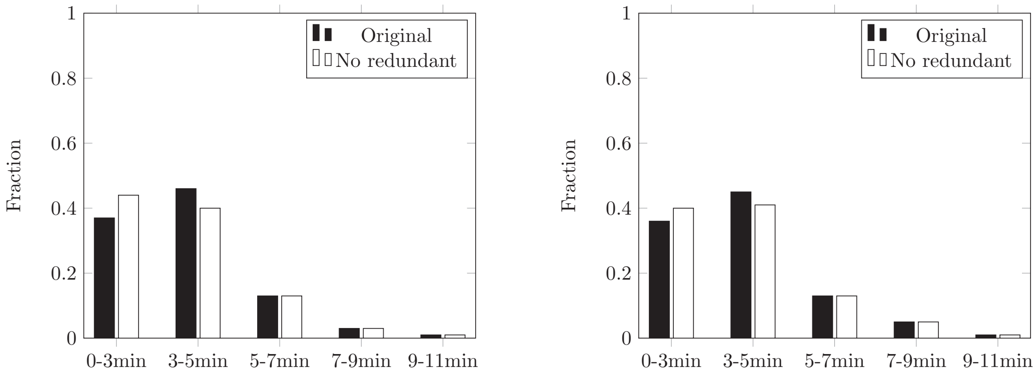
By not doing the redundant rolling stock step, we can save one iteration. As a result, in total we are able to save 42 iterations in case of a complete blockage, and 30 iterations in case of a partial blockage in the General Framework. No iterations can be saved by not doing a redundant rolling stock rescheduling step in the Crew First Variant, because there were no redundant rolling stock rescheduling steps. For the Different Costs Variant, we are able to save in total 61 iterations in case of a complete blockage and 74 in case of partial partial blockages.

Table 2.10 shows the average computation time required to solve the instances with and without the redundant rolling stock rescheduling step. As can be seen, on average around 10 seconds of computation time is saved for the instances with a full blockage and around 5 seconds for the partial blockages for both the General Framework and the Different Costs Variant.

Type	Module	General	Crew First	Different Costs
Complete	RSR L	290.59	207.87	285.48
No redundant	RSR R	96.42	74.47	95.90
	Total	448.41	344.71	442.16
Complete	RSR L	296.84	207.87	294.66
Redundant	RSR R	101.08	74.47	103.86
	Total	459.32	344.71	459.30
Partial	RSR L	250.52	192.69	241.93
No redundant	RSR R	96.40	75.27	92.21
	Total	460.06	383.06	443.21
Partial	RSR L	251.68	192.69	243.82
Redundant	RSR R	100.50	75.27	102.61
	Total	465.32	383.06	455.49

Table 2.10: Average computation times: the first column denotes whether instances with a complete or partial blockage are solved and whether the redundant rolling stock step is performed or not. The second column denotes for which part of the framework we present the results (RSR L, RSR R, or the total framework). The third, fourth, and fifth column denote the average computation time required to solve the instances for different variants of the framework.

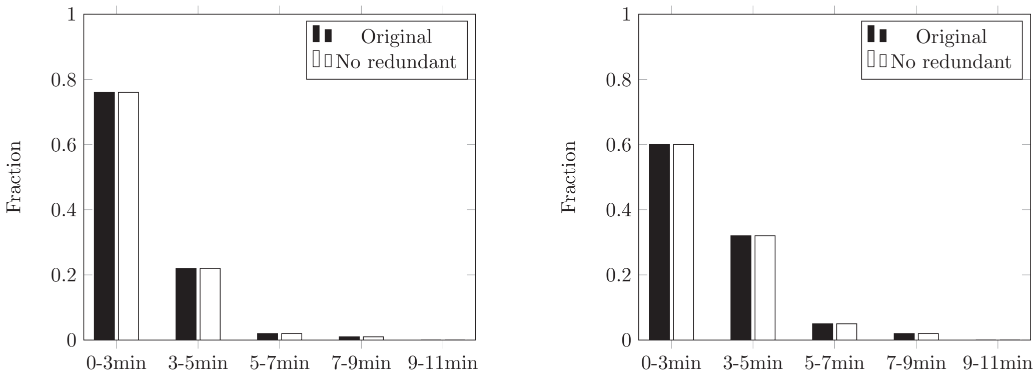
Figures 2.13, 2.14, and 2.15 present the computation time distribution for the General Framework, the Crew First Variant and the Different Costs Variant, respectively. Every figure presents both the distribution when redundant steps are performed and when not. For both the General Framework and the Different Costs Variant, around 5%-10% more instances can now be solved within 0-3 minutes.



(a) Complete blockages

(b) Partial blockages

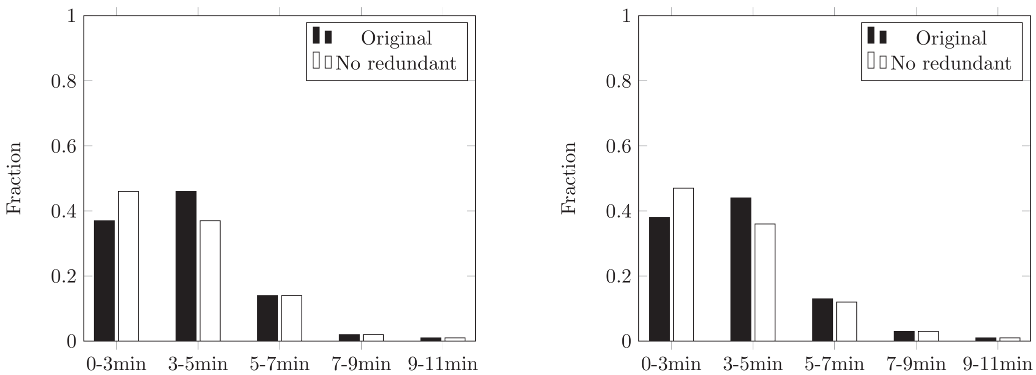
Figure 2.13: Total computation time distribution for the General Framework



(a) Complete blockages

(b) Partial blockages

Figure 2.14: Total computation time distribution for the Crew First Variant.



(a) Complete blockages

(b) Partial blockages

Figure 2.15: Total computation time distribution for the Different Costs Variant.

2.4.4.3 – Evaluation of contingency plans

We have described our iterative framework as a tool for real-time rescheduling. However, it can also be applied in a different setting. NS and Prorail are currently implementing the framework to test the quality of their contingency plans. These contingency plans describe what to do in case of a disruption, such as a blockage of one or more tracks. If a track is blocked, the contingency plan prescribes which train services to cancel in the timetable, and how to adjust the turnings in the rolling stock circulation. After manually inserting the timetable from the contingency plan as a disposition timetable, the rolling stock and crew are rescheduled by the framework. This allows to investigate whether a feasible rolling stock and crew schedule exist for the given contingency plan, given a certain start time and duration of the disruption.

If a feasible solution exists for a certain given percentage of possible start times and durations, the contingency plan is accepted. However, if it is not always possible to cover all tasks from the disposition timetable by rolling stock and crew, the contingency planner has a detailed look at the cause of these issues. For instance, it is possible that several tasks cannot be covered by rolling stock, because the total travel and turn around time in the contingency plan is larger than in the original timetable. In such a case, the contingency plan needs to be adjusted. In another situation, where, e.g., for a certain instance one task cannot be covered by a driver, it could be possible to cover this task by allowing a small violation of a labour rule. In such a case, the contingency plan will be accepted.

2.5 – CONCLUSION

Most studies on disruption management in passenger railways focus on the rescheduling of one resource (timetable, crew or rolling stock) schedule at the time, see Cacchiani et al. (2014). However, it was not yet investigated whether these approaches could be combined to find an overall feasible solution. Therefore, we presented an iterative framework considering all the resource schedules in this paper.

This framework is tested with existing models on a large number of disruption scenarios of Netherlands Railways. The experiments demonstrate that with this framework a railway operator is able to find a new timetable, rolling stock, and crew schedule in short time in case of track blockages. Furthermore, few trips tend to get cancelled in the rolling stock rescheduling and the crew rescheduling modules.

Our computational experiments show that the framework does not need many iterations between the different modules. The general framework and the different costs variant solves 40 (47)% of the full (partial) blockage instances after one iteration and already 99 (99)% after two iterations. This indicates that the chosen models perform well on an individual basis and do not come up with solutions which make it hard to reschedule the other resources.

The Crew First Variant performs even better by solving 88 (92)% of the full (partial) blockage instances after one iteration. However, this variant might not be applicable in practice, because there are crew specific rules depending on the rolling stock circulation.

Our framework is able to find a new feasible timetable, rolling stock circulation, and crew schedule after the occurrence of a disruption. The potential of this framework is significant. Railway operators can use it for real-time railway disruption management in practice. It provides a feasible schedule based on infrastructure, rolling stock, and crew constraints. The fact that this framework uses individual rescheduling modules for each resource allows railway operators to use their own preferred approaches for the rescheduling modules.

The interchangeability of the rescheduling modules makes the framework also useful for researchers. They can test whether their suggested approaches for rescheduling one resource type also performs well on a global scale where the other resources are considered as well.

We see several interesting directions for future research. Firstly, it would be interesting to compare our results to those obtained with a microscopic approach as a timetabling module. By using a microscopic representation of the infrastructure, the running and headway times can be computed more accurately. Our framework allows to replace the macroscopic timetabling module by a microscopic one, or by a model that incorporates both microscopic and macroscopic aspects. Secondly, it would be interesting to improve the feedback loop from the rolling stock and crew rescheduling modules to the timetabling module. By adjusting the solution of the timetabling module instead of cancelling the trips that cannot be covered by rolling stock or crew, solutions might be obtained that are overall of better quality.

3 – A Comparison of Two Exact Methods for Passenger Railway Rolling Stock (Re)Scheduling

This chapter considers the paper (Haahr et al. (2015c)). The corresponding paper is published in *Transportation Research Part E: Logistics and Transportation Review*.

Co-authors: J.T. Haahr, L.P. Veelenturf and L.G. Kroon

3.1 – INTRODUCTION

In passenger railway rolling stock (re)scheduling one of the main goals is to make efficient plans that can accommodate all passengers or, if that is not possible, that minimize the seat shortages. In the planning process, a railway operator tries to match the demand by first selecting an appropriate timetable followed by a matching rolling stock schedule, and finally, by appointing the drivers and conductors to operate the trains in the timetable. We focus in this research on the second stage where a rolling stock schedule must be found given the timetable and passenger demand. Not only the planning process, in which a long time ahead a rolling stock schedule must be determined, is considered, but also the real-time construction of rolling stock schedules due to disruptions.

If during operations an unexpected event causes the timetable, the rolling stock, and crew schedule to become infeasible, then these schedules need to be rescheduled. To ensure that the operations can be resumed quickly, new feasible schedules must be found promptly. The major difference between rolling stock scheduling in the planning phase and rolling stock rescheduling during disruptions is the time available to come up with a solution. Next to that, during the rescheduling phase there is less flexibility since the trains are already running, and choices made before the disruption occurred cannot be reversed. A natural consequence is that it may not be possible to assign rolling stock to all train services, i.e., some train services may need to be cancelled. Therefore in the rescheduling case it is highly undesirable, but considered feasible, to assign no rolling stock to some services.

There is a body of literature tackling the problem of assigning rolling stock to passenger train services. These papers focus either on the planning process or on the rescheduling process. The majority of the approaches existing in literature considers one specific network, and are not benchmarked against other approaches. In this paper we want to make a start with comparing different approaches, as well as comparing them on different networks. We hope to encourage other researchers to do the same.

We consider two rolling stock (re)scheduling approaches for self-propelled train units. The first approach is based on the algorithm introduced by Fioole et al. (2006). This approach makes use of a general purpose solver (CPLEX) to solve a Mixed Integer Linear Program (MILP) for rolling stock (re)scheduling. The second approach is a new approach, which extends the algorithm introduced by Haahr et al. (2014), and which involves path generation for individual rolling stock units. The method of Haahr et al. (2014) does not consider the order of rolling stock units within a train composition. However, this order is of significant importance to determine which units can be coupled/ uncoupled to/from a composition. For example, at certain stations only the unit at the rear of a composition can be decoupled. Therefore we introduce an extension which considers the order of rolling stock units within compositions. This new path based formulation requires adaptations to the column generation heuristic introduced by Haahr et al. (2014). Furthermore, in order to improve the performance in terms of computation time, a row generation variant of the algorithm is introduced as well. The advantage of a path based model in comparison to a flow based model (e.g. the model by Fioole et al. (2006)) is that unit specific constraints such as maintenance required by specific units can be included.

Although the MILPs of the two approaches are different, they are set up such that they solve exactly the same problem. Since both solution methods are exact solution methods, they will come to solutions with the same objective value. Note that the solutions themselves can differ if multiple solutions with the same objective value exist.

We benchmark both approaches on rolling stock scheduling and rescheduling instances of Netherlands Railways (NS) and the Copenhagen Suburban Railway Operator DSB S-tog. In the scheduling instances a timetable and the passenger demand are given. The methods must assign rolling stock compositions to every train service such that the seat coverage of passenger demand is satisfactory while minimizing operational costs. For the rescheduling instances, the original rolling stock schedule, an updated timetable, and passenger demand are given. As the demand will be input, modelling the changed passenger demand during a disruption is considered as out of the scope for this paper. In these rescheduling instances the main objective is to assign an appropriate rolling stock composition to as many train services as possible. However, it is no longer a hard constraint that all train services need a rolling stock composition assigned to them. The secondary objectives are to cover the passenger demand as well as possible and to deviate as little as possible from the original schedule. All deviations from the original schedule require additional shunting movements like couplings and uncouplings of train units. Unplanned or cancelled shunting movements require additional communication and coordination with shunting personnel; in some cases even additional shunting personnel must be arranged. Introducing deviations to the planned schedule is thus not preferred, especially since the available time is limited to communicate all changes to the involved crew members.

The remainder of this paper has the following structure. First, the contributions of this paper are discussed in Section 3.1.1. Thereafter, a literature overview of the existing literature is given in Section 3.2. The problem description and assumptions are discussed in Section 3.3. The mathematical formulations for the two solution approaches are discussed in Section 3.4. In Section 3.5 computational experiments are presented. The paper is concluded in Section 3.6.

3.1.1 – Contributions

This paper has several methodological and practical contributions. The methodological contribution is the introduction of a new rolling stock (re)scheduling approach by: *i)* extending an existing column generation formulation, and *ii)* introducing a new row generation method. The new mathematical formulation is a path based MILP formulation for the rolling stock (re)scheduling problem. In this formulation the order of units within compositions is taken into account. Finally, a row generation method is adopted for a significant speedup in runtime of the solution approach.

The practical contributions are: *i)* realistic tests on DSB S-tog and Netherlands Railways instances, *ii)* a comparison between different rolling stock (re)scheduling approaches, and *iii)* comparisons of different instances within different countries. Note that this is the first work that actually incorporates the order of units within compositions in test instances of DSB S-tog. The transition (i.e. composition) rules have been made in co-operation with DSB S-tog. A comparison is made between different rolling stock (re)scheduling approaches by testing them on the same data sets. These approaches are benchmarked on two different railway networks with cyclic timetables, namely a large train service network in the Netherlands and the suburban network in Copenhagen, Denmark.

3.2 – LITERATURE OVERVIEW

Two main categories of rolling stock (re)scheduling problems in passenger railways have been studied in literature. The first consists of assigning both carriages and locomotives to trips. Each carriage can be coupled individually and independently to a convoy (or composition), but at least one locomotive is required to pull the convoy. The other branch of research consists of assigning self-propelled train units that are not required to be pulled by a locomotive. These units consist of a fixed number of carriages and have their own traction engines. It is common that these train units can be coupled together to form larger train compositions.

The problem of assigning locomotives and/or carriages to trains can be applied to passenger trains (see e.g. Brucker et al. (2003), Cordeau et al. (2001), Cordeau et al. (2002), and Lingaya et al. (2002)), but also to freight trains (see e.g. Ahuja et al. (2005), Bouzaiene-Ayari et al. (2014), Fügenschuh et al. (2008), Rouillon et al. (2006), and Vaidyanathan et al. (2008)). The models used in most of these papers are based on a multi-commodity flow formulation, which are therefore referred to as flow-based models. In these mentioned papers, the approaches are tested only on a single network from one country and are not benchmarked against other approaches.

The problem of (re)scheduling self-propelled units is also considered in multiple publications (see e.g. Alfieri et al. (2006), Borndörfer et al. (2015), Cacchiani et al. (2010), Cadarso and Marín (2011), Cadarso and Marín (2014), Fioole et al. (2006), Haahr et al. (2014), Nielsen et al. (2012), and Peeters and Kroon (2008)). Most of these papers use flow-based approaches. The papers Cacchiani et al. (2010) and Haahr et al. (2014) assign paths (a list of subsequent trips) to individual units. Approaches assigning such paths to individual units are from now on referred to as path-based approaches. In the current path-based approaches the order of units within the composition are not considered. Also for the research on (re)scheduling self-propelled train units it holds that the approaches in the papers are only tested on a single network from one country and that they are not benchmarked against other approaches.

For more references, with respect to papers considering rolling stock (re)scheduling approaches, we refer the interested reader to Cacchiani et al. (2014) and Piu and Speranza (2014).

As explained before, in this paper we consider two rolling stock (re)scheduling approaches for self-propelled train units. The first approach is based on the algorithm introduced by Fioole et al. (2006). This approach makes use of a general purpose solver (CPLEX) to solve a MILP for rolling stock scheduling but could actually be used for rescheduling as well as demonstrated in Nielsen et al. (2012). The second approach is an extension of the algorithm introduced by Haahr et al. (2014) which involves the generation of paths for individual rolling stock units. This algorithm makes use of column generation to solve a MILP. We note that Cacchiani et al.

(2010) also investigated a path-based model for the rolling stock scheduling problem. Column generation in combination with rolling stock (re)scheduling is also considered by Peeters and Kroon (2008). However, a different decomposition method was performed that did not involve the generation of paths for individual rolling stock units.

The advantage of the path-based formulation based on Haahr et al. (2014) is that it models rolling stock unit duties explicitly, thereby enabling dealing with unit specific constraints naturally; a good example of such constraints are units requiring a maintenance appointment at a workshop. An optimal solution of the MILP formulation of Fioole et al. (2006) only contains information on which compositions are assigned to which trips and how compositions change between trips. However, it does not produce paths for each individual rolling stock unit. In a post processing step, a simple heuristic can construct these individual routes, because an integer flow can always be decomposed into unit valued path flows (see Ahuja et al. (1993)). However, taking constraints on individual units into account is not possible. For a discussion and other solutions for this problem we refer the reader to Wagenaar et al. (2016) (see Section 4).

Table 3.1 gives a summary of the mentioned papers in rolling stock (re)scheduling. Here, we classify every paper, including ours, based on 6 characteristics. Note that we do not compare the papers based on their computation times and practical results, as the approaches of these papers are all tested on different instances with different complexities. Therefore, in this current paper a start is made to gain more insight into the practical value of the approaches by comparing different approaches on the same instances.

From the table it can be concluded that the current path based models (Cacchiani et al. (2010), Cordeau et al. (2002) and Haahr et al. (2014)) do not take the order of units within a composition into account, while the model in the current paper does not neglect this important issue. Some of the flow based approaches take the order of units within the composition into account (Alfieri et al. (2006), Fioole et al. (2006), Lingaya et al. (2002), Nielsen et al. (2012), Peeters and Kroon (2008) and Wagenaar et al. (2016) (see Section 4)) or can be adapted to do so (Borndörfer et al. (2015)). However, as discussed above, the disadvantage of these flow based approaches is that it is not possible to put constraints on the duties of individual rolling stock units.

For a general overview of models for railway rescheduling during disruptions and disturbances we refer the reader to Cacchiani et al. (2014), and for an overview on disruption management processes in general we refer to Jespersen-Groth et al. (2009).

Paper	Scheduling (S) or Rescheduling (RS)	Locomotives & Carriages (LC) or Self-Propelled units (SP)	Path or Flow based	Order within composition	Single or Multiple (national) networks	Benchmark methods
Alfieri et al. (2006)	S	SP	Flow	Yes	Single	No
Borndörfer et al. (2015)	S	SP	Flow	Yes	Single	No
Brucker et al. (2003)	S	LC	Flow	No	Single	No
Cacchiani et al. (2010)	S	SP	Path	No	Single	No
Cadarso and Marín (2011)	S	SP	Flow	No	Single	No
Cadarso and Marín (2014)	S	SP	Flow	No	Single	No
Cordeau et al. (2001)	S	LC	Flow	No	Single	No
Cordeau et al. (2002)	S	LC	Path	No	Single	No
Fioole et al. (2006)	S	SP	Flow	Yes	Single	No
Haahr et al. (2014)	S	SP	Path	No	Single	No
Lingaya et al. (2002)	S	LC	Flow	Yes	Single	No
Nielsen et al. (2012)	RS	SP	Flow	Yes	Single	No
Peeters and Kroon (2008)	S	SP	Flow	Yes	Single	No
Wagenaar et al. (2016)	RS	SP	Flow	Yes	Single	No
This paper	S + RS	SP	Path	Yes	Multiple	Yes

Table 3.1: An overview of existing literature and their contributions. In total 6 characteristics are included: **1.** *Is the focus on scheduling or rescheduling?* **2.** *Are self propelled units or locomotives and carriages considered?* **3.** *Is the model path or flow based?* **4.** *Is the order of units within a composition taken into account?* **5.** *Is the formulation tested on a single or on multiple networks?* **6.** *Are multiple solution methods compared?*

3.3 – PROBLEM DESCRIPTION

In this section we describe the rolling stock (re)scheduling problem in more detail, and the assumptions we make.

The rolling stock *scheduling* problem consists of assigning a rolling stock composition to every trip in the timetable of one planning horizon, e.g., one day of operation. The rolling stock *rescheduling* problem consists of assigning rolling stock compositions to as many trips as possible for the remainder of the day after the timetable is adapted due to a disruption. A trip is a part of a train service, as specified by the timetable, between two major stations where the composition of a train can be changed. A trip consists of a departure station, departure time, arrival station, and arrival time. Furthermore, a composition is an ordered set of coupled rolling stock units. The assignment of rolling stock compositions should be done such that it minimizes the number of seat shortages for passengers, the total number of carriage kilometers, and several other objectives. Consequently, the rolling stock (re)scheduling problem essentially is a multi-criteria decision problem.

For each trip a maximum length of the allowed composition is given to ensure that the length of the composition assigned to a trip is not longer than the length of the shortest platform amongst all platforms where the train has a stop. Platform lengths along the network may differ, so different trips may have different maximum composition lengths.

After a trip has been operated, the composition is usually assigned to a next *succeeding* trip. Such a combination of two succeeding trips is also called a *connection* (note that this also holds for end stations of a line). It is allowed to first change the composition before it is used on the next trip. However, the transition from one composition to another must follow certain business rules. Depending on the direction in which the trip is operated and the station layout, it is stated on which side of the composition it is allowed to add (couple) extra rolling stock units and on which side it is allowed to decouple units. Such a composition change is called a *transition*.

The possibility of coupling and decoupling units requires that we keep track of the order of the rolling stock units within the composition. Most of the time it is only allowed to (de)couple on one side of the composition. For example, in the Netherlands, if a train continues in the same direction, coupling is usually done at the front of the train. This will speed up the process since the rolling stock unit could, in this situation, already be placed there before the train arrives. Decoupling will most likely take place at the rear of the train if the train continues in the same direction. As a consequence, the train can leave before the decoupled unit is shunted away. Figure 3.1 shows an example of a transition network illustrating how different composition transition rules work. An explanation of the Figure is given in the caption. Keeping track of the order of the units within a composition makes the problem more complicated than

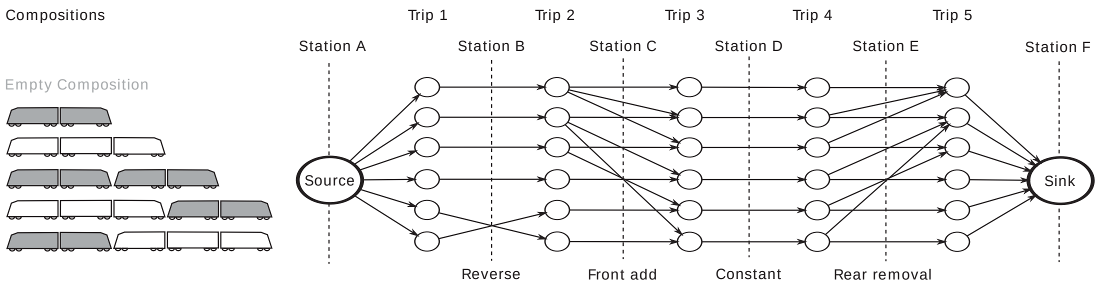


Figure 3.1: An example of a transition network illustrating how different rules affect the possible composition transitions from one trip to the successor trip. On the left the example compositions are represented, and to the right the transition network is depicted. Any path from the source to the sink represents a transition scheme that is feasible. A train travels from one station to the next. The transition rules depend on the station layout and business rules. Examples in the figure are: reversing the composition (it returns to the direction it came from), only coupling units to the front, allowing no change, and only allowing units to be decoupled from the rear.

a simple multi-commodity flow problem. Furthermore, coupling or decoupling units requires shunting personnel and time, and is therefore penalized in the objective function.

In this paper, we assume that shunting yards associated with certain stations have infinite capacity to accommodate all composition changes and to park decoupled rolling stock units. After decoupling a unit, we require a certain minimum time duration before that unit can be coupled to a new trip in another composition. This restriction reflects the time required to move a decoupled unit to the shunting yard and from the yard back to a platform to be coupled again.

For each trip an estimated passenger demand is given. It is not strictly required that the capacity for passengers of the composition assigned to a trip is equal or larger than the passenger demand for that trip. However, assigning a composition to a trip which cannot accommodate all passengers is penalized.

In contrast, considering depreciation, maintenance and energy efficiency, having more capacity than there is demand for a trip is also not preferred. Therefore, the number of kilometers driven by all carriages is penalized as well. A trade-off between the number of seat-shortages and the number of carriage kilometers must be found in the rolling stock circulation.

We must ensure that the next day the rolling stock schedule can be applied as planned. Therefore, the schedule must be such that at the end of the day the rolling stock units are parked at the shunting yards such that at each shunting yard there are as many rolling stock units as are required at the start of the next day (also called *end-of-day balance*). We allow differences in the end-of-day balance, but against a certain penalty, since for each negative unit difference a new deadheading trip should be planned during the night to re-balance the inventories. Deadheading is expensive in practice since it requires additional manpower and causes additional rolling stock costs.

Especially in the case of rescheduling, it is of significant importance that solutions are found very fast. In case of a disruption, the trains need to keep running, which means that the operator can not wait for one hour to decide how to adapt the resource schedules in order to handle the ongoing disruption.

3.4 – MATHEMATICAL FORMULATIONS

In this section we introduce the mathematical formulations of the two solution methods. We present a compact arc-flow formulation first, termed the Composition Model, which is equivalent to the formulation presented by Nielsen et al. (2012) (based on Fioole et al. (2006)). The Composition Model is a state-of-the-art solution approach for solving the considered (re)scheduling problem, and therefore provides a baseline for comparison with other new or extended approaches. The second presented approach is a path based formulation, which is an extension of the model by Haahr et al. (2014). The existing formulation is extended to consider the order of rolling stock units in train compositions in order to prohibit certain infeasible composition changes - similar to the Composition Model. With this addition, the resulting model enables greater expressive power. Rolling stock units can now be subject to unit-specific constraints such as maintenance limitations. A common notation for both methods is presented before both formulations in Sections 3.4.1 and 3.4.2 respectively.

In Tables 3.2, 3.3 and 3.4, the sets, parameters, and variables are introduced and explained respectively.

Set	Description
$\mathcal{D}$	Set of stations that have associated depot tracks
$\mathcal{C}$	Set of all compositions
$\mathcal{T}$	Set of all trips
$\mathcal{T}_{\leftrightarrow}$	Set of all connections
$\mathbf{s}_r$	The first trip of connection $r \in \mathcal{T}_{\leftrightarrow}$
$\mathbf{t}_r$	The second trip of connection $r \in \mathcal{T}_{\leftrightarrow}$
$\mathbf{d}_r$	The station at which connection $r \in \mathcal{T}_{\leftrightarrow}$ takes place
$\mathcal{U}$	Set of train unit types
$\mathcal{C}_r^2$	Set of all combinations of compositions (c, c') allowed for connection r. This means that composition c is allowed for trip $\mathbf{s}_r$ and composition c' is allowed for trip $\mathbf{t}_r$

Table 3.2: List of sets and elements

Parameter	Description
sod_d^u	Starting station inventory of unit type $u \in \mathcal{U}$ at station $d \in \mathcal{D}$
eod_d^u	Target end-of-day inventory of unit type $u \in \mathcal{U}$ at station $d \in \mathcal{D}$
$compCost_c^t$	Combined costs of trip cancellation, trip seat shortages, and operational costs if composition $c \in \mathcal{C}$ is used on trip $t \in \mathcal{T}$
$transCost_{c,c'}^r$	Coupling costs of changing from composition $c \in \mathcal{C}$ to $c' \in \mathcal{C}$ between trip $\mathbf{s}_r$ and $\mathbf{t}_r$ of connection r
$eodCost$	The penalty for a single end-of-day balance shortage

Table 3.3: List of parameters and coefficients

Variable	Description
y_c^t	Binary decision variable deciding whether composition $c \in \mathcal{C}$ is used on trip $t \in \mathcal{T}$
$z_{c,c'}^r$	Binary decision variable deciding whether composition $c \in \mathcal{C}$ and $c' \in \mathcal{C}$ are used for trip $\mathbf{s}_r$ and $\mathbf{t}_r$ respectively
i_d^u	Integer decision variable representing the end-of-day balance shortage for unit type $u \in \mathcal{U}$ at station $d \in \mathcal{D}$

Table 3.4: List of Variables

Both formulations are Mixed Integer Linear Programs, where the objective function is defined as:

$$\text{Minimize:} \quad \sum_{t \in \mathcal{T}} \sum_{c \in \mathcal{C}} compCost_c^t \cdot y_c^t \tag{3.1}$$

$$+ \sum_{r \in \mathcal{T}_{\leftrightarrow}} \sum_{c \in \mathcal{C}} \sum_{c' \in \mathcal{C}} transCost_{c,c'}^r \cdot z_{c,c'}^r \tag{3.2}$$

$$+ \sum_{u \in \mathcal{U}} \sum_{d \in \mathcal{D}} eodCost \cdot i_d^u \tag{3.3}$$

The objective function consists of three parts: costs for assigning compositions to trips, costs for assigning transitions between compositions, and costs for having end-of-day off balances. Trip cancellation, seat shortage and operational costs are included in (3.1). The shunting costs are included in (3.2), and the end-of-day shortage costs are accounted for in (3.3).

In both formulations, the relationship between the composition variables and the transition variables is constrained in order to comply with the physical rules and business logic. For a given trip the rules stipulate which compositions are allowed on the following trip:

$$\sum_{c \in \mathcal{C}} y_c^t = 1 \qquad \forall t \in \mathcal{T} \tag{3.4}$$

$$y_c^{\mathbf{s}_r} = \sum_{c' \in \{c_2 | (c, c_2) \in \mathcal{C}_r^2\}} z_{c,c'}^r \qquad \forall r \in \mathcal{T}_{\leftrightarrow}, c \in \mathcal{C} \tag{3.5}$$

$$y_{c'}^{\mathbf{t}_r} = \sum_{c \in \{c_1 | (c_1, c') \in \mathcal{C}_r^2\}} z_{c,c'}^r \qquad \forall r \in \mathcal{T}_{\leftrightarrow}, c' \in \mathcal{C} \tag{3.6}$$

Constraints (3.4) ensure that exactly one composition is assigned to each trip and Constraints (3.5) and (3.6) ensure that a feasible path is found in the transition network, see Figure 3.1. Note, that the empty composition is a valid composition; however, this composition assignment has a high penalty, as it corresponds to canceling a trip. The composition assigned to the incoming trip in a connection ($y_c^{\mathbf{s}_r}$) must be equal to the actual incoming composition in the chosen transition, as modelled by Constraints (3.5). Similarly, the composition assigned to the outgoing trip in a connection ($y_{c'}^{\mathbf{t}_r}$) must be equal to the actual outgoing composition in the chosen transition, as modelled by Constraints (3.6).

These constraints do not consider the availability of rolling stock units and do not measure the end-of-day shortages i_d^u. In the following two sections we discuss how the two different models take this into account.

3.4.1 – The Composition Model

The first option to consider the availability of rolling stock is to keep track of the inventory of rolling stock units at the stations. This option is applied in the formulation of Nielsen et al. (2012), which is based on Fioole et al. (2006). In this section this formulation is summarized.

For this formulation additional parameters and variables are necessary to determine at each station the inventory and the number of coupled and decoupled units. The parameters $coup_{c,c'}^u$ and $uncoup_{c,c'}^u$ indicate how many rolling stock units of type $u \in \mathcal{U}$ should be coupled or uncoupled respectively if the composition changes from $c \in \mathcal{C}$ to $c' \in \mathcal{C}$. These values can not be negative. For instance, if 2 units of type u need to be coupled during a composition change from composition c to composition c', then $coup_{c,c'}^u = 2$ and $uncoup_{c,c'}^u = 0$. Furthermore, we assume a certain processing time to shunt a decoupled unit to a shunting yard and to get it back from the shunting yard. Therefore, a parameter τ_r^- is used for each connection r indicating the time when the units that are decoupled during connection $r \in \mathcal{T}_{\leftrightarrow}$ (in between trips $\mathbf{s}_r$ and $\mathbf{t}_r$) are available again for coupling. Also a parameter τ_r^+ is required for each connection r indicating the time at which connection r takes place. This is the time the units, which need to be coupled to the composition between trips $\mathbf{s}_r$ and $\mathbf{t}_r$, should be available. The station at which connection r takes place is indicated by $\mathbf{d}_r$. Note that this is the station where trip $\mathbf{s}_r$ ends and where trip $\mathbf{t}_r$ starts.

For connection $r \in \mathcal{T}_{\leftrightarrow}$ and rolling stock unit type $u \in \mathcal{U}$, the non-negative integer decision variables $v^+_{r,u}$ and $v^-_{r,u}$ indicate respectively the number of rolling stock units of type u that are coupled to the composition between trips $\mathbf{s}_r$ and $\mathbf{t}_r$, and the number of rolling stock units of type u that are uncoupled from the composition between trips $\mathbf{s}_r$ and $\mathbf{t}_r$. Furthermore, a decision variable $inv_{r,u}$, representing the inventory just after connection $r \in \mathcal{T}_{\leftrightarrow}$ (τ^+_t) of rolling stock units of type $u \in \mathcal{U}$ at station $\mathbf{d}_r$, is required.

With these decision variables we can formulate the overall model, where constraints are added to ensure that: *i)* the inventory is non-negative at each station and each time period and *ii)* the end-of-day off-balance is correctly measured.

$$\text{Objective: } (3.1) - (3.3) \tag{3.7}$$

$$\text{Constraints: } (3.4) - (3.6)$$

$$v^+_{r,u} = \sum_{(c,c') \in \mathcal{C}^2_r} coup^u_{c,c'} \cdot z^r_{c,c'} \qquad \forall r \in \mathcal{T}_{\leftrightarrow}, u \in \mathcal{U} \tag{3.8}$$

$$v^-_{r,u} = \sum_{(c,c') \in \mathcal{C}^2_r} uncoup^u_{c,c'} \cdot z^r_{c,c'} \qquad \forall r \in \mathcal{T}_{\leftrightarrow}, u \in \mathcal{U} \tag{3.9}$$

$$inv_{r,u} = sod^u_{\mathbf{d}_r} - \sum_{\substack{r' \in \mathcal{T}_{\leftrightarrow}: \\ \mathbf{d}_{r'} = \mathbf{d}_r, \\ \tau^+_{r'} \le \tau^+_r}} v^+_{r',u} + \sum_{\substack{r' \in \mathcal{T}_{\leftrightarrow}: \\ \mathbf{d}_{r'} = \mathbf{d}_r, \\ \tau^-_{r'} \le \tau^+_r}} v^-_{r',u} \qquad \forall r \in \mathcal{T}_{\leftrightarrow}, u \in \mathcal{U} \tag{3.10}$$

$$i^u_d \ge (eod^u_d - sod^u_d) - \sum_{r \in \mathcal{T}_{\leftrightarrow}: \mathbf{d}_r = d} v^-_{r',u} + \sum_{r \in \mathcal{T}_{\leftrightarrow}: \mathbf{d}_r = d} v^+_{r',u} \qquad \forall d \in \mathcal{D}, u \in \mathcal{U} \tag{3.11}$$

$$i^u_d \in \mathbb{Z}^+ \qquad \forall d \in \mathcal{D}, u \in \mathcal{U} \tag{3.12}$$

$$v^+_{r,u}, v^-_{r,u}, inv_{r,u} \in \mathbb{R}^+ \qquad \forall r \in \mathcal{T}_{\leftrightarrow}, u \in \mathcal{U} \tag{3.13}$$

Constraints (3.10) determine the inventory just after a connection takes place. The inventory is equal to the start inventory at the station, minus all units being coupled to compositions in earlier connections, plus all units which are available again after being decoupled from compositions in earlier connections. Note, that since the variables $inv_{r,u}$ are non-negative it also ensures that the inventory is zero or positive at all times. Constraints (3.11) make sure that the end-of-day shortage is captured. This is equal to the planned difference between the start and end inventory, minus the realized difference between the number of units coupled (departed) and decoupled (arrived) during the day at that station. Constraints (3.12) and

(3.13) ensure the non-negativity of the number of (de)couplings, inventories and end-of-day shortages. We note that the integrality of the end-of-day balance variables can be relaxed as these will be naturally integer valued.

3.4.2 – The Path Based Model

In Section 3.4.1 a flow based model for the Rolling Stock (Re)Scheduling Problem is presented that identifies the correct number of units flowing from station to station. The Rolling Stock (Re)scheduling Problem can also be formulated using a path based model. In contrast to the flow based model, the path based formulation explicitly considers the route of each individual rolling stock unit and in this way the availability of rolling stock is guaranteed. The main advantages of this approach is the ability to model unit-specific constraints more easily, e.g., mileage-limitations before next maintenance, or route-choices for units displaying region-dependent commercials. We refer to Haahr et al. (2014) for a study using unit specific constraints. In contrast, the main disadvantage is the added complexity of monitoring the unit paths, in comparison to only considering the flow of unit types in the Composition Model.

The number of possible paths for a rolling stock unit grows exponentially in the number of trips, making the full path model computationally intractable. We therefore propose to solve this model by using column generation, i.e., only a subset of all possible paths is considered. By iteratively solving this reduced model (also known as the master problem) and adding columns with negative reduced cost (found using a so-called subproblem) we are able to get an optimal solution. We refer to Desaulniers et al. (2005) for a detailed introduction to column generation and *Branch and Price* frameworks.

Recall from Section 3.1 that this model is an extension of the model presented in Haahr et al. (2014). The model is solved using a *Branch and Price* framework, i.e, a *Branch And Bound* (BAB) approach where columns are added dynamically (as needed to prove optimality) at every node in the BAB search tree. The main difference is that we have to consider the order of the units within a composition. For instance, the composition *aba* is different from the composition *baa*. This is important for the composition change rules in practice. The unit in the middle of a composition can, for example, not be uncoupled. Therefore Constraints (3.4)-(3.6) are used in this formulation as well.

In the remainder of this formulation, $\mathcal{P}_d^u$ is defined as the set of all paths for unit type $u \in \mathcal{U}$ starting in station $d \in \mathcal{D}$. A path describes a chronological list of trips that are performed by a single rolling stock unit, i.e., a unit's schedule for the planning period. The set of all possible paths is thus denoted by $\mathcal{P} := \bigcup_{u \in \mathcal{U}, d \in \mathcal{D}} \mathcal{P}_d^u$. Shunting operations are implicit as a unit has to be uncoupled (and therefore shunted) whenever the unit exits a trip connection, and coupled whenever the unit, going out of the shunting yard, enters a connection.

The (re)scheduling problem simply consists of assigning exactly one path to each unit, subject to a number of constraints. Therefore, the master formulation contains additional sets of binary decision parameters and variables. The first set of binary parameters, α_p^t, states for each path $p \in \mathcal{P}$ and trip $t \in \mathcal{T}$ whether path p covers trip t ($\alpha_p^t = 1$) or not ($\alpha_p^t = 0$). The second set of binary parameters, β_p^d, takes the value 1 if path $p \in \mathcal{P}$ terminates at station $d \in \mathcal{D}$, and otherwise equals 0. The third set of integer parameters, μ_c^u, indicates how many units of type $u \in \mathcal{U}$ are assigned to composition $c \in \mathcal{C}$. The binary decision variables $\lambda_p \in \{0,1\}$ determine whether path $p \in \mathcal{P}$ is selected in the final solution or not.

We note that the penalties in equations (3.1) and (3.2) of the objective function can be included in the subproblem instead of in the master problem, or alternatively partially included in both. Our preliminary results have shown that it is beneficial for the overall column generation convergence to put a part of the coupling costs in the subproblem. Without awareness of the coupling costs in the subproblem it may generate paths that are unnecessarily expensive, and possibly also incompatible in the master problem.

This leads to the following constraints in the master problem:

$$\text{Objective: } (3.1) - (3.3) \tag{3.14}$$

$$\text{Constraints: } (3.4) - (3.6)$$

$$\sum_{d \in D} \sum_{p \in \mathcal{P}_d^u} \alpha_p^t \lambda_p = \sum_{c \in \mathcal{C}} \mu_c^u y_c^t \qquad \forall u \in \mathcal{U}, t \in \mathcal{T} \tag{3.15}$$

$$\sum_{p \in \mathcal{P}_d^u} \lambda_p = sod_d^u \qquad \forall u \in \mathcal{U}, d \in \mathcal{D} \tag{3.16}$$

$$i_d^u + \sum_{p \in \mathcal{P}_d^u} \beta_p^d \lambda_p \geq eod_d^u \qquad \forall u \in \mathcal{U}, d \in \mathcal{D} \tag{3.17}$$

$$\lambda_p \in \{0,1\}, \quad y_c^t \in \{0,1\}, \quad i_d^u \in \mathbb{Z}_0^+ \tag{3.18}$$

Without equations (3.4)-(3.6) the master problem solely consists of finding a set of rolling stock paths. Together with Constraints (3.15) these constraints ensure that only feasible compositions and composition transitions are made.

In comparison with the Composition Model the path-formulation does not consider inventories at the stations, since it models individual paths for rolling stock units, thereby implicitly handling the inventories. The composition and path variables are linked by Constraints (3.15) to ensure that the correct number of rolling stock units (and types) are assigned to each trip composition. Constraints (3.16) ensure that exactly <u>one</u> path is assigned to every rolling stock unit. Note that it is necessary for all units to have a path in order to account for the end-of-day balance. A unit's path contains no trips if the unit stays at a depot for the entire period. The end-of-day balance is enforced in Constraints (3.17). A slack (which is penalized in the objective) is inflicted if insufficient units terminate at respective stations. Finally, the domains of the variables are shown in (3.18). We note that the integrality of the end-of-day balance variables can be relaxed as these will be naturally integer valued.

The master problem ((3.1)-(3.6) and (3.15)-(3.18)) is solved iteratively while adding new columns (path variables) that have negative reduced cost. A subproblem is solved, using the duals of (3.15)-(3.17), to find such columns or to prove that they do not exist. This problem can be solved as a shortest path problem (or with resource constraints when unit-specific constraints are enforced). When no columns exist with negative reduced cost we have solved the LP relaxation to optimality.

A *Branch and Price* framework is required to solve the problem, as the LP relaxations of the master problem are not necessarily integral. The framework setup and branching rules as described by Haahr et al. (2014) are adopted. Here, branches can be made to stipulate that the number of units on one trip are integral, and that the number of units originating or ending in a station are integral.

In addition to these branching rules, we introduce a new type of branching on the composition variables. Note that for any given trip, an optimal LP solution is not required to assign a value of 1 to one of the composition variables, but can assign a fractional value to some of them as long as the sum is 1. In such cases the branching rule partitions the trip's composition variables into two groups, such that the sum of the fractional values is non-zero in both groups. Note that this is stronger than finding one variable to branch on. Given a trip $t \in T$ with multiple non-zero composition assignments in an LP relaxed solution, we obtain two branches:

$$\sum_{c \in C_1} y_c^t \geq 1 \tag{3.19}$$

and

$$\sum_{c \in C_2} y_c^t \geq 1 \tag{3.20}$$

where C_1 and C_2 define the described partitions. In each branch the sum of one group should be greater than or equal to one. Consequently the sum of the other group is then zero. The same reasoning holds in the other branch.

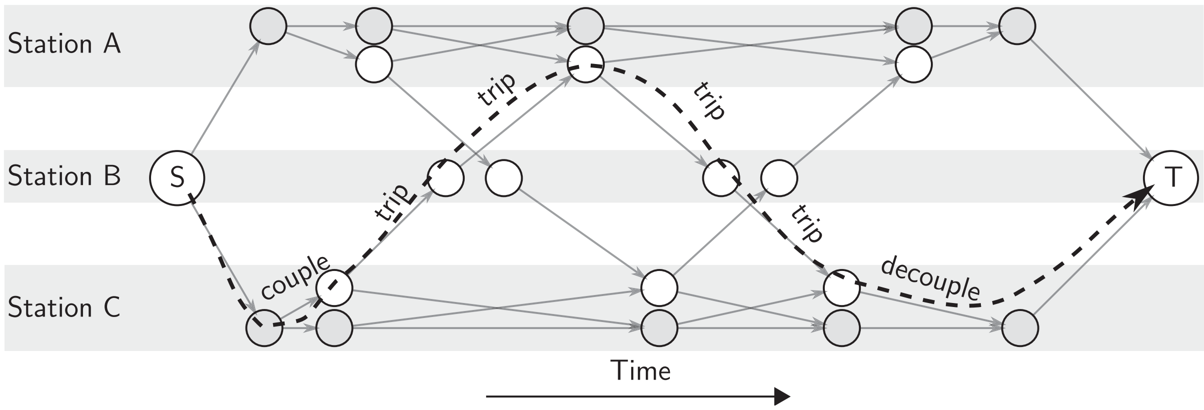


Figure 3.2: A simplified example of the acyclic time-space network showing some of the possible coupling arcs. Three stations (A,B and C) are illustrated where only two of them (A and C) have associated depot tracks. An example of a feasible unit schedule has been highlighted using a dashed path.

3.4.2.1 – Subproblem Graph Example

An example of the subproblem graph is shown in Figure 3.2. Note that the underlying graph is acyclic and the weights can be negative. We refer to Haahr et al. (2014) for an in-depth description. Note, that the structure of the subproblem is the same as in Haahr et al. (2014) even though we take the order of train units within a composition into account in the overall method. The example contains three stations whose events (departures and arrivals) are shown as vertically aligned vertices. Stations with associated depot tracks have a paired (gray color) node, representing the depot (i.e. shunting yard). The graph contains one source (S) that has one out-going arc to the first events of all stations, and one target node (T) that has one in-going arc for the last events on different stations. Intermediate arcs represent train movements (moving units between the depot tracks and a platform) and trips travelling from one station to another. Mileage costs are set on trip arcs and coupling costs are set on shunting arcs. Dual costs from the linking Constraints (3.15) are assigned to the trip arcs. Inventory duals from Constraints (3.16) are assigned to the arcs extending from the source. Finally, duals from Constraints (3.17), on the end-of-day balances, are assigned to the target node arcs. We note that, without loss of generality, the subproblem can be changed slightly in implementation as some nodes and edges can be altered or removed.

3.4.2.2 – Alternative Formulations

A mixed integer program equivalent to the formulation in (3.14)-(3.18) can be obtained by replacing the composition transition rules (Constraints (3.5)-(3.6)) with Constraints (3.21) which are based on the composition variables only. For any pair of successive trips say t_1 and

t_2 (i.e. a connection) we would require:

$$y_c^{t_1} \leq \sum_{c' \in C_c^3} y_{c'}^{t_2} \quad \forall c \in \mathcal{C} \tag{3.21}$$

Here C_c^3 is the set of allowed compositions in t_2 following the composition c on trip t_1. These constraints limit the origin composition c, for example $y_c^{t_1}$ cannot be set to 1 if the following trip has an incompatible composition. Likewise, a similar set of constraints can be defined that limit the target composition. The resulting formulation has fewer variables and constraints, which may provide a significant speedup when solving the LP relaxations. As a column generation framework relies on solving the LP relaxation many times, this could significantly reduce total computation time. However, preliminary results show that this alternative formulation weakens the LP relaxation too drastically; the relaxed solutions are more fractional. The benefit of faster LP solution times does not outweigh the weaker relaxation in general.

Another equivalent formulation to (3.14)-(3.18) can be obtained by modeling the composition transition rules of Constraints (3.5)-(3.6) by conflicting path constraints as in Equation (3.22). Given a full enumeration of all possible paths in the formulation presented in Haahr et al. (2014) (which routes all rolling stock units without any restrictions on train compositions), the composition rules can be enforced by including constraints that prohibit the selection of pairs of conflicting routes. Therefore, the set Π is introduced, this set contains all pairs of conflicting paths. A pair of paths is conflicting if by choosing both paths in the solution, it means that one of the composition rules is violated, e.g. only allowing a unit to couple or uncouple on one end of the existing train composition, or disallowing incompatible unit types to be combined on train services.

$$\lambda_{p_1} + \lambda_{p_2} \leq 1 \quad \forall (p_1, p_2) \in \Pi \tag{3.22}$$

For example, consider a sequence of trips { a, b, c, d } where no intermediate turnings take place. Say path p_1 services all trips, path p_2 only services trips {b, c}, and path p_3 only services trips {b, c, d}. Assume that the station between trips a and b only allows units to couple and uncouple at the front of the train, and that the station between trips c and d only allows units to couple or uncouple at the back of the train. As a consequence, paths p_1 and p_2 cannot coexist, as the latter path cannot be coupled to the front of the train and uncoupled from the back when path p_1 is used as well. Thus $(p_1, p_2) \in \Pi$. On the other hand, p_1 and p_2 are both compatible with p_3.

The obvious disadvantage of this formulation is the potentially large number of resulting conflict constraints. This also turned out to be the case after testing this approach on the instances discussed in the computation results section. A delayed cut callback routine, that generates the conflicting constraints as they become violated, can be adopted in a Branch-and-Cut method, and possible remedy this drawback, as only few paths are expected to enter the Simplex basis. However, the non-static row-dimension of the formulation complicates an exact delayed column generation as there is no obvious effective solution approach for the resulting subproblem, where new variables with negative reduced-costs must be found or proven to not exist. Without formally knowing all the possible conflict constraints in advance, it is non-trivial to formulate a subproblem that can account for all necessary dual contributions.

If a full enumeration of all paths is intractable, then a heuristic framework can be adopted that pre-generates an initial set of paths. When all paths are known in advance, the set of pairwise conflicting paths can be deduced and inserted into the formulation. A MIP solver can then be used to solve the resulting formulation. A disadvantage of this approach is the generation of the initial pool of paths. A set of initial paths must be generated, which must contain good and non-conflicting paths. Some of the paths must be able to contain overlapping trips without being in conflict. One approach can include all paths generated while solving the LP-relaxation of the root node in the formulation presented by Haahr et al. (2014). The resulting paths are not proven to be compatible, but this approach requires no special phase for selecting initial paths. Applying this is, however, out of the scope of this paper. Therefore, we use the version with composition constraints in the remainder of this paper.

3.4.3 – Delayed Transition Constraint Generation

Preliminary results show that the LP relaxation of the formulation requires a significant amount of pivot operations to solve. This is highly undesirable since we are continuously solving the formulation after adding columns in every iteration of the column generation process. We discovered that solving the linear relaxation of the master problem with the Dual Simplex Method was significantly faster than using the Primal Simplex Method. However, in column generation we unfortunately rely on the Primal Simplex Method as subsequent LP resolves can be hot-started using the old optimal solution after adding new columns.

These observations did however lead us to experiment with the formulation in order to improve convergence or runtime speed of the linear relaxation. Constraints (3.5) and (3.6) do not only represent a significant amount of constraints in the presented formulation, but also have a large influence on which columns are feasible during the convergence process. As a result, unnecessary columns are generated making the approach slow. When converging against the optimal linear relaxation, these constraints are unnecessarily restrictive as the intermediate set of states (fractional LP solutions) is not important. By removing these constraints from consideration initially, the advantages are three-fold: we achieve a smaller linear program to resolve, we may gain a faster convergence, and finally we get better duals due to the omission of equality constraints. In the method we therefore propose a variant that dynamically adds Constraints (3.5) and (3.6) as they become violated once the linear program has converged. This step merely consists of looping through the allowed composition transitions and checking whether the current solution violates any constraint. This extension comes at the cost of more complexity, but preliminary results demonstrated that the benefits outweight the costs in most instances.

For the sake of simplicity we only add the dynamic constraints once no negative reduced cost columns can be found. Thus, the proposed variant searches for new columns first, and for violated constraints second. Note, after adding new constraints, the framework searches for negative reduced cost columns again since a new constraint changes the solution. The LP model has been solved to optimality once no columns can be generated and none of the constraints is violated.

3.5 – COMPUTATIONAL EXPERIMENTS

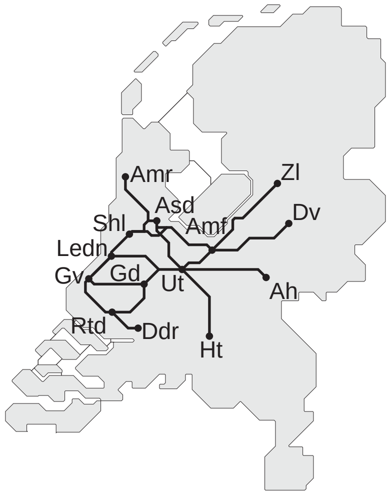
In this section we present and discuss our computational experiments. We test both models on a scheduling case from Netherlands Railways (NS) and on a scheduling case from the Copenhagen Suburban Railway Operator (DSB S-tog). Additionally, we test the models on different rescheduling cases from both NS and DSB S-tog.

We start with describing the railway network characteristics for both networks. Thereafter, we introduce the parameter settings we have adopted. These parameter settings are chosen after several runs to benchmark their influence. Then, a comparison of the computational results of the scheduling instances is made. Afterwards, the rescheduling instances are described in detail for both the NS and DSB S-tog instances. Finally, we conclude the section with an overview of the different rescheduling results.

3.5.1 – Railway network characteristics

3.5.1.1 – Netherlands Railways

A single instance on the Dutch railway network that spans the major part of the intercity network of NS is selected as computational case. Figure 3.3 shows the trajectory of the network. The majority of the lines are in the western part of the Netherlands which is the busiest part of the Dutch railway network. In total we consider the timetable of 16 distinct lines of a certain weekday, namely the Monday. There are different timetables required on the other days. However, the differences between those timetables are small. Therefore, we only consider the timetable of the Monday as scheduling instance. Note that all trips of the 16 lines during the day are taken into account. In general the frequency of the operated lines is half-hourly, but some lines are operated on an hourly basis as can be seen in the table. This leads to a total of 2324 trips between 14 different major stations. The longest trip is between Zl and Amf and has length 67km, while the shortest trip is between Ledn and Gvc with length 16km.



Line	Stations	Frequency
500	Gv Gd Ut Amf Zl	hourly
700	Shl Amf Zl	hourly
800	Amr Zd Asd Ut Ht	half hourly
1500	Asd Amf Dv	half hourly
1600	Shl Amf Dv	hourly
1700	Gv Gd Ut Amf Dv	hourly
1900	Gv Rtd Ddr	half hourly
2000	Gv Gd Ut Ah	half hourly
2100	Asd Shl Ledn Gv Rtd Ddr	half hourly
2600	Asd Shl Ledn Gv	half hourly
2800	Rtd Gd Ut Amf	half hourly
3000	Amr Asd Ut Ah	half hourly
3500	Shl Ut Ht	half hourly
8800	Ledn Ut	half hourly
20500	Rtd Gd Ut	hourly
21700	Rtd Gd Ut	hourly

Figure 3.3: The NS network considered in the test instances.

We have two different rolling stock types available. A rolling stock unit of type a consists of 4 carriages and has a passenger capacity of 405 seats, and a rolling stock unit of type b consists of 6 carriages and has a passenger capacity of 597 seats. The maximum composition length for all trips is 14 carriages in the considered network. This leads in total to 11 different compositions that can be appointed to a trip: a, aa, aaa, b, bb, ab, ba, aab, aba, baa, and the empty composition meaning that a trip is cancelled.

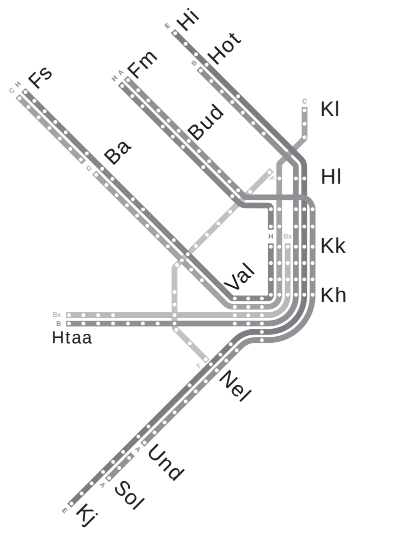
There are many trips between stations where no shunting is allowed. As a consequence, the compositions of those trips are the same as the ones appointed to their predecessor trips. We use a preprocessing step to merge those trips. After preprocessing, the total number of non-reducible trips is 727, i.e., in fact only 727 trip compositions need to be decided as the rest will be fixed due to the composition transition rules. Shunting is allowed at the stations between the remaining trips. It depends on the station whether (un)coupling activities are allowed at the front or the rear side of the incoming train. Furthermore, it is not allowed to both couple and uncouple train units at the same time from the incoming train.

3.5.1.2 – DSB S-tog

The suburban train service network provided by DSB S-tog in Copenhagen is operated using a weekly periodic timetable and an hourly periodic timetable on a daily basis. Weekdays and weekends are operated using slightly different timetables, and an additional set of night trains are operated on Friday and Saturday nights. This leaves a total of four different timetable schedules: Monday (to Thursday), Friday, Saturday and Sunday. Table 3.5 summarizes the instance specifics. Note that the A, B, C, and F lines are considered in all cases, while the Monday and Friday instances contain more lines than just these 4 lines. As can be seen, the trains are operated with a frequency of either every 10 or every 20 minutes. The longest trip is between the stations FS and KL with length 55km, while the shortest trip is between the stations KK and KH with length 3km.

Name	Stops	Trips	Trips*	Weekday	Lines
DSBfri	28 719	4 558	886	Friday	A,B,Bx,C,E,F&H
DSBsat	20 474	1 916	590	Saturday	A,B,C&F
DSBsun	19 919	1 871	574	Sunday	A,B,C&F
DSBmon	28 017	4 468	868	Monday	A,B,Bx,C,E,F&H

Table 3.5: Four timetables operated by DSB S-tog. The columns respectively show the instance names, total number of stops, total number of trips, total number of non-reducible trips (Trips), weekday, and finally the lines that are running.*



Line	Stations	Frequency	
A	Kj Sol Und Kh Kk Bud Fm	10&20	20
B	Htaa Val Kh Kk Hl Hot	10&20	20
C	Fs Ba Val Kh Kk Hl Kl	10&20	20
E	Kj Sol Und Kh Kk Hl Hot Hi	10&20	20
F	(Kl) Hl Nel	10&20	20
H	Fs Ba Val Kh Kk Bud Fm	20	20

Figure 3.4: The DSB S-tog network considered in the test instances. Only key stations are listed in the table. The table columns show line name, station names and the operated frequencies in minutes for Monday-Friday and Saturday-Sunday respectively.

Two unit types are used to perform all train services. The smaller unit type *e* is 42 meters long and contains 150 passenger seats, the larger unit type *f* is 83 meters and holds 336 passenger seats. Platform lengths vary from station to station in Denmark as well, and the current lines partition the composition lengths into two sets. The first set allows only small compositions *e*, *ee*, and *f*. The second set also allows larger compositions *eee*, *fe*, *ef*, and *ff*.

There are also many trips between stations where no shunting is allowed. The original number of trips and the number of remaining trips after merging are both showed in Table 3.5. Shunting is allowed between the remaining trips and it depends on the station whether coupling and uncoupling take place at the front or rear side of the incoming train.

3.5.2 – Rolling Stock Scheduling

In this section we start with describing the parameters used for the scheduling instances for Netherlands Railways and DSB S-tog. Thereafter we present the results of the scheduling instances.

3.5.2.1 – Instance parameters

Here we will present the parameters used for the scheduling instances. We start in Section 3.5.2.1.1 with the parameters for Netherlands Railways and in Section 3.5.2.1.2 the parameters for DSB S-Tog.

3.5.2.1.1 – Netherlands Railways

In the scheduling phase of NS the goal is to determine the start and end inventory at every station, and to appoint compositions to trips to fulfill the passenger demand. This is solved in two steps. First a suitable initial fleet distribution is found. Afterwards the found initial inventory setting is used to create the rolling stock circulation.

In the first step, our objective is to create a rolling stock circulation that covers all demand while using the least amount of carriages. To this end, the model determines the number of rolling stock units required at the start of the day at each station to create such a circulation. We set a small penalty on the number of carriage kilometers. No penalty at all is set on the number of shunting operations applied during the day, and no penalty is given for deviations between the start and end inventory.

Our first step provides us with a start circulation that contains in total 113 units of type a and 58 units of type b. If this start circulation is used, NS is able to cover the forecasted passenger demand.

In the second step, we use the start inventory created in the first step as fixed input. It is not allowed to deviate from this start inventory and the objective is then to find a rolling stock circulation that minimizes the costs using the objective coefficients as shown in Table 3.6. These penalty values are commonly used in literature, but we check the influence of changing the Mileage and Seat Shortage penalty ratio on the results in Section 3.5.2.2.

Here, *Cancel* means the penalty for cancelling a trip, *Mileage* denotes the penalty for driving a single carriage over one kilometer, *Seat Shortage* defines the penalty for having one seat shortage per kilometer, *Shunting* refers to the penalty for doing a shunting operation, and, finally, *End diff* denotes the penalty for a negative difference between the preferred end-of-day inventory and the actual end inventory at a station. Results are shown and discussed in Section 3.5.2.2.

Coefficient	NS
Cancel	1 000 000
Mileage	1
Seat Shortage	0.5
Shunting	1 000
End diff	10 000

Table 3.6: Penalty values used for the NS scheduling case

3.5.2.1.2 – DSB S-tog

DSB S-tog provided us with a start inventory. Thus, for S-tog instances it is not necessary to perform the first step of scheduling, that was described for the NS instance. The start inventory contains 31 units of type e and 103 units of type f.

Furthermore, we use distinct parameter settings for NS and S-tog since the characteristics of the network and train unit fleets are different. The parameter settings for the S-tog network can be found in Table 3.7. As can be seen, the only difference between the two settings is the mileage penalty. This is, among other factors, because the rolling stock units of DSB S-tog are measured by length in meters in contrast to the number of carriages. The length in meters is substantially larger than the number of carriages.

3.5.2.2 – Computational results

In this section we benchmark the two proposed methods for solving the rolling stock scheduling problem on both NS and DSB S-tog instances. We compare both NS and DSB S-tog instances on the scheduling phase. The mathematical formulations are equivalent and we have verified

Coefficient	DSB S-tog
Cancel	1 000 000
Mileage	0.1
Seat Shortage	0.5
Shunting	1 000
Start end diff	10 000

Table 3.7: Penalty values used for the DSB S-tog scheduling case

that the optimal objective costs are identical for the two methods. The objective function is a mix of several penalties where the exact balance between them is non-trivial to set. Due to this, there is little justification for solving to true optimality. In practice, it might be enough to accept solutions that are within 1% of optimality, but to be able to compare the approaches accurately we solve to optimality.

In order to justify the parameter values as presented in Tables 3.6 and 3.7, we investigated the trade-off between the seat shortage and mileage costs. All other penalties are fixed on their value as given in Tables 3.6 and 3.7, only the mileage costs are increased. The results are shown in Figure 3.5. As can be seen, there is a steady increase in mileage costs as the seat cover moves towards 100%. However, when we get close to 99% we observe diminishing returns. The final percentages to get a 100% cover are the most difficult percentages. The passenger demand on some trips is too large, as a consequence train units from other stations have to be transported in order to cover the trip with enough passenger demand. Consequently, too large compositions are appointed to the other trips which leads to a solution with larger costs. A cover of 100% is preferred by rolling stock planners, however not at all cost. In practice a seat cover close to 100% with a rolling stock circulation that is not overly expensive is chosen.

The settings of the parameters do not only influence the costs of the rolling stock circulation, but also the computation time to find the circulation. Information about how the computation times are affected by the different penalties for seat shortages are provided in Table 3.8 for all scheduling instances. Here A denotes the Path Based Model without row generation, B the Path Based Model with row generation, and N the Composition Model. As can be seen, B is able to solve the scheduling instances on average faster than A. This will be discussed in more detail in Section 3.5.4. Furthermore, N is able to solve the instances even faster. All solution times are acceptable for the planning phase. We assume that a solution method resulting in optimal schedules within 15 minutes can be used in practice, because there is plenty of time available in the planning phase.

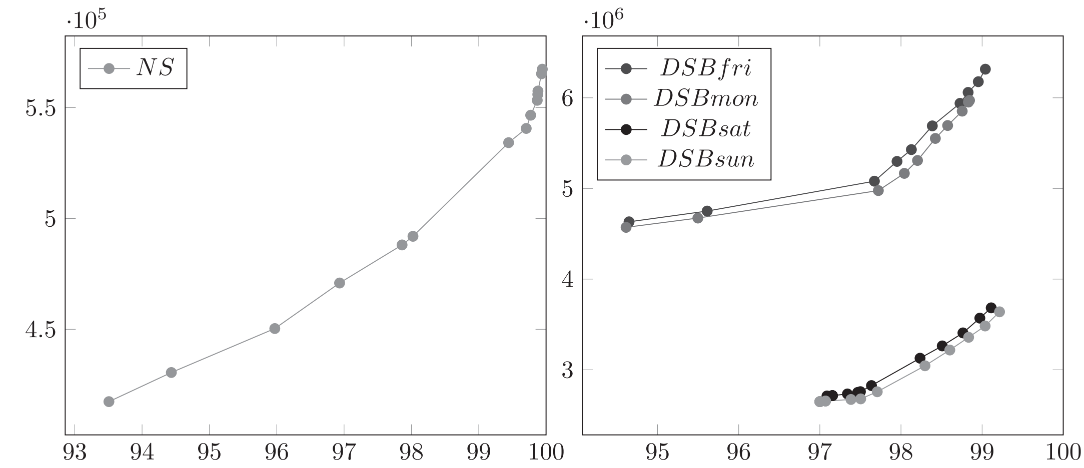


Figure 3.5: Trade-off between mileage costs and seat/demand cover for the NS (left) and DSB S-tog (right) instance. The vertical and horizontal axis respectively show the mileage usage (in km) and seat cover percentage.

	NS			DSBmon			DSBfri			DSBsat			DSBsun		
Costs	A	B	N	A	B	N	A	B	N	A	B	N	A	B	N
0.1	444	84	27	398	30	14	314	63	14	14	3	2	21	3	3
0.2	545	65	23	575	44	13	423	59	13	21	6	3	29	4	2
0.3	742	274	16	652	345	21	556	48	21	31	6	3	40	4	3
0.4	712	493	17	579	96	14	479	38	12	57	5	4	41	4	4
0.5	709	465	19	653	119	19	664	37	12	58	10	7	35	4	2
0.6	946	781	16	760	47	11	709	40	14	36	8	5	31	4	3

Table 3.8: Detailed runtimes in seconds using the solution methods on the non-disrupted test instances. The columns respectively show the computation times to solve the instances with the A, B, and N method.

Recall that the Composition Model and the Path Based Model give the same results in terms of objective values. We have selected an appropriate balance in the following benchmarks by choosing the settings as shown in Table 3.6 and Table 3.7. These settings lead to the results as shown in Table 3.9. We have left out the results of the A method, because they are all dominated by the B method. Note that in the DSB S-tog cases it is not possible to reach a seat coverage equal to 100%. The seat coverage for all instances is above 98%, so almost all passengers fit in the appointed compositions with a seat. Furthermore, there is no deviation between the start and end-of-day inventories. This is highly appreciated in practice, because otherwise empty trains need to be scheduled to balance the differences. We can conclude that both methods are useful to be used as decision support tool in practice to create rolling stock schedules in both the Netherlands and Denmark.

Instance	Obj	MC	SSC	SC	δ	Cover	B (s)	Columns	N (s)
NS	639 065	553 310	15 755	70 000	0	99.9%	465	14 340	19
DSBmon	719 184	555 970	132 214	31 000	0	98.5%	119	5 937	19
DSBfri	727 159	583 505	119 654	24 000	0	98.6%	37	3 627	12
DSBsat	418 148	313 469	87 679	17 000	0	98.3%	10	1 052	7
DSBsun	413 062	297 574	93 489	22 000	0	98.1%	4	1 266	2

Table 3.9: Scheduling Results. The columns respectively show the solved instance, the objective value, the mileage costs, the overall seat-shortage costs, the costs for performing shunting operations, the costs for negative differences between start and end inventories, the total seat cover percentage, computation times in seconds and columns generated using the B method, and finally the computation times for the N method.

3.5.3 – Rolling Stock Rescheduling

In this section we discuss the computational experiments for the rescheduling instances. For the instances we assume the demand to be the same as in the original situation. See, for instance, Kroon et al. (2014) for a paper that takes dynamic passenger flows into account by means of a simulation step in the Rolling Stock Rescheduling Problem. This aspect is outside the scope of the current paper. Note, however, that it is possible to use the approaches as discussed in the framework of Kroon et al. (2014) without any adaptations.

In this section, we first give a brief overview of the different rescheduling cases for Netherlands Railways (Section 3.5.3.1) and DSB S-tog (Section 3.5.3.2). Thereafter, the computational results are discussed (Section 3.5.3.3).

3.5.3.1 – Netherlands Railways

In the following we compare both models on different disruption scenarios from NS. Consequently, we have chosen 12 different scenarios, of which an overview is given in Table 3.10. The instances are sorted by the complexity of the problem. Note that a problem with many trips that have to be rescheduled is more complex than a problem with little trains that need to be rescheduled. The disruptions take place on the main parts of the network, causing one track to be blocked for at least three hours in both directions. Consequently, there is no railway traffic possible between the stations where the disruption takes place.

The parameter settings used for the NS rescheduling instances are given in Table 3.11. The largest penalty is still on cancelling a trip, this is the most important criteria. The penalties for carriage kilometers and seat-shortage kilometers are the same as in the scheduling case, because in practice there is no time available to determine the optimal balance for every rescheduling instance separately. Three new penalties are used in the rescheduling cases:

Case	Disrupted area	Time interval	Number of trips
1	Gd - Ut	07:00-10:00	2187
3	Gd - Ut	11:00-15:00	1662
5	Rtd - Gv	11:00-15:00	1662
8	Amf - Ut	11:00-15:00	1662
10	Gv - Ledn	11:00-15:00	1662
12	Asd - Ut	11:00-15:00	1662
2	Gd - Ut	16:00-19:00	1002
4	Rtd - Gv	16:00-19:00	1002
6	Ledn - Ut	16:00-19:00	1002
7	Amf - Ut	16:00-19:00	1002
9	Gv - Ledn	16:00 - 19:00	1002
11	Asd - Ut	16:00 - 19:00	1002

Table 3.10: Different disruption scenarios NS. The columns show the case numbers, the location of the disruption, the affected time-slot, and the number of trips needing to be rescheduled.

Shunting Unplanned, *Shunting Cancelled*, and *End Diff*. Shunting Unplanned denotes the cost for doing a shunting operation that was not originally scheduled. Every new (un)coupling action requires shunting crew members to perform them. As a consequence, the shunting crew needs to be rescheduled which costs time and consequently money. So, we want to minimize the number of additional shunting movements. On the other hand, Shunting Cancelled stands for the penalty for cancelling a scheduled shunting movement. Shunting crew members are instructed to do planned shunting movements during the day, cancelling such a shunting movement means that the tasks for those crew members have to be cancelled. This has to be communicated to the involved crew members. Therefore, we want to minimize the number of cancelled shunting operations as well. However, cancelling shunting movements is preferred over adding additional shunting movements, because arranging new shunting crew members for a task is usually more difficult than cancelling a shunting crew task. Finally, End Diff denotes the penalty for having a negative deviation from the scheduled end-of-day balance. This coefficient is kept identical to the one used in the planned instance for the start-end-of-day deviation.

3.5.3.2 – DSB S-tog

For the DSB S-tog instances, there are also 12 different disruption scenarios, see Table 3.12. These instances are sorted on their complexity. The disruptions occur either on the Monday or Friday instances, leading to 24 instances in total.

Coefficient	NS	DSB S-tog
Cancel	1 000 000	1 000 000
Mileage	1	0.1
Seat Shortage	0.5	0.5
Shunting Unplanned	1 000	1 000
Shunting Cancelled	100	100
End Diff	10 000	10 000

Table 3.11: Penalty values used for the NS and S-tog rescheduling cases.

Case	Disrupted Area	Time interval	Number of trips
6	Val-Fs	09:00-10:00	3550
5	Val-Fs	09:00-13:00	3500
12	Kh-Kk	09:00-10:00	3550
11	Kh-Kk	09:00-13:00	3500
3	Val-Fs	11:00-12:00	3000
4	Val-Fs	11:00 - 15:00	2970
9	Kh-Kk	11:00-12:00	3000
10	Kh-Kk	11:00-15:00	2970
1	Val-Fs	15:00 - 16:00	1920
2	Val-Fs	15:00 - 19:00	1870
7	Kh-Kk	15:00-16:00	1920
8	Kh-Kk	15:00-19:00	1870

Table 3.12: Different disruption scenarios DSB S-tog. The columns denote the different case numbers, the location of the disruption, the affected time-slot, and the number of trips needing to be rescheduled.

The same objective coefficients are used as in the NS rescheduling instances. The only difference is the mileage penalty, which is 0.1 instead of 1.0, because of the reasons mention in Section 3.5.2.2.

3.5.3.3 – Computational Results

All three approaches give the same results after rescheduling a disruption instance. Figure 3.6 gives an overview of the solution costs for the rescheduling results. Here, the objective value, the mileage costs, and the seat-shortage costs are displayed per rescheduled instance. When the difference between objective value and the sum of the mileage and seat-shortage costs is large, then at least one additional trip is cancelled during the rescheduling. The other costs are not shown, because they are very small in comparison. The objective value is largest in the cases where trips are cancelled. In the other cases, the objective value is close to the mileage costs. So, the mileage costs are much larger than the seat-shortage costs in all cases.

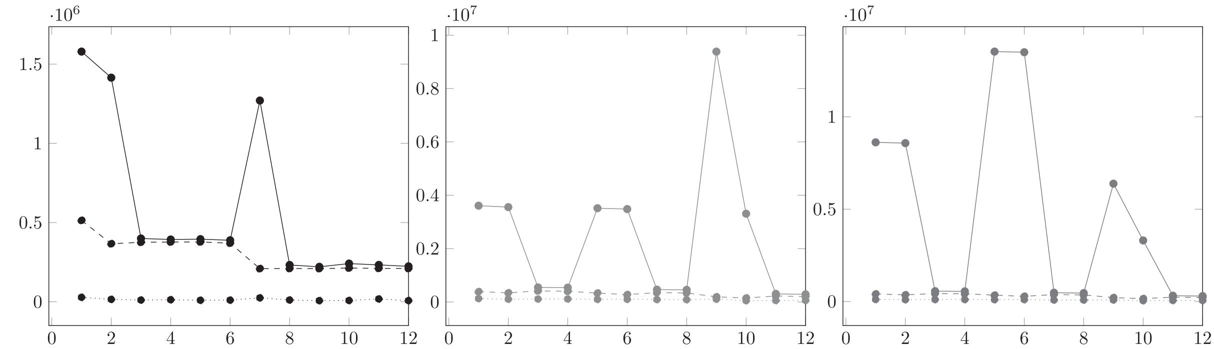


Figure 3.6: Results for the NS (left) and DSB S-tog (Monday middle, Friday right) rescheduling instances. A straight line denotes the objective value, a dashed line the mileage costs, and a dotted line the seat shortage costs. The horizontal axis denotes the case numbers and the vertical axis the values.

This can also be noted by Figure 3.7, where the total seat coverage percentage per rescheduling instance is shown. The seat coverage is always between 97% and 99.9%, which means that only few passengers do not get a seat in the train. As a consequence, the penalty for seat-shortages is small. These results are desired in practice, because, next to not cancelling trips, keeping a good passenger service is one of the important characteristics during rescheduling.

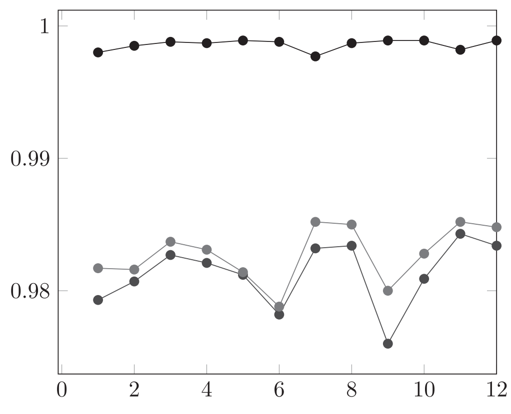


Figure 3.7: The seat-coverage for the rescheduling cases. The black line is for the Dutch instances, the blue line for the Friday DSB S-Tog instances, and the red line for the Monday DSB S-Tog instances. The horizontal axis denotes the case numbers and the vertical axis the seat cover percentage, where 1 stands for 100%.

Besides finding good results, the computation time is of importance as well in real-time. In a disrupted situation, little time is available to reschedule the rolling stock. Therefore, we will compare our models based on their computation times. In Table 3.13 an overview of all computation times required to solve the different instances with the different models is shown. First of all, the solution method B is substantially faster than the solution method A. However, both models are considerably slower than method N. On the other hand, as explained before, the Path Based Model has other advantages over the Composition Model.

We note that the computation times are influenced by the start time of the disruption - a late disruption involves fewer trips than an earlier disruption, thus a late disruption is easier to solve. We assume that solutions have to be found within 5 minutes in order to be useful in real-time when a disruption occurs. In conclusion, both methods B and N are applicable for usage in practice. They are able to to produce good results in relatively short time, consequently the rescheduled circulation can be used by practitioners in real-time. On the other hand, too high computation times are observed in some cases for the A approach.

	NS			DSBmon			DSBfri		
Case	A	B	N	A	B	N	A	B	N
1	722	269	5	286	207	8	303	40	7
2	138	23	5	176	31	4	151	22	6
3	133	34	3	156	9	3	235	15	6
4	159	39	3	222	17	7	284	15	4
5	157	26	3	72	15	5	113	17	5
6	163	35	3	63	23	5	51	7	3
7	20	6	2	62	6	3	96	10	4
8	11	3	1	106	18	4	103	8	3
9	14	2	1	38	22	4	18	7	3
10	13	13	2	4	2	2	5	2	2
11	10	2	1	8	2	1	13	4	2
12	15	2	1	6	2	2	9	2	2
Average	130	38	3	100	29	4	115	12	4

Table 3.13: Computation time (in seconds) for solving the disruption instances.

3.5.4 – Delayed Transition Constraint Generation

Finally, in this section we present details of the average computation time and average number of columns and rows of the A and B methods in Table 3.14. The results show that the dynamic row generation B method is on average around 5 times faster than the normal column generation method A.

	A			*B*			*N*
Case	Time	Columns	Rows	Time	Columns	Rows	Time
NS	190	6 178	7 214	51	6 434	3 573	2
DSBmon	143	7 321	6 523	31	4 701	5 677	4
DSBfri	161	7 040	6 891	16	4 373	4 704	4

Table 3.14: The columns show computation time (in seconds), number of columns and rows generated for solving planning and disruption instances. Each row shows the average over 1 planning case and 12 disruption cases.

Interestingly, the table shows that, on average, the number of generated columns decreases in B for the DSB cases, but increases slightly for the NS cases in comparison with the A method. However, we observe a consistent decrease in the number of rows.

Although the B method often results in fewer generated columns and rows than the A method, there are cases (considering the non-aggregated results) where this is not the case. There exist cases where the runtime of the B method is improved in comparison with the A method even if there are more columns and rows are generated in total in the B method - we believe this is due to the better convergence property of the B method, see Section 3.4.2. Consequently, we conclude that the B method performs better for (re)scheduling the rolling stock when the order of the train units within a composition are of importance.

3.6 – CONCLUSION

In this paper a comparison is made between two rolling stock (re)scheduling models by testing them on data sets from two countries (the Netherlands and Denmark). The results show that the considered approaches are not limited or tailored to specific networks. Furthermore, it is the first time that the Composition Model is tested on instances of the DSB S-tog network in Denmark and the Path Based Model on instances from Netherlands Railways.

In order to schedule a rolling stock circulation that can be used in practice, a sensitivity analysis of the two models is carried out to determine reasonable values for the mileage and seat-shortage penalties for the train services. The results demonstrate that a higher seat-cover requires significantly more kilometers to be carried out by the carriages. In practice, the operator can decide which ratio between seat-shortages and carriage kilometers suits them best. For the NS instances it was possible to provide a seat for all passengers, but for the DSB S-tog case it was, with the different penalties we have tested, not possible to provide a seat for all passengers. This is due to the fact that the passenger demand is sometimes larger than the capacity of the largest composition we can appoint.

In the current tests the column generation approach of the Path Based Model has longer computation times than the Composition Model. However, the computation times of the path based model are faster than most models from existing literature that are able to take unit specific constraints into account (see for instance Borndörfer et al. (2015), Cacchiani et al. (2010), and Wagenaar et al. (2016) (see Section 4)). Therefore, we believe that the column generation approach can be quite interesting, especially when we want to include unit specific constraints, e.g. due to maintenance. Furthermore, we can conclude that adding dynamic row generation to the column generation approach significantly reduced the computation time.

Next to the scheduling instances, we have tested both approaches on rescheduling instances in the Netherlands and Denmark. Disruption instances are smaller than the planning counterparts since all trip decisions up to the start of the disruption are fixed. So in some sense they are easier. However, since we also want to minimize the deviations from the original plan, they are also harder. The experiments show that both models are able to reschedule the rolling circulation fast enough to be used in real-time in both countries.

The proposed approach is of high practical relevance for railway operators. As explained before, in practice there is very limited time available to reschedule the rolling stock during a disruption. Currently, this is still done manually in most of the countries. There are two advantages by using a decision support system that is able to reschedule the rolling stock automatically: 1) The process can be speeded up drastically and 2) The solution quality is most likely much better (i.e. note that our solution approach always provides the optimal solution, while it is doubtful whether the dispatchers are able to find this solution manually).

In future research we want to consider unit specific constraints for the rolling stock, in both a planning and a disruption context. Examples of interesting additional constraints are maintenance appointments and minor rolling stock defects. In the former case units have to reach a maintenance facility in time for an appointment, and in the latter case units with a small defect have to comply with other unit-specific constraints.

4 – Maintenance Appointments in Railway Rolling Stock Rescheduling

This chapter considers the paper (Wagenaar et al. (2016)) which is accepted in Transportation Science. In 2013 a preliminary version of this paper has won the first place at the Trail Best Paper Award and in 2014 a preliminary version has been granted a third place in the Student Paper Award Competition of the Railway Application Section of INFORMS

Co-authors: L.G. Kroon and M. Schmidt

4.1 – INTRODUCTION AND CONTRIBUTIONS

4.1.1 – Introduction

In passenger railway transportation, an extensive planning process is used to develop a satisfying rolling stock circulation. First, in the *strategic* planning phase, the purchase of rolling stock takes place and specific train lines are determined. A train series represents a line between stations A and B and back again, with possible intermediate stations. In the *tactical* planning phase, the timetable is created and train units are assigned to all trips within the timetable. This results in a rolling stock circulation, usually involving anonymous rolling stock duties, i.e. no physical train units have been assigned to the rolling stock duties yet.

In the *operational* planning phase, physical train units are assigned to the anonymous duties. Furthermore, the rolling stock circulation is modified by taking into account specific operational aspects, such as the short-term maintenance that is required by certain physical train units. A train unit requires maintenance after a certain number of kilometers or a certain amount of time since its previous maintenance appointment. A train unit requiring maintenance gets a fixed maintenance appointment, assigned by the maintenance company, at a given time and location. In the operational planning phase, the rolling stock circulation obtained in the tactical planning phase is modified in such a way that the maintenance appointments are met by the corresponding physical train units. The latter means that they arrive at the appropriate locations on time, see for instance the maintenance routing models of Maróti and Kroon (2005, 2007).

In the *real-time* phase, the railway network inevitably experiences disruptions and therefore fast rescheduling is required. There are three major resource schedules which need to be rescheduled due to a disruption: The timetable, the rolling stock circulation, and the crew schedule. In the Netherlands, the plan is usually to first reschedule the timetable based on a predefined contingency plan. Then, with the rescheduled timetable as input, the rolling stock is rescheduled, and, finally, with both the rescheduled timetable and rolling stock circulation as input, the crew is rescheduled. However, in practice it could be chaotic and all rescheduling steps are done disorderly. Therefore, a great need is required for computerized effort during rescheduling.

In this paper the focus is on the second step: The Rolling Stock Rescheduling Problem(RSRP). We assume that the timetable has already been rescheduled based on a contingency plan. Given the rescheduled timetable, the RSRP aims to find a new feasible rolling stock circulation that upholds as much of the passenger service as possible. It is required that the fixed maintenance appointments are taken into account directly in the rescheduling process.

However, current rolling stock rescheduling models, see for example Nielsen (2011), assign anonymous train units to the trips during a disruption. They assume that all train units of the same type are interchangeable. That means, for instance, that there is no distinction between train units that require maintenance and train units that do not. As a result, if maintenance appointments are not taken into account, then the train units scheduled for maintenance will probably not be in time for their maintenance appointments. Thus, when rescheduling the rolling stock, the maintenance appointments of the train units must be considered.

In this paper, three MIP formulations for solving additonally constrained multi-commodity flow problems are used as solution method for this problem. The maintenance appointments that have been scheduled in the operational planning phase are taken into account in these model formulations. The developed models are able to reschedule the rolling stock in real-time such that the maintenance appointments are still met by the corresponding train units as much as possible.

4.1.2 – Contributions and structure of the paper

Although there exist models for rescheduling the rolling stock circulation in the operational phase including maintenance appointments, the current paper is, to the best of our knowledge, the first to include maintenance appointments in the real-time rescheduling phase. In other words, existing papers schedule maintenance while in this paper existing maintenance appointments are taken into account. By including these maintenance appointments in the RSRP models, the models are able to guide the maintenance units to their scheduled maintenance appointments after the occurrence of a disruption.

The main contribution of this paper is the development and comparison of three MIP models which are able to handle the complicating factor that physical train units of the same type are not fully interchangeable due to their maintenance appointments.

The contributions of the current paper can be summarized as follows:

- We take scheduled maintenance appointments into account while rescheduling the rolling stock, with the rescheduled timetable as input.
- We describe one straightforward extension of an existing model and introduce two new models.
- We provide an experimental comparison of the three models.

The paper begins in Section 4.2 with explaining the maintenance problem in detail. Then a literature overview is given in Section 4.3. Thereafter the Composition model from Fioole et al. (2006) and Nielsen (2011) for rescheduling the rolling stock without maintenance appointments is presented. This model is used as the base model for all three models that take maintenance into account. The notation used for describing the maintenance aspects is explained in Section 4.5.

Following, three approaches for including maintenance in rolling stock rescheduling models are given. Firstly, the Extra Unit Type model is discussed in Section 4.6. Secondly, the Shadow-Account model is presented in Section 4.7. Finally, the Job-Composition model is proposed in Section 4.8. Then, in Section 4.9, all models are tested on real life instances of Netherlands Railways (NS), the main operator of passenger trains in the Netherlands. All models use the same objective function. Therefore, we mainly compare the models with respect to their computation time and the number of times a proven optimal solution is found within a certain time limit. In Section 4.10, conclusions and topics for further research are given.

4.2 – MAINTENANCE PROBLEM

In this section the maintenance problem is explained in detail. We start with some general remarks on rolling stock scheduling. Thereafter we present an example of the maintenance problem. Finally we discuss the assumptions that are taken into account in this paper.

Rolling stock units of different types are available for passenger transportation. There exist large differences between the different types. First, there exist self-propelled train units and carriages hauled by a locomotive. In this paper we focus on self-propelled train units.

Furthermore, there exist train units with two floors (called double-deck) and train units with a single floor. The main difference between the types we consider in this paper is the number of carriages of which they consist (e.g. a VIRM6 unit consists of 6 carriages and a VIRM4 unit consists of 4 carriages). See, for instance, Figure 4.1 for a train unit of type VIRM4.

Figure 4.1: Train unit of type VIRM4

Train units can be coupled onto each other before being used on a trip. Such an ordered combination of train units is called a *composition*. Compositions are used to assign sufficient capacity for the passenger demand on a trip. *Composition changes* can possibly take place at stations in order to increase or decrease the capacity assigned to the next trip. This is done by either coupling or uncoupling train units to and from the train. In the Netherlands it is predefined in the station rules whether a train unit is (un)coupled at the front or at the rear of a train at a station. In addition, the middle unit of a composition cannot be uncoupled. As a consequence, the order of the train units within a composition is important.

Figure 4.2 shows a time-space diagram of the scheduled rolling stock circulation on the 3000 series between the stations Nijmegen (Nm) and Den Helder (Hdr). Time is displayed on the horizontal axis and the stations are shown on the vertical axis. Every line between two stations represents a train unit assigned to the corresponding trip. Two (or more) lines close to each other form a composition of two (or more) train units.

This circulation is infeasible due to a disruption between Utrecht (Ut) and Amsterdam (Asd) from 09:00-11:00 indicated by the black rectangle. There are in total 25 train units available, where two train units require maintenance: one that starts in Alkmaar (Amr) with an appointment at 16:00 in station Nijmegen (indicated by a black line), and one that starts in Den Helder with an appointment at 22:00 in station Nijmegen (indicated by a dark grey line). Both units have a maintenance appointment that lasts for two hours, thereafter they are available for usage again. The maintenance appointments are visualized by a black and a grey triangle, representing their location and timing. Due to the disruption, the circulation needs to be rescheduled for the remainder of the day, such that the maintenance appointments are still met by the corresponding train units.

Figure 4.3 shows the solution after rescheduling. The modified timetable, rescheduled based on a contingency plan, is given as input while rescheduling the rolling stock. As can be seen in the figure, in the new timetable the trains turn in Amsterdam and Utrecht during the disruption, as is specified in the contingency plan. Furthermore, in the rescheduled rolling

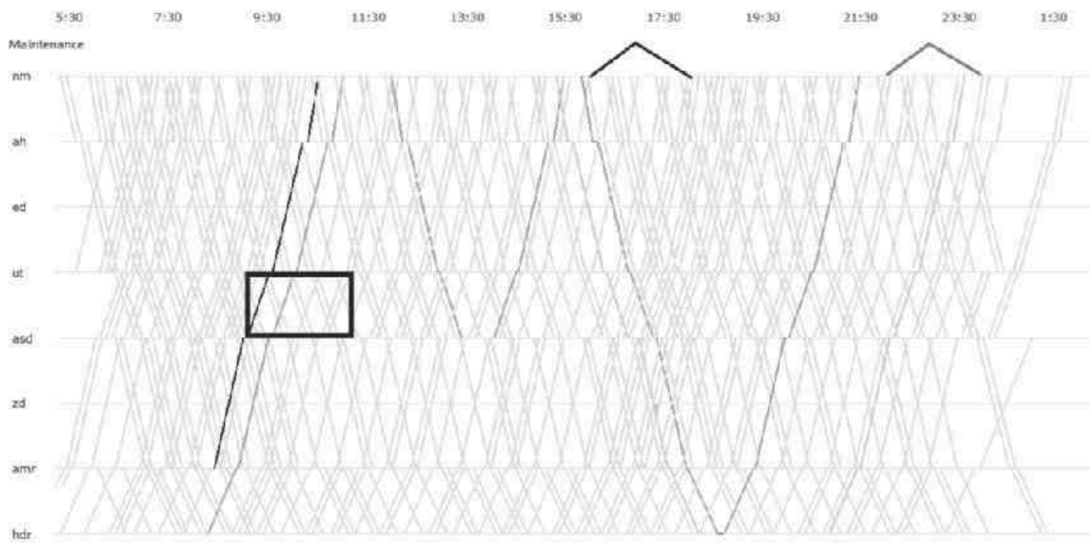


Figure 4.2: Time space diagram with a disruption

stock circulation, both units are still on time for their maintenance appointment. The models we discuss in this paper are able to reschedule the rolling stock in this way.

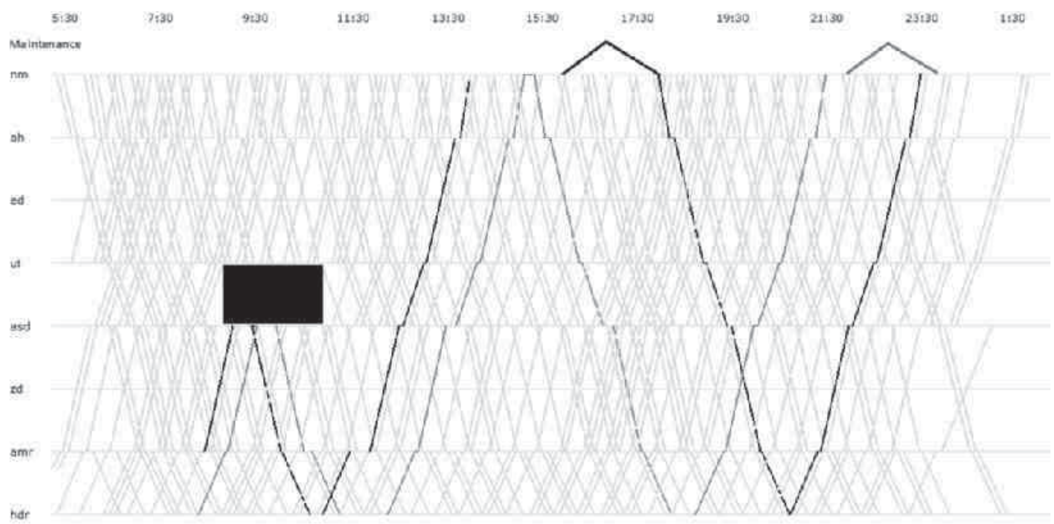


Figure 4.3: Rescheduled time space diagram

The problem is thus to guide certain individual train units in time to their maintenance appointment while rescheduling the rolling stock. This requires extensions of the existing rolling stock rescheduling models, since these models do not distinguish individual train units.

The assumptions that are taken into account in the developed models are the following:

1. The timetable has already been rescheduled and is used as input in the model for rescheduling the rolling stock.

2. The maintenance appointments of the train units have been fixed in the operational planning phase, and cannot be modified in the real-time rescheduling phase.
3. In the Netherlands, usually less than 5% of the train units have a maintenance appointment. Due to this fact, there is never more than one maintenance unit in a composition in real-life. Therefore we make the realistic assumption that at most one train unit requiring maintenance is present in each composition.
4. In the Netherlands, it is very unusual to both couple and uncouple train units from and to an arriving train. Usually, either a coupling activity, or an uncoupling activity, or no shunting activity at all takes place when a train arrives at a station. Therefore we assume that coupling and uncoupling at a station at the same time is not allowed. Note, however, that this assumption can be relaxed by slightly adjusting the models.
5. The passenger demand remains unchanged during a disruption. Including dynamic passenger flows in the rolling stock rescheduling problem is out of the scope in this paper. We refer to, for instance, Kroon et al. (2014) for a paper that focusses on dynamic passenger flows during a disruption.

4.3 – LITERATURE

Table 4.1 gives an overview of the literature related to rolling stock (re)scheduling. These papers can be classified based on two characteristics: models developed for either scheduling or rescheduling, and models where maintenance is included or not.

	Scheduling	*Real-time rescheduling*
No maintenance	• Fioole et al. (2006) • Cordeau et al. (2001) • Lingaya et al. (2002) • Brucker et al. (2003) • Mellouli and Suhl (2007) • Peeters and Kroon (2008)	• Nielsen (2011) • Nielsen et al. (2012) • Kroon et al. (2014) • Sato et al. (2009) • Sato and Fukumura (2012) • Haahr et al. (2015c)
Maintenance	• Maróti and Kroon (2005) • Maróti and Kroon (2007) • Giacco et al. (2014) • Borndörfer et al. (2015)	• Wagenaar et al. (2016)

Table 4.1: Overview railway literature

4.3.1 – Scheduling, no maintenance included

Fioole et al. (2006) formulate a MIP model to assign rolling stock to the timetable in the tactical planning phase. The model is called the *Composition model* and is an integer multi-commodity flow model with additional constraints. This model is solved by CPLEX. The model can handle complicated line structures, such as combining and splitting of trains. NS has been using this model to generate rolling stock schedules since 2004. The model takes the order of the train units in each composition into account. However, maintenance routing is out of its scope. Peeters and Kroon (2008) consider the same problem, but they describe a Branch & Cut approach as solution method.

Cordeau et al. (2001) describe the rolling stock scheduling problem as the routing of locomotives and carriages through a railway network. The locomotives and carriages have to be combined to form a train group which has to be routed through the network. Their problem focusses on the tactical planning phase. They do not take the order of carriages into account. The problem is modelled as an integer multi-commodity flow model and is solved with CPLEX. Similar problems are considered by Brucker et al. (2003) and Mellouli and Suhl (2007).

Lingaya et al. (2002) also study the problem of scheduling locomotives and carriages in the tactical planning phase. However, they do take the order of the carriages in a train into account. They consider a train as a Last-In-First-Out (LIFO) stack, where carriages can be coupled or uncoupled from the rear part of the train in LIFO order only.

4.3.2 – Scheduling, maintenance included

There are several papers that take maintenance into account in the operational phase. First of all, the problem was tackled in the airline industry before it was considered in the railway industry. For instance, Barnhart et al. (1998), Talluri (1998), and Clarke et al. (1997) propose models to solve the routing of maintenance for aircraft. Their models cannot be directly translated to models for scheduling rolling stock with maintenance constraints in the railway industry due to practical complications, such as the order of the train units in a composition.

One of the first to include maintenance routing in the operational phase of the railway industry were Maróti and Kroon (2005, 2007). They propose two different MIP formulations for maintenance routing of rolling stock for passenger trains: the "Transition Model" and the "Interchange Model". Both models use the scheduled rolling stock circulation as input, and exchange train unit duties such that maintenance requirements are met. Both models are designed for the operational planning phase.

Giacco et al. (2014) develop a MIP formulation for integrating maintenance planning in the rolling stock planning problem in the operational planning phase. Their formulation does not consider the order of the train units in a composition. Train units have to undergo maintenance after a certain time or a certain number of kilometers since their previous maintenance appointment. These maintenance appointments are not fixed, but determined by the model, in combination with the rolling stock circulation. A commercial MIP solver is used to find efficient solutions in short time. The model is tested on real-world instances of Trenitalia, the main Italian railway company.

Recently, Borndörfer et al. (2015) introduced a hypergraph formulation to create a rolling stock circulation for a generic week in a long distance railway network. The hypergraph formulation is used for the tactical planning phase. In this model several practical requirements are taken into account, such as scheduling of the maintenance for the train units. As in Giacco et al. (2014), the maintenance appointments are not fixed, but determined by the model. The model is tested on real life instances of the German railway company Deutsche Bahn. Circulations are found in between 10 minutes and 4 days of computation time. The model is not applicable for rescheduling in real-time due to computation time limitations.

4.3.3 – Real-time phase, no maintenance included

All of the above models are applicable in the tactical or operational planning phase of the railway process. Maintenance requirements are taken into account to schedule maintenance appointments for certain train units. During a disruption the rolling stock circulation becomes infeasible, but the train units requiring maintenance still have their appointment. Furthermore, during a disruption less time is available, and, as a result, fast models are required for rescheduling.

Cacchiani et al. (2014) give an extensive literature overview on recent research within passenger railway disruption management. Papers on rescheduling the timetable on microscopic and macroscopic level, rescheduling the rolling stock, and rescheduling the crew are discussed. We refer to this paper for all literature on timetable and crew rescheduling. In the current paper the focus is on rescheduling the rolling stock, so the remainder of the discussed literature is on rolling stock.

Nielsen (2011) extends the model of Fioole et al. (2006) to cope with rescheduling. He formulates a MIP model with the adjusted timetable and the original rolling stock schedule as input, and an adjusted rolling stock schedule as output. This model will be used as base model in the current paper and is referred to as the *Composition model*. Subsequently, Nielsen et al. (2012) propose a rolling horizon to solve the RSRP. The idea behind the rolling horizon is that at the beginning of the disruption not all information about the duration of the disruption is known: This information becomes gradually available. The rescheduling is periodically performed within a limited rolling horizon length, possibly taking new information into account. At each time instant where an updated timetable becomes available, or when a certain amount of time has passed without any update, the MIP model is solved for the rolling horizon time window. This model is tested on instances of NS. Solutions with small deviations from the original plan are found in a short time.

Kroon et al. (2014) consider real-time rescheduling of rolling stock during large disruptions while taking dynamic passenger flows into account. They use the rescheduled timetable as input. Then they apply a two-stage feedback loop, where in the first stage the rolling stock allocation is rescheduled by using the model of Nielsen (2011) and in the second stage the effect of the rolling stock allocation on the passenger flows is determined by means of simulation. This passenger simulation provides feedback in terms of passenger delays due to limited capacity of the assigned rolling stock. The feedback is then used in the optimization model to reallocate the rolling stock again, in such a way that the total passenger delay is reduced. Given the reallocation of the rolling stock, the passenger simulation is performed again and feedback is given to the optimization model. This process continues for a number of iterations. Results demonstrate that passenger delays can be reduced significantly.

Haahr et al. (2015c) (Chapter 3) focusses on rescheduling the rolling stock by using a path-based formulation instead of a flow-based formulation such as Nielsen et al. (2012). The advantage of this approach is that individual constraints could be set upon rolling stock units, however, this has not yet been tested. Results demonstrate that the approach is currently not as fast as flow-based approaches.

Sato et al. (2009) give a formulation to reallocate resources in a railway network in case of a disruption. Resources may refer either to rolling stock or to crew. The resources are reallocated to trips in such a way that the resource allocation differs as little as possible from the ones in the original plan. They use two phases to solve the problem. In the first phase conflicts created by the disruption are resolved through changes in the resource duties. The second phase is a local search heuristic which attempts to iteratively improve the rescheduled resource duties. The algorithm is tested on one line of the Japanese railway network.

In a subsequent paper, Sato and Fukumura (2012) consider the problem of reassigning locomotives to tasks in the case of a disruption in the railway network. A task consists of hauling a number of carriages from one station to another. They first enumerate possible sequences of tasks, to determine the corresponding costs for each sequence. A MIP model based on set-partitioning is used in order to assign locomotives to sequences of tasks with minimum cost, and a column generation technique is proposed as a solution approach. Based on the solutions found for instances of the Japan Freight Railway Company between Kuroiso and Shimonoseki in Japan, the authors conclude that locomotive reassignments can be found within a practical amount of time.

4.3.4 – Real-time phase, maintenance included

None of the above rescheduling models includes maintenance appointments. These appointments should be taken into account during rescheduling, otherwise the train units will most likely miss their appointments. To the best of our knowledge no rescheduling models exist that take maintenance appointments into account. In this paper we fill that gap in the existing literature.

4.4 – COMPOSITION MODEL

We start with introducing the base model. This is the Composition model developed by Fioole et al. (2006) and Nielsen (2011). As shown in Nielsen (2011), this model is fast enough to be used during rescheduling. However the model does not distinguish between train units of the same type. Therefore, the model does not include maintenance appointments of the rolling stock.

Let T be the set of trips in the timetable and S the set of stations. A trip is defined as a train driving from one station until the next station at a fixed point in time. Only stations where the composition of a train may be changed are taken into account. Denote $s_t^{dep}(s_t^{arr})$ as the station where trip $t \in T$ starts (ends) and define $\tau_t^{dep}(\tau_t^{arr})$ as the departure (arrival) time of trip $t \in T$.

In many countries, such as the Netherlands, trips are part of a predefined route. That means either that a trip has a predefined successor trip, or that the route ends after the trip. Take Figure 4.4 as an example of a predefined route between stations A, B, and C.

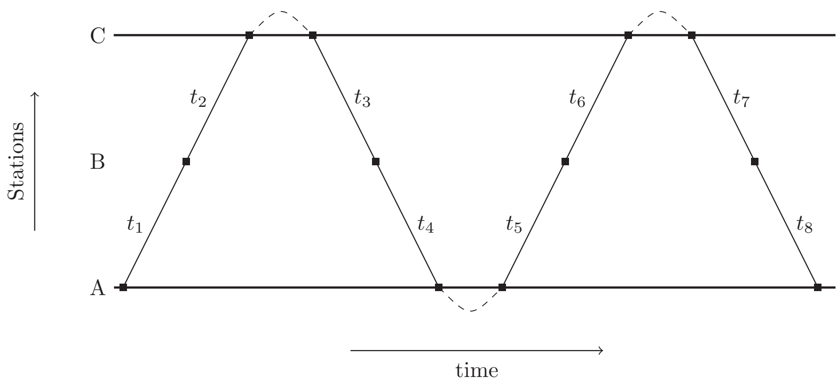


Figure 4.4: Predefined route

There are two trips between stations A and B, two trips between stations B and C, two trips between stations C and B, and two trips between stations B and A. Trip t_1 is the first trip of the route, so t_1 is not a successor trip of any other trip. Thereafter, we have that trip t_2 succeeds t_1, t_3 succeeds t_2, and so on. Trip t_8 does not have a successor trip, so the route ends after trip t_8. If no coupling or uncoupling takes place, it means that the successor trip uses exactly the same train units as its predecessor. If a trip does not have a successor trip, then this means that all train units are moved to the shunting yard.

As can be seen in Figure 4.4, also trips at an end station of a line can have a successor trip. For instance, trip t_3 succeeds trip t_2. In that case, the train units that will be used on the successor trip wait at the platform track where the trip arrives until the successor trip starts. A dashed line represents a turning between two succeeding trips.

We define $\sigma(t)$ as the successor trip of trip t, R as the set of routes, and $r := (t_1, \ldots, t_n)$ is a route consisting of a sequence of trips, such that t_1 does not have a predecessor, $\sigma(t_i) = t_{i+1}$ for all $i = 1, \ldots, n-1$ and $\sigma(t_n) = \emptyset$. Then, $r(t)$ is the (uniquely defined) route trip t belongs to.

Now, let M be the set of rolling stock types. We denote P as the set of possible compositions, where a composition is an ordered combination of train units that can be used on a trip. For example, in Figure 4.5 the composition ab is assigned to trip t and the composition a is assigned to trip $\sigma(t)$. For each trip $t \in T$, $P(t)$ denotes the set of allowed compositions on the trip. Note that for each trip the empty composition is an element of $P(t)$, meaning that each trip may be cancelled. However, cancelling a trip is highly undesirable.

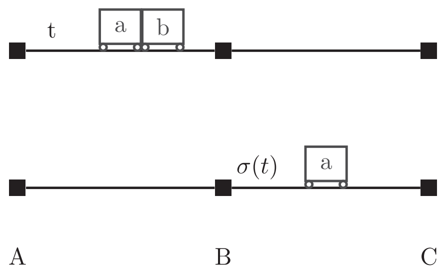


Figure 4.5: The successor trip of trip t

At the end of a trip, the composition of a train can be changed, depending on the shunting rules at the station, before the train departs on its successor trip. Recall that coupling and uncoupling takes place at either the front or the rear side of the train, this is defined by the station rules. A composition change denotes the composition of the incoming trip, the composition of the outgoing trip, and which train units are coupled or uncoupled during the composition change. To that end, let $\rho(t)$ be the set of possible composition changes at the end of trip $t \in T$, p_q the incoming composition of a trip when composition change $q \in \rho(t)$ is used, and o_q the outgoing composition when composition change q is used. For a given composition change $q \in \rho(t)$, $\alpha_{q,m}$ denotes the number of uncoupled train units of type $m \in M$ and $\beta_{q,m}$ denotes the number of coupled train units of type $m \in M$.

For instance, the composition change $ab \rightarrow a$ takes place at station B in Figure 4.5, so $p_q = ab$ and $o_q = a$. Furthermore, $\alpha_{q,a} = 0$, $\alpha_{q,b} = 1$, and $\beta_{q,m} = 0$ for both $m = a$ and $m = b$.

The time at which coupling takes place just before the start of trip $t \in T$ is denoted by τ_t^+ and the time at which an uncoupled unit is available after uncoupling after trip $t \in T$ is denoted by τ_t^-.

The available number of train units of type $m \in M$ at station $s \in S$ at the beginning of the planning period is denoted by $i_{s,m}^0$ and the desired number of available train units of type $m \in M$ at station $s \in S$ at the end of the planning period is given by the parameter $i_{s,m}^\infty$. This is usually the end of the day.

Besides the defined parameters, the model uses the following decision variables:

- $X_{t,p} \in \{0, 1\}$ denotes whether composition $p \in P(t)$ is used on trip $t \in T$.
- $Z_{t,q} \in \{0, 1\}$ denotes whether composition change $q \in \rho(t)$ is used at the end of trip $t \in T$.
- $I_{t,m} \in \mathbb{Z}_0^+$ denotes the number of train units of type $m \in M$ in the inventory at station s_t^{dep} immediately after time τ_t^+.
- $C_{t,m}$ and $U_{t,m} \in \mathbb{Z}_0^+$ denote the number of train units $m \in M$ that are coupled and uncoupled at the start and end of trip $t \in T$, respectively.
- $D_{s,m} \in \mathbb{Z}$ denotes the deviation from the desired end-of-day balance at station $s \in S$ for rolling stock type $m \in M$.

Model:

$$\min f(X, Z, D) \tag{4.1}$$

subject to:

$$\sum_{p \in P(t)} X_{t,p} = 1 \qquad \forall t \in T \tag{4.2}$$

$$X_{t,p} = \sum_{q \in \rho(t): p_q = p} Z_{t,q} \qquad \forall t \in T, p \in P(t) \tag{4.3}$$

$$X_{\sigma(t),p} = \sum_{q \in \rho(t): o_q = p} Z_{t,q} \qquad \forall t \in T, p \in P(\sigma(t)) \tag{4.4}$$

$$C_{\sigma(t),m} = \sum_{q \in \rho(t)} \beta_{q,m} Z_{t,q} \qquad \forall t \in T, m \in M \tag{4.5}$$

$$U_{t,m} = \sum_{q \in \rho(t)} \alpha_{q,m} Z_{t,q} \qquad \forall t \in T, m \in M \tag{4.6}$$

$$i^{\infty}_{s,m} + D_{s,m} = i^{0}_{s,m} - \sum_{\substack{t \in T \\ s^{dep}_t = s}} C_{t,m} + \sum_{\substack{t \in T \\ s^{arr}_t = s}} U_{t,m} \qquad \forall s \in S, m \in M \tag{4.7}$$

$$I_{t,m} = i^{0}_{s,m} - \sum_{t' \in A_t} C_{t',m} + \sum_{t' \in B_t} U_{t',m} \qquad \forall t \in T, m \in M, s \in S : s = s^{dep}_t \tag{4.8}$$

$$X_{t,p} \in \{0, 1\} \qquad \forall t \in T, p \in P(t) \tag{4.9}$$

$$C_{t,m}, U_{t,m}, I_{t,m} \in \mathbb{R}_+ \qquad \forall t \in T, m \in M \tag{4.10}$$

$$D_{s,m} \in \mathbb{R}_+ \qquad \forall s \in S, m \in M \tag{4.11}$$

$$Z_{t,q} \in \mathbb{R}_+ \qquad \forall t \in T, q \in \rho(t) \tag{4.12}$$

We start with explaining the objective function (4.1). The objective function is a linear function based on the appointed compositions to a trip, the applied composition changes just after a trip, and the end-of-day balances at stations.

In order to define the objective function with respect to the appointed compositions, we introduce the parameters p_0, $F^{ss}_{t,p}$, and $F^{carr}_{t,p}$. These parameters represent the empty composition $p_0 \in P$, the number of seat-shortage kilometers when using composition p on trip $t \in T$, and the number of carriage kilometers when using composition p on trip t. With these additional parameters we can define the first part of the objective function as in Equation (4.13). Here, a penalty value of κ is used for each cancelled trip. Note that a trip is cancelled if the appointed composition is the empty composition. Furthermore, a penalty value of Δ is used for each seat-shortage kilometer. Finally, a penalty value of ξ is used for each carriage kilometer.

$$\sum_{t\in T}\left(\sum_{\substack{p\in P(t)\\ p=p_0}}\kappa X_{t,p}+\sum_{p\in P(t)}X_{t,p}(\Delta F^{ss}_{t,p}+\xi F^{carr}_{t,p})\right) \tag{4.13}$$

The second part of the objective function refers to the number of additional shunting movements in comparison with the original circulation. Each additional shunting movement gets a penalty of ϕ. To measure the number of additional shunting movements, we define the parameter $s^{add}_{t,q}$. This parameter represents whether composition change $q \in Q$ is an additional shunting movement after trip $t \in T$ compared to the original composition change taking place after trip t. The number of additional shunting movements is then penalized in the second part of the objective function, as shown in Equation (4.14).

$$\sum_{t\in T}\sum_{q\in Q}\phi Z_{t,q}s^{add}_{t,q} \tag{4.14}$$

The final part of the objective function penalizes the deviation from the end-of-day balances, as shown in Equation (4.15).

$$\sum_{s\in S}\sum_{m\in M}\varrho D_{s,m} \tag{4.15}$$

The complete objective function can then be expressed as in Equation (4.16).

$$\sum_{t\in T}\left(\sum_{\substack{p\in P(t)\\ p=p_0}}\kappa X_{t,p}+\sum_{p\in P(t)}X_{t,p}(\Delta F^{ss}_{t,p}+\xi F^{carr}_{t,p})+\sum_{q\in Q}\phi Z_{t,q}s^{add}_{t,q}\right)+\sum_{s\in S}\sum_{m\in M}\varrho D_{s,m} \tag{4.16}$$

Constraints (4.2) specify that to each trip exactly one composition is assigned, this composition is in the set of allowed compositions, $P(t)$, of that trip. Note that the compositions of the trips before and at the start of the disruption are fixed, because these trips are already underway. For those trips the set of allowed compositions consists of only a single composition. Constraints (4.3) state that if composition $p \in P(t)$ is assigned to trip $t \in T$, then only a composition that can originate from composition p can be assigned to the succeeding trip $\sigma(t)$. Constraints (4.4) state that if composition $p \in P(\sigma(t))$ is assigned to the succeeding trip $\sigma(t)$, then only a composition that fits with composition p can be assigned to trip $t \in T$.

Constraints (4.5) specify the number of coupled train units at the beginning of a trip and Constraints (4.6) specify the number of uncoupled train units at the end of a trip. Constraints (4.7) specify the end-of-day balance at a station plus the total deviation from the scheduled end-of-day balance. Their sum equals the initial inventory at the station ($i^0_{s,m}$), minus all units that have been coupled onto a train and plus all units that have been uncoupled from a train during the day. Constraints (4.8) keep track of the inventory of rolling stock type $m \in M$ at station s_t^{dep} immediately after the coupling time τ_t^+. The sets A_t and B_t are explained below in detail. This inventory equals the initial inventory at the station, minus all train units that have been coupled onto a departing train before time τ_t^+ (all train units that have been coupled at the start of the trips in the set A_t), and plus all train units that have been uncoupled from an arriving train before time τ_t^+ (all train units that have been uncoupled at the end of the trips in the set B_t). Finally, Constraints (4.9), (4.10), (4.11), and (4.12) specify the domain of the decision variables. Since $X_{t,p}$ is binary, all other variables can be defined as continuous variables, see Maróti (2006).

In Constraints (4.8), the subsets A_t and B_t are defined as:

1. $A_t = \{t' \in T : s_{t'}^{dep} = s_t^{dep}, \tau_{t'}^+ \leq \tau_t^+\}$
2. $B_t = \{t' \in T : s_{t'}^{arr} = s_t^{dep}, \tau_{t'}^- \leq \tau_t^+\}$

The set A_t contains all trips which depart from station s_t^{dep} before time τ_t^+. This is the set of trips to which train units may have been coupled from station s_t^{dep} up to (and including) the departure time of trip t. We take as an example Figure 4.5 showing 8 different trips $\{t_1, t_2, t_3, ..., t_8\}$. To explain the set A_t, we focus on trip t_8. Trip t_8 departs from station B. All trips that have departed from station B before and including trip t_8 are the trips t_2, t_4, t_6, and t_8. So, $A_{t_8} = \{t_2, t_4, t_6, t_8\}$.

Furthermore, the set B_t contains all trips which have arrived at station s_t^{dep} before time τ_t^+. This is the set of trips from which train units may have been uncoupled to the inventory of station s_t^{dep} up to the departure time of trip t. If we look at Figure 4.5 again, then all trips arriving at station B before the departure time of trip t_8 are the trips t_1, t_3, t_5, and t_7. So, $B_{t_8} = \{t_1, t_3, t_5, t_7\}$.

The output of the Composition model is a list of trips with compositions assigned to them. Note that these compositions can be decomposed into individual duties for train units in a postprocessing step, because an integer flow can always be decomposed into train unit valued path flows, see Ahuja et al. (1993). However, this does not guarantee that there exists a feasible individual duty for train units that have a maintenance appointment. This is because the Composition model assumes all train units of the same type $m \in M$ to be interchangeable. As a consequence, no distinction can be made between train units requiring maintenance and train units that do not require maintenance. Thus, individual maintenance constraints cannot be imposed on the train units requiring maintenance.

Therefore we describe in Sections 4.6, 4.7, and 4.8 three extensions of the Composition model that allow guiding individual train units to their maintenance appointments.

4.5 – MAINTENANCE NOTATION

The notation for maintenance units we use throughout this paper is the following. Let M' be the set of train units that require maintenance. Denote h_m as the time that train unit $m \in M'$ has its maintenance appointment, g_m as the duration of the appointment, and f_m as the location of the appointment. Furthermore, all maintenance units still belong to their regular rolling stock type (e.g. a train unit of type a that requires maintenance is still a train unit of type a). To that end, let $b_m \in M$ be the corresponding regular rolling stock type of train unit $m \in M'$. Finally, train units with a maintenance appointment at the same time, at the same location, and with the same corresponding regular type can have the same maintenance type $m \in M'$. Then a_m denotes the number of train units with the specific maintenance appointment.

Train units that require maintenance need to be in inventory at the right maintenance location and in time for their appointment. The inventory is measured immediately after the coupling time, τ_t^+, of every trip $t \in T$, see Constraints (4.8). Thus, to be able to measure the inventory at the maintenance station at relevant points in time, we introduce for each maintenance unit $m \in M'$ an additional set of trips T_m. This set T_m contains the following (dummy) trips:

(i): One trip t' with parameters: $\tau_{t'}^{dep} = \tau_{t'}^{arr} = \tau_{t'}^{+} := h_m$ and $s_{t'}^{dep} = s_{t'}^{arr} := f_m$.

(ii): One trip t'' with parameters: $\tau_{t''}^{dep} = \tau_{t''}^{arr} = \tau_{t''}^{+} := h_m + g_m$ and $s_{t''}^{dep} = s_{t''}^{arr} := f_m$.

(iii): For each trip $t \in T$ with $s_t^{arr} = f_m$ and $h_m \le \tau_t^{arr} \le h_m + g_m$, the set T_m contains one trip t^* with parameters: $\tau_{t^*}^{dep} = \tau_{t^*}^{arr} = \tau_t^{arr}$, and $s_{t^*}^{dep} = s_{t^*}^{arr} := f_m$, .

These trips are used to trigger the measurement of the inventory (i) just after the start of the maintenance appointment, (ii) just after the end of the maintenance appointment, and (iii) just after the arrival of a trip at the maintenance station at a time instant in between. The trips t^* may bring a maintenance unit at a too late point in time to its maintenance loaction. Since these trips are used only to measure the inventory, no composition may be assigned to these trips, so the set of allowed compositions $P(t)$ for $t \in T_m$ consists of only the empty composition.

4.6 – EXTRA UNIT TYPE MODEL

The Extra Unit Type (EUT) model is an extension of the Composition model. By adding an additional rolling stock type for every train unit that has a maintenance appointment, maintenance constraints can be imposed on such a train unit.

Consider the same example as in Section 4.2. There are 25 train units, this time 10 train units of type a and 15 train units of type b. There are again 2 train units that require maintenance, one of type a starting in Alkmaar with an appointment in Nijmegen at 16:00 and one of type b starting in Den Helder with an appointment in Nijmegen at 22:00. That means that the following rolling stock types are used in the EUT model: a (9 train units), b (14 train units), a' (1 train unit) and b' (1 train unit). So, two additional rolling stock types have been added to the model.

The train units that require maintenance are used to redefine the set M by adding the train units requiring maintenance: $M := M \cup M'$. Furthermore, we introduce the decision variable $A'_{t,m}$, to denote the number of train units of type $m \in M'$ that are not present at their maintenance location immediately after the coupling time τ_t^+ of a trip $t \in T_m$. Then, Constraints (4.17) denote that maintenance units need to be in inventory at the time of their appointment and during their appointment. Otherwise the decision variable $A'_{t,m}$ is equal to the number of train units of type $m \in M'$ that are not at their appointment immediately after time τ_t^+ for $t \in T_m$.

$$I_{t,m} + A'_{t,m} \geq a_m \qquad \forall m \in M', t \in T_m \tag{4.17}$$

As a result, a penalty value θ_t can be set upon train units missing their appointment completely and on being late for their appointment. The objective function (4.16) is extended with Equation (4.18) for missing maintenance appointments.

$$\sum_{m \in M'} \sum_{t \in T_m} \theta_t A'_{t,m} \tag{4.18}$$

Together with Constraints (4.17) and Constraints (4.2)-(4.7) this forms the EUT model.

A drawback of this approach is that, by taking additional rolling stock types into account, the number of possible compositions increases rapidly. As a result of Assumption 3. from Section 4.2 that a composition contains at most one maintenance unit, we have that the increase in the number of compositions by adding an additional rolling stock type due to maintenance appointments only depends on the number of regular rolling stock types and on the allowed composition length. Indeed, a composition of length k, measured in the number of train units, consists of at most 1 train unit that requires maintenance and at least $k-1$

regular train units that do not require maintenance. Denote n as the number of available regular rolling stock types. Adding one additional type leads thus to $n^{k-1} \cdot k$ new compositions of size k. The maximum length of a composition, measured in the number of train units, is denoted by c. Then, adding one additional train unit that requires maintenance leads to a maximum total increase in the number of compositions that is equal to:

$$\sum_{k=1}^{c}(n^{k-1} \cdot k) = \frac{c \cdot n^{c+1} - (c+1) \cdot n^{c} + 1}{(n-1)^2}$$
$$= \frac{(c(n-1)-1) \cdot n^{c} + 1}{(n-1)^2}$$

This is polynomial in n, since c is fixed. In the Netherlands c is usually not larger than 5 and n not larger than 3, without taking maintenance appointments into account. Figure 4.6 shows a 3D surface plot of the above formula for c up to 5 and n up to 3. As can be seen, the number of additionally required compositions grows rapidly in c and n.

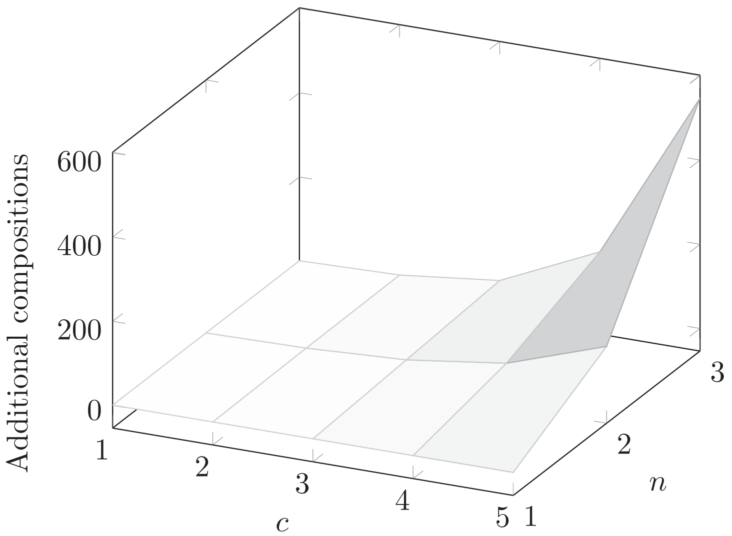


Figure 4.6: 3d plot of the number of additional compositions required

4.7 – SHADOW-ACCOUNT MODEL

To overcome the problem that the EUT model quickly grows when taking additional rolling stock types into account for every train unit that requires maintenance, we introduce the Shadow-Account (SA) model. We start with an introduction to the model in Section 4.7.1.

Thereafter, in Section 4.7.2, we explain the shadow account part. Finally, in Section 4.7.3, we explain the linking part.

4.7.1 – Introduction

The second approach to include maintenance in the rolling stock rescheduling problem is to create two parallel rolling stock circulations. The first circulation takes care of covering all trips with train units in the same way as in the Composition model. Thereby it does not make a distinction between train units requiring maintenance and train units that do not. The second circulation takes care of the maintenance appointments. To this end, a so called "shadow train unit" is created for each regular train unit. A shadow train unit is not denoted by a rolling stock type, (e.g. $a, b, \ldots$), but by a 'Shadow Account' type ('SA' type), (e.g. $0, a', b', \ldots$), representing maintenance appointments. A train unit with SA type 0 stands for a train unit that does not require maintenance and a train unit with SA type $a', b', \ldots$ stands for a train unit having a maintenance appointment. So, most train units are of SA type 0 and just a few train units have a different SA type. A train unit and its corresponding shadow train unit are synchronized, resulting in matching regular and shadow rolling stock circulations. For this reason the model is called the Shadow-Account model.

For instance, consider the same situation as in the previous section. There are 10 train units of type a and 15 train units of type b. The same 2 train units require maintenance, one of type a starting in Alkmaar with an appointment at 16:00 in Nijmegen and one of type b starting in Den Helder with an appointment at 22:00 in Nijmegen. In the Composition part of the model there are still 10 train units of type a and 15 train units of type b, however, in the shadow account there are 23 train units of SA type 0, one train unit of SA type a', and one train unit of SA type b'.

See Figure 4.7 for the corresponding composition circulation of the train units of the example. As can be seen, the composition circulation represents only the regular train units, it is not clear which train units have a maintenance appointment and which train units do not. On the contrary, in Figure 4.8 the SA circulation is visualized. In this circulation there is no distinction between train units that do not require maintenance. They are all represented by light gray lines. However, there is a distinction between train units that require maintenance (dark gray and black lines), so this circulation is specifically used to create paths to the maintenance appointments.

The two circulations have to match in, among other factors, terms of the lengths of the assigned rolling stock compositions, otherwise the maintenance paths cannot be used. The precise definition of the matching of the two circulations will be presented in Section 4.7.3.

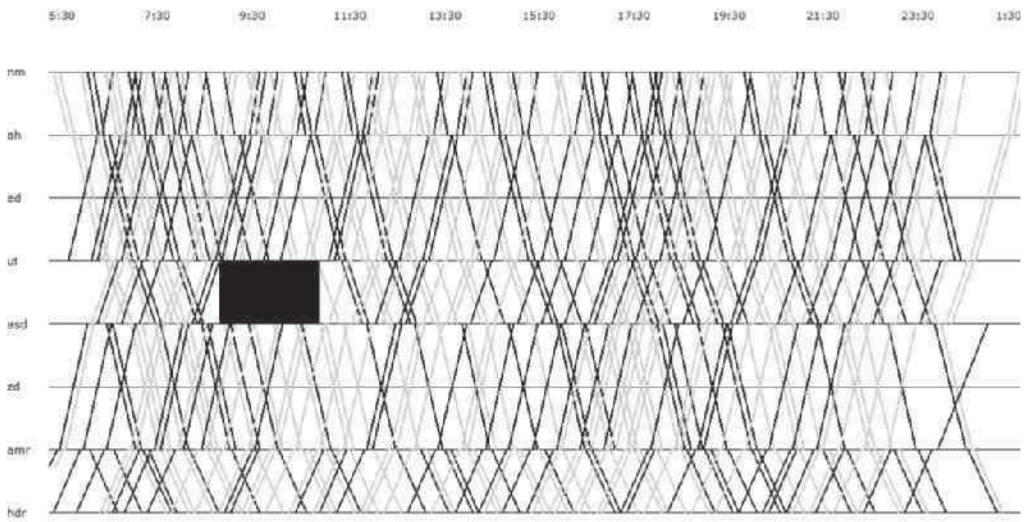


Figure 4.7: A solution for the Composition part, gray line = type a, black line = type b

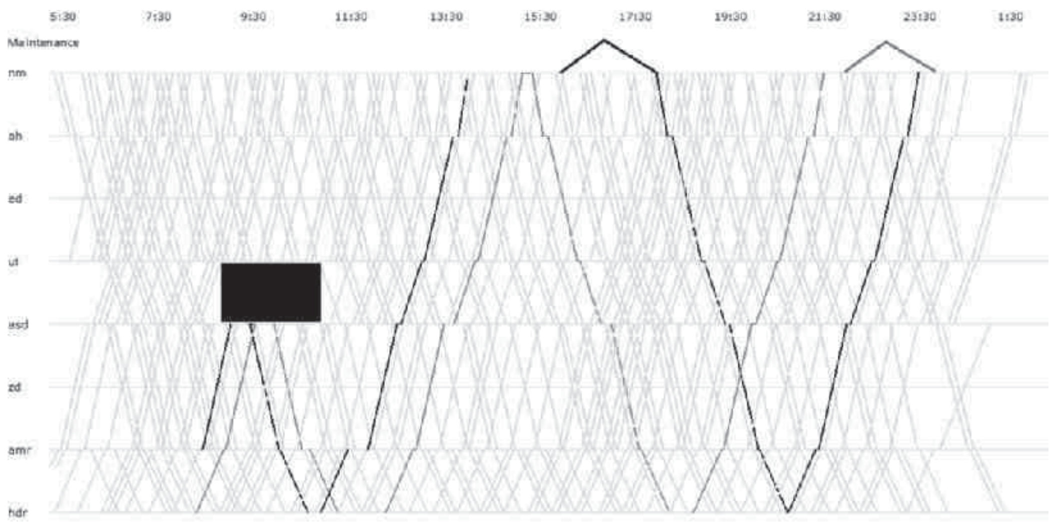


Figure 4.8: A matching solution for the SA part, light gray line = SA type 0, black line = SA type a', dark gray line = SA type b'

As in the previous section, we denote M as the set of different rolling stock types, and the maximum size of a composition is still denoted by c. Assuming that every SA composition contains at most one train unit that requires maintenance, one can verify that the total number of additionally required SA compositions after adding one train unit that requires maintenance equals $\sum_{k=1}^{c} k = \frac{1}{2}c(c+1)$. For example, a SA composition of length three (000) leads to three new SA compositions of length 3 ($a'00, 0a'0, 00a'$). The increase is quadratic in c and does not longer depend on n. For instance, with $c = 5$, there are only 15 additional SA compositions required, while there are 547 additional compositions required with the EUT model if $n = 3$ and $c = 5$.

The constraints in the complete SA model can be decomposed into three different parts: The Composition part, the SA part, and the Linking part.

All constraints in the Composition part are exactly the same as the constraints in the Composition model described in Section 4.4. So, the Composition part consists of constraints (4.2)-(4.7). The other parts will be discussed in the coming subsections.

4.7.2 – Shadow-Account part

The SA part creates a rolling stock circulation for the shadow train units. To that end, define the set M' to be the set of SA types $\{0, a', b', \ldots\}$ and P' as the set of possible SA compositions. The SA part of the MIP model consists of a second copy of Constraints (4.2)-(4.7), for the shadow types M' instead of the regular rolling stock types. The same kind of variables are used as well, e.g. the variable $X'_{t,p}$ states whether composition $p \in P'$ is used on trip $t \in T$.

We note that for the start inventory $i'^{,0}_{s,m}$ of the shadow types $m \in M'$ it holds by definition that the total number of SA train units in inventory at the start of the day is equal to the total number of regular train units in inventory at the start of the day.

$$\sum_{m \in M'} i'^{,0}_{s,m} = \sum_{m \in M} i^{0}_{s,m} \qquad \forall s \in S \tag{4.19}$$

This is not a constraint, but a condition that is to be satisfied by the data.

In a similar way as in the EUT model, we introduce the decision variable $A'_{t,m}$ denoting the number of train units of type $m \in M'$ that are not in inventory at their corresponding maintenance location at time τ_t^+ for $t \in T_m$. Then, Constraints (4.20) specify that either a maintenance unit is present at the station where its maintenance appointment is scheduled at the time of the appointment for the duration of the appointment, or the train unit is too late or misses its appointment completely. Constraints (4.20) are only needed for maintenance units $m \in M'$. As a consequence, the restriction $m \neq 0$ is used in the constraint set. The objective function (4.16) is then extended with Equation (4.18), just as in the EUT model.

$$I'_{t,m} + A'_{t,m} \geq a_m \qquad \forall m \in M' : m \neq 0, t \in T_m \tag{4.20}$$

Constraints (4.20) are added to the SA copy of Constraints (4.2)-(4.7) to form the SA part.

4.7.3 – Linking part

The Composition part creates a rolling stock circulation for the regular train unit types (e.g. $a, b, \ldots$). The SA part creates a second rolling stock circulation for the SA train unit types (e.g. $0, a', b', \ldots$). The SA part should give a shadow account of the Composition part. That means that the SA part should be linked to the Composition part.

Before introducing what we exactly mean by linking the SA and the Composition part, we first introduce the parameter N_p as the total number of train units in composition $p \in P$. Then the SA part is said to be linked to the Composition part if the following conditions are satisfied:

1. For each trip $t \in T$, the lengths of the compositions assigned to trip t in the Composition part and in the SA part are the same: $\sum_{p \in P(t): N_p = v} X_{t,p} = \sum_{p \in P'(t): N_p = v} X'_{t,p} \quad \forall t \in T, v \in \{0, 1, \ldots, c\}$
2. For each trip $t \in T$, the numbers of uncoupled train units at the end of trip t are the same in the Composition part and in the SA part: $\sum_{m \in M} U_{t,m} = \sum_{m \in M'} U'_{t,m} \quad \forall t \in T$
3. For each trip $t \in T$, the numbers of coupled train units at the start of trip t are the same in the Composition part and in the SA part: $\sum_{m \in M} C_{t,m} = \sum_{m \in M'} C'_{t,m} \quad \forall t \in T$
4. At each point in time, the numbers of regular and SA train units in inventory are the same in the Composition part and in the SA part: $\sum_{m \in M} I_{t,m} = \sum_{m \in M'} I'_{t,m} \quad \forall t \in T$
5. If a train unit of SA type $m \in M' : m \neq 0$ is assigned to trip $t \in T$ in the SA part, then its corresponding regular type $b_m \in M$ is assigned to trip t in the Composition part.
6. If at some point in time a train unit of SA type $m \in M' : m \neq 0$ is in inventory in the SA part, then its corresponding regular type $b_m \in M$ is in inventory in the Composition part.

Theorem 4.7.1. *After including Constraints* (4.21) *the variables* $U_{t,m}$ & $U'_{t,m}$, $C_{t,m}$ & $C'_{t,m}$, *and* $I_{t,m}$ & $I'_{t,m}$ *are linked in the way as stated in conditions* (1.), (2.), (3.), *and* (4.).

$$\sum_{p \in P(t): N_p = v} X_{t,p} - \sum_{p \in P'(t): N_p = v} X'_{t,p} = 0 \qquad \forall t \in T, v \in \{0, 1, \ldots, c\} \tag{4.21}$$

Proof: Constraints (4.21) are the same as condition (1.). Next we will prove in steps that conditions (2.), (3.) and (4.) are true as well if Constraints (4.21) are satisfied.

- $U_{t,m}$ and $U'_{t,m}$. We will show that $\sum\limits_{m \in M} U_{t,m} = \sum\limits_{m \in M'} U'_{t,m}$ for all $t \in T$ by contradiction. Assume that $\sum\limits_{m \in M} U_{t,m} > \sum\limits_{m \in M'} U'_{t,m}$ for at least one trip $t \in T$. This means that at the end of trip t more train units are uncoupled in the Composition part than in the SA part. By Constraints (4.21) the lengths of the compositions in the Composition part and in the SA part are the same for trip t. This holds for trip $\sigma(t)$ as well. It is assumed that more train units are uncoupled at the end of trip t in the Composition part. This is only possible if also more train units are coupled there, otherwise the lengths of the compositions assigned to trip $\sigma(t)$ in the Composition part and in the SA part are not equal anymore. However, it is not allowed to both couple and uncouple train units at the end of a trip. This leads to a contradiction, and so $\sum\limits_{m \in M} U_{t,m} \leq \sum\limits_{m \in M'} U'_{t,m}$.

  The same proof holds in the other direction, thus $\sum\limits_{m \in M'} U'_{t,m} \leq \sum\limits_{m \in M} U_{t,m}$. We can conclude that
$$\sum_{m \in M'} U'_{t,m} = \sum_{m \in M} U_{t,m} \qquad \forall t \in T$$
  .

- $C_{t,m}$ and $C'_{t,m}$. We can use the same proof as for $U_{t,m}$ and $U'_{t,m}$ to find that:
$$\sum_{m \in M'} C'_{t,m} = \sum_{m \in M} C_{t,m} \qquad \forall t \in T$$

- $I_{t,m}$ and $I'_{t,m}$. Assume that $\sum\limits_{m \in M} I_{t,m} > \sum\limits_{m \in M'} I'_{t,m}$ immediately after time τ_t^+ of at least one trip $t \in T$. The inventories at the start of the day are by definition equal on each station, see Equation (4.19), so a difference between $I_{t,m}$ and $I'_{t,m}$ arises during the operations. Note that, from Constraints (4.8), we have that:
$$\begin{aligned}\sum_{m \in M} I_{t,m} &= \sum_{m \in M} \left(i^0_{s,m} - \sum_{t' \in A_t} C_{t',m} + \sum_{t' \in B_t} U_{t',m} \right) \\ &= \sum_{m \in M} i^0_{s,m} - \sum_{t' \in A_t} \sum_{m \in M} C_{t',m} + \sum_{t' \in B_t} \sum_{m \in M} U_{t',m} \end{aligned} \tag{4.22}$$

  and
$$\begin{aligned}\sum_{m \in M'} I'_{t,m} &= \sum_{m \in M'} \left(i'^{,0}_{s,m} - \sum_{t' \in A_t} C'_{t',m} + \sum_{t' \in B_t} U'_{t',m} \right) \\ &= \sum_{m \in M'} i'^{,0}_{s,m} - \sum_{t' \in A_t} \sum_{m \in M'} C'_{t',m} + \sum_{t' \in B_t} \sum_{m \in M'} U'_{t',m} \end{aligned} \tag{4.23}$$

This means that a difference between $I_{t,m}$ and $I'_{t,m}$ can only be caused by a difference in either the start inventory, $C_{t,m}$ or $U_{t,m}$, but we just showed that $\sum_{m \in M} C_{t,m} = \sum_{m \in M'} C'_{t,m}$ and $\sum_{m \in M} U_{t,m} = \sum_{m \in M'} U'_{t,m}$. So, it holds that

$$\sum_{m \in M} I_{t,m} = \sum_{m \in M'} I'_{t,m} \qquad \forall t \in T$$

■

Theorem 4.21 is used for the first four linking conditions. We now introduce the second set of linking constraints for Condition (5.). These are Constraints (4.24). To this end, denote $w_{i,p}$ (or $w'_{i,p}$) as the rolling stock type assigned to position $i \in \{1, \ldots, c\}$ in composition $p \in P$ (or $p \in P'$). Constraints (4.24) then state that when a train unit of SA type $m \in M'$ with $m \neq 0$ resides in a SA composition on position i, then a corresponding regular train unit $b_m \in M$ must reside on position i in the corresponding regular composition as well.

$$\sum_{\substack{p \in P' \\ w'_{i,p} = m}} X'_{t,p} \leq \sum_{\substack{p \in P \\ w_{i,p} = b_m}} X_{t,p} \qquad \forall t \in T, i \in \{1, \ldots, c\}, m \in M' : m \neq 0 \tag{4.24}$$

Finally, for condition (6.), we use Constraints (4.25): if a train unit of type $m \in M' : m \neq 0$ is in inventory in the SA part, then a train unit of type $b_m \in M$ must be in the regular inventory.

$$I'_{t,m} \leq I_{t,b_m} \qquad \forall t \in T,\ m \in M' : m \neq 0 \tag{4.25}$$

Constraints (4.21)-(4.25) take care of synchronizing the two circulations (the Composition part and the SA part). The complete SA model is hence given by the Composition part, the SA part, the Linking part, and the objective function.

4.8 – JOB-COMPOSITION MODEL

In this section the third model to take maintenance into account is introduced. This model is called the Job-Composition (JC) model. This model is based on the concept of *jobs*. At the beginning of a day, all train units are in inventory. During the day, each train unit is assigned to a certain departing trip and fulfills a number of successor trips until the train unit is uncoupled and becomes part of the inventory again. A job is such a sequence of succeeding trips between coupling and uncoupling. So, a job starts when a train unit is coupled to a trip, and the job ends when the train unit is uncoupled from a trip. Note that a train unit may carry out more than one job per day.

The problem now becomes to appoint both regular train units and maintenance units to jobs, while synchronizing the movements of the maintenance units with those of the corresponding regular train units, just as in the SA model. In this way no additional compositions have to be taken into account for every train unit having a maintenance appointment.

A complicating factor is that trains must be considered, to a large extent, as *double sided stacks*. That means that, at both sides of a train, train units can be coupled or uncoupled in principle only in Last-In-First-Out (LIFO) order. As a consequence, if a pair of jobs does not correspond with a correct order of couplings and uncouplings per side of the train, then one assigned train unit will block the other one when the latter is to be uncoupled. In such a case, the two jobs are called *incompatible*. In Section 4.8.1 we characterize the pairs of incompatible jobs, and in Section 4.8.3 we present constraints that prevent two incompatible jobs from being chosen at the same time.

4.8.1 – Jobs

In a preprocessing step, we create a list of all possible jobs during the day, and denote J as this set of possible jobs. Let $T(j)$ be the set of trips covered by job $j \in J$. Every job $j \in J$ has a start (and final) trip denoted by λ_j (γ_j). For all trips $t_1, \ldots, t_n \in T(j)$ we have that $\lambda_j = t_1$, $\sigma(t_i) = t_{i+1}$, and $\gamma_j = t_n$. As can be seen from this notation, every job $j \in J$ takes place on a route $r \in R$, where R denotes the set of routes as defined in Section 4.4. Recall that the length of a predefined route depends on the shunting rules at a station and on the maximum turnaround time, as was explained in Section 4.4. The longer the maximum turnaround time, the longer the route, and the more possible jobs exist. As a consequence, the JC model contains more decision variables in that case.

Along each route $r \in R$ runs a train v_r, which consists of the actual train units that are used on the trips within the route. Each physical train has two sides. For further convenience, from now on these sides are called the A-side and the B-side of the train. We define the A-side of train v_r to be the front side of the train at the first trip of its route $r \in R$. Then we denote ζ_t as the side of train v_r that is the front side during trip $t \in T$ in route $r \in R$, and ζ_t^{-1} as the rear side.

Between two succeeding trips on a route $r \in R$, turnings can take place. In Figure 4.4 a route consisting of trips $t_1, \ldots, t_8$ with 3 turnings is shown. When train v_r turns, its front and rear side are exchanged. Note that turnings only take place at stations between two trips, not during a trip itself. So, in order to keep track of which side is the front side during trip $t \in T$, we need to keep track of the number of turnings taking place in route $r(t)$ up to the start of trip t. To that end, denote h_t^{turn} as the number of turnings taking place in route $r(t)$ up to the start of trip $t \in T$. Then, ζ_t can be determined as in Equation (4.26).

$$\zeta_t = \begin{cases} A & \text{If } h_t^{turn} \text{ is even or } 0 \\ B & \text{Otherwise} \end{cases} \tag{4.26}$$

Coupling can take place before the start of trip $t \in T$. Recall that in the Netherlands it is predefined in the station rules whether a train unit is coupled to the rear or to the front of the train. Let η_t denote whether a train unit has to be coupled to the front ($\eta_t = 0$) or to the rear ($\eta_t = 1$) of the outgoing train on trip t. Either the A-side or the B-side can be the front side of the train, this depends on the number of turnings taking place up to the start of trip t. We define the coupling side $\omega_j \in \{A, B\}$ as the side where coupling takes place before the start of job $j \in J$. The coupling side is determined as in Equation (4.27).

$$\omega_j = \begin{cases} \zeta_{\lambda_j} & \text{If } \eta_{\lambda_j} = 0 \\ \zeta_{\lambda_j}^{-1} & \text{Otherwise} \end{cases} \tag{4.27}$$

At the end of trip t, a train unit can be uncoupled from the composition. Just as with coupling, it is predefined in the station rules, whether a train unit is uncoupled from the rear or from the front of the train. To this end, let η'_t denote whether a train unit has to be uncoupled from the front ($\eta'_t = 0$) or the rear ($\eta'_t = 1$) of the incoming train. Again, this can be the A-side or the B-side of the train, this depends on the number of turnings taking place up to trip t. We denote π_j as the side where uncoupling takes place at the end of job $j \in J$, called the uncoupling side. Turnings cannot take place during a trip, so the number of turnings until the end of a trip is equal to the number of turnings until the start of the trip. Therefore, π_j is defined as in Equation (4.28).

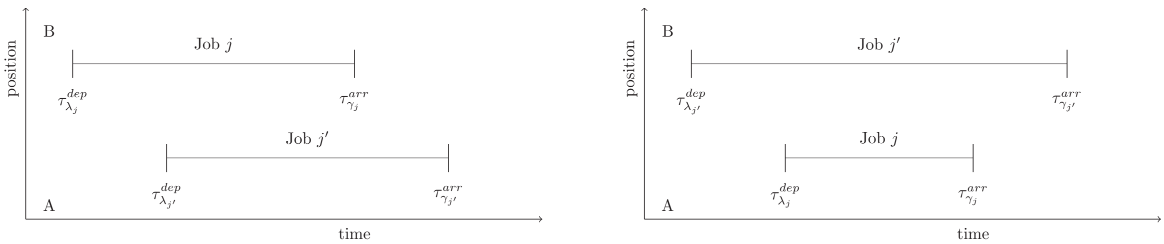
$$\pi_j = \begin{cases} \zeta_{\gamma_j} & \text{If } \eta'_{\gamma_j} = 0 \\ \zeta_{\gamma_j}^{-1} & \text{Otherwise} \end{cases} \tag{4.28}$$

Note that in case the coupling and uncoupling sides are not predefined by the station rules, we can adjust the model by taking them into account as variables in the model.

Definition 1. *A set of jobs $J' \subset J$ on route $r \in R$ is said to be compatible, if for every job $j \in J'$, the train unit assigned to job j, that is coupled at the start of trip $\lambda_j \in T$ with coupling side ω_j, can be uncoupled from its uncoupling side π_j after trip $\gamma_j \in T$ without being blocked by any other train unit assigned to a job $j' \in J'$.*

Lemma 4.8.1. *A set of jobs $J' \subset J$ is compatible if and only if for each pair of jobs j and $j' \in J'$ the following two conditions hold:*

1. *If $\tau^{dep}_{\lambda_j} < \tau^{dep}_{\lambda_{j'}} < \tau^{arr}_{\gamma_j} < \tau^{arr}_{\gamma_{j'}}$, then $\omega_{j'} \neq \pi_j$*
2. *If $\tau^{dep}_{\lambda_{j'}} < \tau^{dep}_{\lambda_j} < \tau^{arr}_{\gamma_j} < \tau^{arr}_{\gamma_{j'}}$, then $\omega_j = \pi_j$*



(a) First compatibility condition *(b) Second compatibility condition*

Figure 4.9: Compatibility conditions

Before proving Lemma 4.8.1, we first visualize the lemma's conditions in Figures 4.9a and 4.9b. Here time is displayed on the horizontal axis and the position of a train unit in the composition on the vertical axis. For convenience, we define the upper side of the figure as the B-side and the bottom side of the figure as the A-side of the train, assuming that no turnings take place. In Figure 4.9a the train unit assigned to job j' is coupled to the train unit assigned to job j at the A-side. Thereafter, the train unit assigned to job j is uncoupled from the train before the train unit assigned to job j' is uncoupled. The train unit assigned to job j can not be uncoupled from the A-side, because the train unit assigned to job j' is still there, so it must be uncoupled from the B-side. In Figure 4.9b the train unit assigned to job j is coupled to the train unit assigned to job j' and later uncoupled from the train unit assigned to job j' again. This is only possible if the train unit assigned to job j is uncoupled from the same side as where it was coupled.

Proof of Lemma 4.8.1: Assume that the set of jobs J' is compatible. Let j and j' be a pair of jobs in J'. First, suppose $\tau^{dep}_{\lambda_j} < \tau^{dep}_{\lambda_{j'}} < \tau^{arr}_{\gamma_j} < \tau^{arr}_{\gamma_{j'}}$. Since, by assumption, the uncoupling side of job j is not blocked by job j', we have that $\omega_{j'} \neq \pi_j$. Otherwise job j is being blocked. Second, suppose $\tau^{dep}_{\lambda'_j} < \tau^{dep}_{\lambda_j} < \tau^{arr}_{\gamma_j} < \tau^{arr}_{\gamma_{j'}}$. Again, the uncoupling of job j is not blocked by job j', so we must have that $\omega_j = \pi_j$. This completes the proof of the "only-if"-part of the lemma.

Next, suppose that each pair of jobs j and $j' \in J'$ satisfies the two conditions, and that the set J' is *not* compatible. Then, by definition there is at least one job $j \in J$, whose uncoupling after trip γ_j from its uncoupling side π_j is blocked by another job $j' \in J'$. Clearly, $T(j) \cap T(j') \neq \emptyset$ and $\tau_j^{arr} < \tau_{j'}^{arr}$. The latter follows from the fact that if the end times of the jobs are the same, then also their uncoupling sides would be the same. Thus job j' would not be blocking the uncoupling of job j in that case.

Furthermore, if $\tau_{\lambda_j}^{dep} = \tau_{\lambda_{j'}}^{dep}$, then, without loss of generality, we may assume that job j' is not blocking the uncoupling of job j. Otherwise the positions of jobs j and j' in the train could have been interchanged just before coupling. Thus we may assume that $\tau_{\lambda_j}^{dep} \neq \tau_{\lambda_{j'}}^{dep}$. That leaves us with the cases $\tau_{\lambda_j}^{dep} < \tau_{\lambda_{j'}}^{dep}$ and $\tau_{\lambda_{j'}}^{dep} < \tau_{\lambda_j}^{dep}$.

If $\tau_{\lambda_j}^{dep} < \tau_{\lambda_{j'}}^{dep}$, then we have $\tau_{\lambda_j}^{dep} < \tau_{\lambda_{j'}}^{dep} < \tau_{\gamma_j}^{arr} < \tau_{\gamma_{j'}}^{arr}$. Thus, by assumption, we have that $\omega_{j'} \neq \pi_j$. In addition, if $\tau_{\lambda_{j'}}^{dep} < \tau_{\lambda_j}^{dep}$, then we have that $\tau_{\lambda_{j'}}^{dep} < \tau_{\lambda_j}^{dep} < \tau_{\gamma_j}^{arr} < \tau_{\gamma_{j'}}^{arr}$. Thus, by assumption, we have that $\omega_j = \pi_j$. However, it is clear that in both cases job j' does not block the uncoupling of job j. This contradiction completes the proof of the lemma. ■

A set of jobs is not compatible if it contains a pair of jobs not fulfilling one of the two conditions in Lemma 4.8.1. As a result, we can add constraints to the model guaranteeing that there is no such pair of jobs selected by the model. So, no pair of jobs of the conflicting sets CJ^1 and CJ^2, described by Equations (4.29) and (4.30), may be chosen.

$$CJ^1 = \{(j, j') \in J \times J : T(j) \cap T(j') \neq \emptyset \wedge \tau_{\lambda_j}^{dep} < \tau_{\lambda_{j'}}^{dep} < \tau_{\gamma_j}^{arr} < \tau_{\gamma_{j'}}^{arr} \wedge \omega_{j'} = \pi_j\} \quad (4.29)$$

$$CJ^2 = \{(j, j') \in J \times J : T(j) \cap T(j') \neq \emptyset \wedge \tau_{\lambda_{j'}}^{dep} < \tau_{\lambda_j}^{dep} < \tau_{\gamma_j}^{arr} < \tau_{\gamma_{j'}}^{arr} \wedge \omega_j \neq \pi_j\} \quad (4.30)$$

4.8.2 – Further notation

During the whole day jobs are carried out by train units. At the moment a disruption occurs, there are jobs already being carried out by train units. Compositions of trips that have already departed at the start of the disruption cannot be changed. However, jobs can be changed, as long as the compositions assigned to the trips before the start of the disruption do not change. Denote the set of trips that have departed before the start of the disruption and that are still underway when the disruption starts by $T^< \subset T$ and set the parameter $G_{t,p}$ equal to 1 if composition $p \in P$ is assigned to trip $t \in T^<$.

Finally, the following additional decision variables are necessary in the JC model:

- $K_t \in \{0,1\}$ denotes whether trip $t \in T$ is cancelled or not.
- $W_j \in \{0,1\}$ denotes whether job $j \in J$ is selected or not.
- $Y_{j,m} \in \mathbb{Z}^+$ denotes the number of train units $m \in M$ assigned to job $j \in J$.
- $Q_{j,m} \in \{0,1\}$ denotes the number of *maintenance* units $m \in M'$ assigned to job $j \in J$.
- $I_{t,m} \in \mathbb{R}^+$ denotes the inventory of maintenance units $m \in M'$ at station s_t^{dep}, just after the departure of trip $t \in T$.
- $A'_{t,m} \in \mathbb{Z}^+$ denotes the number of maintenance units of type $m \in M'$ that are not available at their maintenance location immediately after time τ_t^+ for $t \in T_m$.

4.8.3 – Model

Constraints (4.31)-(4.43) are the constraints present in the JC model.

$$\sum_{j \in J: T(j) \ni t} W_j + K_t \geq 1 \qquad \forall t \in T \tag{4.31}$$

$$\sum_{m \in M} Y_{j,m} - W_j \geq 0 \qquad \forall j \in J \tag{4.32}$$

$$W_j + W_{j'} \leq 1 \qquad \forall (j,j') \in CJ^1 \cup CJ^2 \tag{4.33}$$

$$I_{t,m} = i^0_{s,m} - \sum_{t' \in A_t} \sum_{\substack{j \in J \\ \lambda_j = t'}} Y_{j,m} + \sum_{t' \in B_t} \sum_{\substack{j \in J \\ \gamma_j = t'}} Y_{j,m} \qquad \forall t \in T, m \in M, s \in S : s = s_t^{dep} \tag{4.34}$$

$$I_{t,m} = i^0_{s,m} - \sum_{t' \in A_t} \sum_{\substack{j \in J \\ \lambda_j = t'}} Q_{j,m} + \sum_{t' \in B_t} \sum_{\substack{j \in J \\ \gamma_j = t'}} Q_{j,m} \qquad \forall t \in T, m \in M', s \in S : s = s_t^{dep} \tag{4.35}$$

$$i^\infty_{s,m} + D_{s,m} = i^0_{s,m} - \sum_{\substack{j \in J \\ s^{dep}_{\lambda_j} = s}} Y_{j,m} + \sum_{\substack{j \in J \\ s^{arr}_{\gamma_j} = s}} Y_{j,m} \qquad \forall s \in S, m \in M \tag{4.36}$$

$$I_{t,m} + A'_{t,m} \geq a_m \qquad \forall m \in M', t \in T_m \tag{4.37}$$

$$Q_{j,m} - Y_{j,b_m} \leq 0 \qquad \forall j \in J, m \in M' \tag{4.38}$$

$$I_{t,m} - I_{t,b_m} \leq 0 \qquad \forall t \in T, m \in M' \tag{4.39}$$

$$W_j \in \{0,1\} \qquad \forall j \in J \tag{4.40}$$

$$Y_{j,m} \in \mathbb{Z}^+ \qquad \forall j \in J, m \in M \tag{4.41}$$

$$I_{t,m} \in \mathbb{R}^+ \qquad \forall t \in T, m \in M' \tag{4.42}$$

$$Q_{j,m} \in \{0,1\} \qquad \forall j \in J, m \in M' \tag{4.43}$$

Constraints (4.31) state that at least one job covers trip $t \in T$ or else the trip is cancelled. Every chosen job has to be performed by at least one rolling stock type $m \in M$, see Constraints (4.32). At most one job of each pair of jobs in the sets CJ^1 and CJ^2 can be chosen to be performed. This is modelled by Constraints (4.33).

Constraints (4.34) keep track of the inventory of all train units $m \in M$ just after the departure of trip $t \in T$, and Constraints (4.35) keep track of the inventory of *maintenance* train units $m \in M'$ just after the departure of trip $t \in T$. Then, Constraints (4.36) determine the end-of-day inventory of train units $m \in M$ at station $s \in S$. Furthermore, Constraints (4.37) state that every maintenance unit must be in inventory for the duration of its appointment and at the right location, or else the train unit was either too late or missed its appointment completely. Just as in the SA model, linking constraints are required between the maintenance units and the corresponding regular train units. If a maintenance unit $m \in M'$ is assigned to job $j \in J$, then its corresponding regular type $b_m \in M$ must also be assigned to job $j \in J$, see Constraints (4.38). The same holds for the inventory: if a maintenance unit of type $m \in M'$ is in inventory, then at least one of its corresponding train units of type $b_m \in M$ must also be in inventory, as is required by Constraints (4.39). Finally, Constraints (4.40)-(4.43) specify the domains of the variables.

Note that Assumption 4. from Section 4.2 that multiple maintenance units cannot occur in the same composition does not longer influence the size of the JC model. This is because the number of possible compositions does not depend on the number of maintenance units. However, we do not relax this assumption here, because we want to have comparable results for all three models.

4.8.4 – Composition part

The computational results demonstrated that including the constraints of the Composition model (Constraints (4.2)-(4.7)) speeds up the computation time of the JC model significantly. A possible explanation is that earlier research (Fioole et al., 2006) has shown that the Composition model is a tight model formulation leading to strong LP-bounds. Therefore including this part in the JC model speeds up the computation. Besides improving the computation time, using the constraints in the Composition model makes it easy to fix compositions on trips that have departed before the disruption starts. Therefore, we add Constraints (4.2)-(4.7) to the JC model, together with Constraints (4.44)-(4.46) to link the two parts to each other.

$$X_{t,p} - G_{t,p} = 0 \qquad \forall t \in T^{<}, p \in P(t) \tag{4.44}$$

$$C_{t,m} - \sum_{j \in J: \lambda_j = t} Y_{j,m} = 0 \qquad \forall t \in T, m \in M \tag{4.45}$$

$$U_{t,m} - \sum_{j \in J: \gamma_j = t} Y_{j,m} = 0 \qquad \forall t \in T, m \in M \tag{4.46}$$

All trips $t \in T^{<}$ that have departed before the start of the disruption should have the same composition as originally assigned, as is modelled by Constraints (4.44).

Constraints (4.45) state that the number of coupled train units at the start of a trip is equal to the number of train units that start their job at the trip. The number of uncoupled train units at the end of a trip is equal to the number of train units that finish their job at the end of the trip, as is modelled by Constraints (4.46). Note that Constraints (4.45) and (4.46) are required to link the Job part to the Composition part of the JC model. This is due to the fact that the start or end of a job in the Job part leads to a composition change in the Composition part.

The objective function (4.16) is then extended with Equation (4.18), just as in the EUT and SA model.

4.8.5 – Strengthening the formulation

In the JC model as described in the previous section, there are only constraints forbidding *pairs* of jobs to be chosen at the same time. However, these constraints can be tightened by forbidding *sets* of jobs, instead of pairs, to be chosen at the same time.

To that end, we define the undirected graph $G_r = (V_r, E_r)$, where the jobs $j \in J$ present in route $r \in R$ form the set of vertices V_r. There is an edge e between every two jobs j and j', if and only if $(j, j') \in CJ_1 \cup CJ_2$. This means that every pair of adjacent jobs is not compatible. We call this graph the conflict graph of route r.

A clique is a subset of vertices $cl \subset V_r$ such that for every two vertices in cl there exists an edge connecting the two. Bron and Kerbosch (1973) present a more thorough explanation of cliques and a heuristic to find (maximum) cliques in a graph. So, every clique of jobs, $cl \subseteq V_r$ within the conflict graph G_r is a set of pairwise incompatible jobs. Hence we can strengthen our formulation by replacing Constraints (4.33) with Constraints (4.47) for all cliques $cl \subset V_r$ for all $r \in R$.

$$\sum_{j \in cl} W_j \leq 1 \qquad \forall r \in R, cl \subset V_r : cl \text{ clique} \tag{4.47}$$

Finding and adding all (maximum) cliques may increase the size of the MIP model and the overall solution time drastically, since there may be an exponential number of cliques. For this reason, we add only some easy to find cliques. We use two such types of cliques, as described below.

Both types of cliques contain jobs j^1, j^2, ..., $j^n \in J$. All jobs in a clique are related to the same route and have at least one common trip. Furthermore, for both types it holds that the train unit assigned to job j^i, for $i \in \{1, ..., n-1\}$, is coupled to the composition earlier than the train unit assigned to job j^{i+1}.

The first type of cliques (CCJ^1) in our conflict graph is constructed such that in each clique job j^1 is uncoupled first, then j^2 and so on. Furthermore, for all $i = 1, \ldots, n$ the uncoupling side of job j^i is equal to the coupling side of jobs j^{i+1}, j^{i+2}, ..., j^n ($\pi_{j^i} = \omega_{j_{i+1}} = \omega_{j_{i+2}} = \ldots = \omega_{j^n}$). This is not feasible, because the train units assigned to jobs $j^{i+1}, \ldots, j^n$ are blocking the uncoupling of job j^i. So, all tuples of jobs within CCJ^1 are pairwise incompatible.

$$\begin{aligned} CCJ^1 := \{ & (j^1, j^2, \ldots, j^n) \in J^n : T(j^1) \cap T(j^2) \cap \ldots \cap T(j^n) \neq \emptyset \\ & \wedge \tau^{dep}_{j^1} < \tau^{dep}_{j^2} < ... < \tau^{dep}_{j^n} < \tau^{arr}_{j^1} < \tau^{arr}_{j^2} < ... < \tau^{arr}_{j^n} \\ & \wedge \pi_{j^i} = \omega_{j^{i+1}} = \ldots = \omega_{j^n} \qquad \forall i = 1, 2, \ldots, n-1\} \end{aligned} \tag{4.48}$$

The second type of cliques within our conflict graph (CCJ^2) consists of sets of jobs $\{j^1, \ldots, j^n\}$ such that the train unit assigned to job j^{i+1} is uncoupled before the train unit assigned to job j^i is uncoupled. Furthermore, the uncoupling side of job j^{i+1} is different from the side where it was coupled. This is not allowed, because the train units assigned to job $j^1, j^2, \ldots, j^i$ are still there (see the set CJ^2 as an example of a single job blocking the uncoupling of job j^{i+1}). So, all tuples of jobs within CCJ^2 are pairwise incompatible. Thus the set CCJ^2 can be described as follows:

$$\begin{aligned} CCJ^2 := \{ & (j^1, j^2, \ldots, j^n) \in J^n : T(j^1) \cap T(j^2) \cap \ldots \cap T(j^n) \neq \emptyset \\ & \wedge \tau^{dep}_{j^1} < \tau^{dep}_{j^2} < ... < \tau^{dep}_{j^n} < \tau^{arr}_{j^n} < \tau^{arr}_{j^{n-1}} < ... < \tau^{arr}_{j^1} \\ & \wedge \pi_{j^i} \neq \omega_{j^i} \qquad \forall i = 1, 2, \ldots, n\} \end{aligned} \tag{4.49}$$

There is a final constraint that can strengthen the model formulation. Recall that in the Netherlands it is not allowed to both couple and uncouple between two succeeding trips. That means that it is not allowed that two selected jobs end and start directly after each other at the same station. So, Constraints (4.50) can be added to the formulation as valid inequality.

$$W_j + W_{j'} \leq 1 \qquad \forall (j, j') \in J \times J : \lambda_j = \sigma(\gamma_{j'}) \tag{4.50}$$

4.9 – RESULTS

In this section we discuss the results of applying the EUT model, the SA model, and the JC model on different instances of the main Dutch passenger railway operator NS. We start in Section 4.9.1 with a description of the instances and the used parameters. Thereafter, in Sections 4.9.2 and 4.9.3 we present the results with a maximum turnaround time of 10 minutes and with a maximum turnaround time of 30 minutes, respectively. Finally, in Section 4.9.4 we give an overview of the objective function components for each model.

All computations described in this section are ran with CPLEX 12.5.1 on an Intel (R) Core (ITM) i5-3210M processor with 2.50 GHz and 8GB RAM. The maximum computation time is set to 500 seconds per instance and the allowed gap size is set to 0%, thus no gap is allowed.

4.9.1 – Instances and Parameters

In this section we first describe the instances and thereafter the parameters. We ran different experiments on trips of the 2200, 2800, and 3000 line in the Netherlands. Here trains are travelling from Breda (Bd) to Amsterdam (Asd) (2200 line), from Rotterdam (Rtd) to Deventer (Dv) (2800 line), and from Den Helder (Hdr) to Nijmegen (Nm) (3000 line). These lines lead to a total of 1095 trips per day. See Figure 4.10 for a visual representation, where the 2200 line is represented by black edges, the 2800 line by black dotted edges, and the 3000 line by dark grey edges.

Case number	#RS types	Maximum turnaround time	Disrupted area
1a	2	10	Ut - Asd
1b	2	10	Gv - Rtd
2a	3	10	Ut - Asd
2b	3	10	Gv - Rtd
3a	2	30	Ut - Asd
3b	2	30	Gv - Rtd
4a	3	30	Ut - Asd
4b	3	30	Gv - Rtd

Table 4.2: Different instances

Table 4.2 gives an overview of the instances on which the models have been tested. Here, "#RS types" denotes the number of regular rolling stock types used. As can be seen, this is either two or three. In the instances with two rolling stock types, the train units consist of either three or four carriages, while in the instances with three different rolling stock types they consist of either three, four or five carriages. The maximum number of carriages in a train

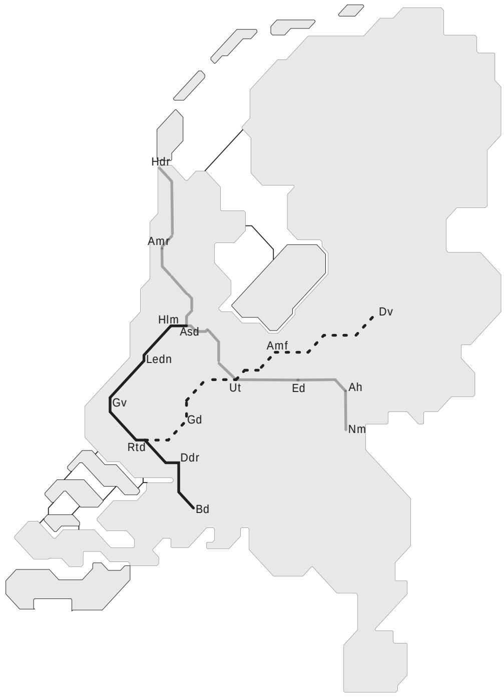


Figure 4.10: Case lines of the 2012 network of Netherlands Railways

equals 15, so in total there are 31 compositions and 356 composition changes possible when using two different rolling stock types, and 72 compositions and 884 composition changes are possible when using three different rolling stock types.

The column "Maximum turnaround time" denotes the maximum time a train is allowed to wait for its succeeding return trip in an end station. This maximum time equals 10 minutes in half of the instances and 30 minutes in the rest. The actual turnaround time denotes the time between the arrival time of an incoming trip and the departure time of the first return trip in an end station.

In the instances with a maximum turnaround time of 30 minutes most of the incoming trips have a succeeding return trip in an end station, since the lines are operated with a frequency of 2 trains per hour. As a consequence, in these instances the routes are long, and thus for each route $r \in R$ there are many jobs $j \in J$ with $j \subset r$. This leads to a large number of possible jobs in the JC model, which increases the computation time.

If in an end station the time between an incoming trip and the first return trip exceeds the maximum turnaround time, then the train units are assumed to be transferred to the corresponding shunting yard in between the trips. In that case, the incoming trip and the return trip do not belong to the same route anymore.

The column "Disrupted area" describes the location where the disruption takes place. A disruption takes place either between the stations The Hague (Gv) and Rotterdam (Rtd) or between the stations Utrecht (Ut) and Amsterdam (Asd). In order to test whether the start time of the disruption has any influence on the computation time, we let disruptions take place between 07:00-09:00, 07:03-09:03, 07:06-09:06, ..., and 07:57 - 09:57, so in total during 20 different time slots. Only time slots in the morning rush are chosen, because these are the most complex instances to solve. Furthermore, we experiment with a number of train units requiring maintenance varying between one and six. All instances in Table 4.2 are solved for the different time slots and for the different numbers of train units requiring maintenance. As a result, there are in total $8 * 20 * 6 = 960$ instances which we solve with each of the three models.

Parameter	Symbol	Penalty
Cancelling	κ	10000
EOD deviation	ϱ	100
Capacity shortage kilometer	Δ	1
Carriage kilometer	ξ	1
Deviation original plan	ϕ	50
Missing maintenance	θ	300

Table 4.3: Objective function penalties

All models use the *same* objective function based on the objective coefficients mentioned in Table 4.3. The complete objective function is a weighted sum of the objective coefficients multiplied with their accompanying decision variables. Here "Cancelling" denotes the penalty for cancelling a trip. Since we consider cancelling a trip as the worst thing that can happen, the penalty for cancelling a trip is higher than any other penalty. "EOD deviation" means the penalty for deviating from the scheduled end-of-day balance. For each negative difference, a dead-heading trip must be scheduled during the night to rebalance. This is expensive, so we want to keep the deviation small.

"Capacity shortage kilometer" stands for the penalty on the number of passengers that do not fit in an assigned composition, measured per kilometer. Note that we use the original passenger demand for each trip as provided by NS. So, the demand on the trips that are operated is assumed to be unchanged during the disruption. Taking dynamic passenger demand directly into account during the disruption is outside the scope of this paper. We refer to Kroon et al. (2014) for a paper that does take dynamic passenger demand into account (but no maintenance appointments). A similar approach could have been applied in the current paper as well.

"Carriage kilometer" is the penalty on the number of carriages assigned per kilometer. A trade-off exists between minimizing the number of carriage kilometers and minimizing the seat-shortages for passengers, because both objectives are conflicting. Minimizing seat-shortages will lead to appointing large compositions to trips, while minimizing the number of carriage kilometers will lead to appointing small compositions to trips.

"Deviation original plan" stands for the penalty on the difference between the original and the rescheduled plan in terms of the numbers of couplings and uncouplings taking place. Each additional shunting movement requires an additional crew task for which a crew member must be found. This takes time, and in a real-time situation not much time is available. As a consequence, we want to keep the number of additional shunting movements low.

Finally, "Missing maintenance" stands for the penalty on the number of train units that miss their scheduled maintenance appointment. As explained before, it is undesirable that a train unit misses its maintenance appointment, so we set a large penalty on this. The applied penalties come from existing literature or from discussions with dispatching experts of NS.

It is important to note that that, in case the models are able to prove optimality, then they find optimal solutions with the same optimal objective value . The models were able to prove optimality for most of the instances. This does not necessarily mean that the models produce exactly the same rolling stock circulation. However, the circulations are equally good with respect to the objective function. Therefore, we will mainly compare the models in terms of the computation time and in terms of the number of times an optimal solution was found.

4.9.2 – Maximum turnaround time 10 minutes

In this subsection we compare the results obtained by the three models for all instances with a maximum turnaround time of 10 minutes. We show average computation times for all problem instances with the same number of rolling stock types on the same initial locations, with the same maximum turnaround time, and with the same number of train units requiring maintenance. In other words, the average is taken over the 20 different disruption time slots while all other instance parameters remain fixed.

In all tables presenting the results, the first column (M) denotes the number of train units that require maintenance, the second column (Model) denotes which model was used to solve the instances, the third column (Time) represents the average computation time required to solve the problem instances, the fourth column ($\#NO$) presents the total number of times no proven optimal solution was found, the fifth column ($\#C$) presents the average number of constraints in the model, and the sixth column ($\#V$) presents the average number of variables used to solve the instances.

First, we show the results for the instances with a maximum turnaround time of 10 minutes. The results of using two regular rolling stock types are shown in Table 4.4 and Figure 4.11.

As can be seen, the JC model performs significantly better than the SA and the EUT model: Both the number of times a proven optimal solution was found and the average computation time are better in the JC model than in the EUT and the SA model. Furthermore, in some of the instances when there are many train units requiring maintenance, the EUT model is not able to find a proven optimal solution within 500 seconds. The JC model finds a proven optimal solution for all instances, and the SA model for all instances but two.

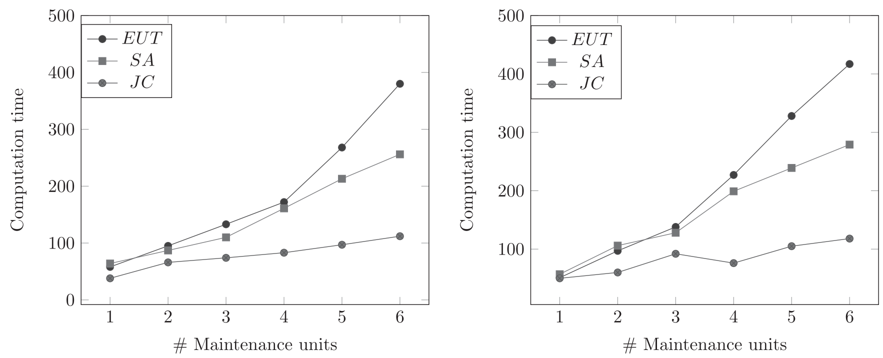
M	Model	Time	#NO	#C	#V
	EUT	58	0	165339	1063531
1	JC	38	0	83992	421309
	SA	64	0	132646	645108
	EUT	95	0	274219	1823416
2	JC	66	0	87004	438030
	SA	87	0	168126	855405
	EUT	133	0	383100	2562576
3	JC	74	0	90017	439832
	SA	110	0	203607	1065572
	EUT	172	0	474373	3216358
4	JC	83	0	92309	441043
	SA	161	0	229304	1205559
	EUT	268	0	568219	3931422
5	JC	97	0	94641	452064
	SA	213	0	260132	1443211
	EUT	380	6	666201	4803231
6	JC	112	0	97032	459216
	SA	256	1	291240	1603491

(a) Case 1a Ut - Asd

M	Model	Time	#NO	#C	#V
	EUT	51	0	165444	1064512
1	JC	50	0	84115	436677
	SA	57	0	132762	645703
	EUT	97	0	274498	1825098
2	JC	60	0	87104	438490
	SA	106	0	168275	856194
	EUT	138	0	383490	2585684
3	JC	92	0	90166	440303
	SA	128	0	203789	1066685
	EUT	227	0	474856	3219325
4	JC	76	0	92461	441531
	SA	199	0	229509	1206651
	EUT	328	2	569021	3942681
5	JC	105	0	96721	452497
	SA	239	0	260405	1443981
	EUT	417	8	667091	4805197
6	JC	118	0	97333	460134
	SA	279	1	291865	1604754

(b) Case 1b Rtd - Gv

Table 4.4: Results with 2 regular types and 10 minutes maximum turnaround time



(a) Ut-Asd (b) Rtd-Gv

Figure 4.11: Computation times with 2 regular types and 10 minutes maximum turnaround time

The results of applying the models on instances with three regular rolling stock types are shown in Table 4.5 and Figure 4.12. The JC model performs again best, both in terms of the computation time and in terms of the number of proven optimal solutions. The SA model is second and the EUT model performs worst, having difficulty to solve instances with many maintenance appointments.

The JC model performs better because it does not need any additional compositions for an additional train unit that requires maintenance. The more additional train units require maintenance, the more beneficial this becomes. As can be seen, it results in fewer variables and constraints than the SA and EUT model require. We conclude that the JC model performs best if the maximum turnaround time is 10 minutes.

M	Model	Time	#NO	#C	#V
1	EUT	112	0	305545	2048938
	JC	141	0	169642	1013684
	SA	101	0	219016	1221847
2	EUT	191	0	471149	3217491
	JC	173	0	172654	1015486
	SA	180	0	257384	1432144
3	EUT	276	5	572205	3921008
	JC	200	0	175666	1017288
	SA	227	0	285972	1571981
4	EUT	358	10	737810	5089561
	JC	220	0	178679	1019090
	SA	270	0	324341	1782278
5	EUT	404	14	885807	6131286
	JC	251	0	181688	1020892
	SA	304	2	352926	1922115
6	EUT	462	16	986867	6834803
	JC	297	1	184704	1022694
	SA	359	4	381618	2061952

(a) Case 2a Ut - Asd

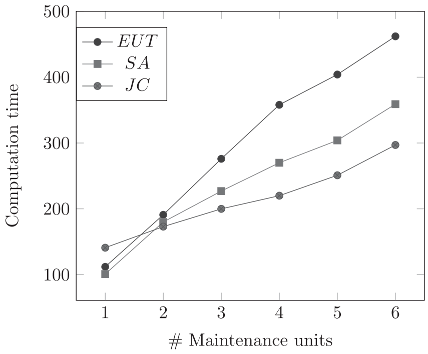
M	Model	Time	#NO	#C	#V
1	EUT	88	0	305856	2050828
	JC	89	0	169852	1014675
	SA	73	0	219217	1222974
2	EUT	131	0	471629	3220459
	JC	121	0	172877	1016488
	SA	125	0	257618	1433465
3	EUT	221	1	572788	3924625
	JC	190	0	175902	1018301
	SA	160	0	286229	1573431
4	EUT	304	3	738562	5094256
	JC	183	0	178928	1020114
	SA	200	0	324631	1783922
5	EUT	421	11	886710	6136942
	JC	226	0	181950	1021927
	SA	227	0	353239	1923888
6	EUT	456	15	987873	6841108
	JC	241	0	184979	1023740
	SA	294	2	381854	2063854

(b) Case 2b Rtd - Gv

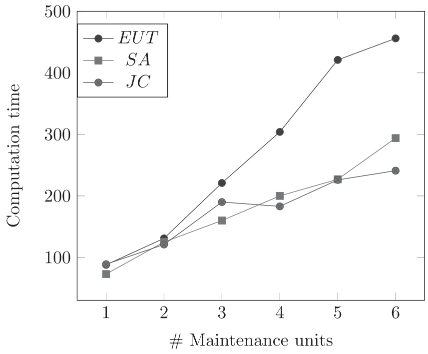
Table 4.5: Results with 3 regular types and 10 minutes maximum turnaround time

4.9.3 – Maximum turnaround time 30 minutes

In contrast with the results with a maximum turnaround time of 10 minutes, the SA model outperforms both the EUT and the JC model when the maximum turnaround time equals 30 minutes. Due to the larger maximum turnaround time, the jobs are now longer and there are many more possible jobs. This makes it a harder problem to solve.



(a) Ut-Asd



(b) Rtd-Gv

Figure 4.12: Computation times with 3 regular types and 10 minutes maximum turnaround time

M	Model	Time	#NO	#C	#V
1	EUT	45	0	175221	1063525
	JC	190	0	148931	443767
	SA	39	0	142730	645102
2	EUT	103	0	291243	1823410
	JC	238	0	153389	447078
	SA	81	0	183456	855399
3	EUT	203	0	407249	2583295
	JC	270	0	157845	450389
	SA	156	0	224180	1065696
4	EUT	235	0	504462	3216352
	JC	315	1	162300	453700
	SA	232	0	252641	1205533
5	EUT	361	2	602319	3904671
	JC	369	3	167027	456918
	SA	270	0	280120	1360012
6	EUT	430	7	700232	4700129
	JC	425	8	171872	499812
	SA	320	2	301321	1421208

(a) Case 3a Ut - Asd

M	Model	Time	#NO	#C	#V
1	EUT	41	0	175340	1064506
	JC	182	0	137231	443456
	SA	37	0	141586	645697
2	EUT	84	0	291522	1825092
	JC	190	0	141613	446626
	SA	71	0	181967	856188
3	EUT	172	0	407639	2585678
	JC	210	0	145993	449796
	SA	123	0	222346	1066679
4	EUT	231	0	504882	3219319
	JC	220	0	150372	452966
	SA	188	0	253262	1206645
5	EUT	290	2	607123	3924021
	JC	296	1	154871	456723
	SA	211	1	282432	1399864
6	EUT	410	7	710023	4647910
	JC	398	5	158120	459102
	SA	280	1	310212	1581904

(b) Case 3b Rtd - Gv

Table 4.6: Results with 2 regular types and 30 minutes maximum turnaround time

For the instances with two regular rolling stock types, the results are shown in Tables 4.6 and Figure 4.13. All models tend to be slower than in the instances with a maximum turnaround time of 10 minutes. The SA model outperforms the EUT model in terms of the computation time and in terms of the number of proven optimal solutions. The JC model performs worst. As explained before, this is due to the fact that the number of possible jobs has increased. On the other hand, the computation time for the JC model does not increase

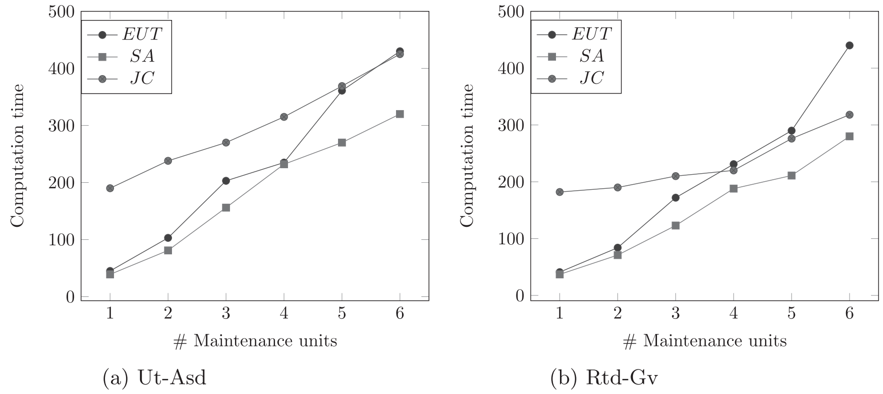


Figure 4.13: Computation times with 2 regular types and 30 minutes maximum turnaround time

as quickly as for the other models when more train units require maintenance. In the end, the JC model performs even better than the EUT model when 6 train units have a maintenance appointment.

In conclusion, the SA model performs better than both the JC and the EUT model when using two regular types and a maximum turnaround time of 30 minutes.

The results when using three regular rolling stock types are shown in Tables 4.7 and Figure 4.14. As can be seen, the SA model outperforms both the JC and the EUT model in terms of the computation time and in terms of the number of proven optimal solutions again. The model is able to find proven optimal solutions for most of the instances within the time limit, while the other models have more problems with finding proven optimal solutions.

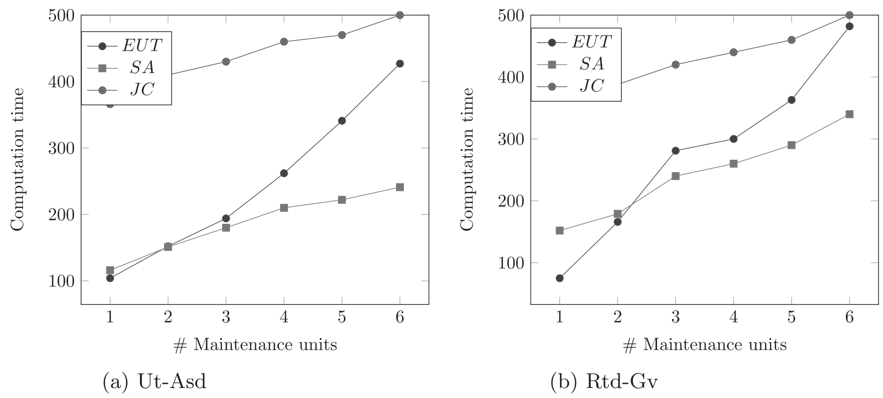


Figure 4.14: Computation times with 3 regular types and 30 minutes maximum turnaround time

Just as with two regular rolling stock types, it takes longer to find an optimal solution for the JC model from the start, but the computation time does not increase quickly when more train units require maintenance. However, the JC model is not able to find a proven optimal solution for any of the instances with 6 maintenance units. Note that the model was able to

M	Model	Time	#NO	#C	#V
1	EUT	104	0	324633	2049838
	JC	366	3	252781	1026777
	SA	116	0	232064	1221847
2	EUT	152	0	482393	3069888
	JC	410	9	257343	1030192
	SA	151	0	257159	1361684
3	EUT	194	0	590055	3794180
	JC	430	10	258106	1030312
	SA	180	0	282257	1501521
4	EUT	262	4	747896	4835905
	JC	460	12	262743	1033742
	SA	210	1	307352	1641358
5	EUT	341	9	855553	5508888
	JC	470	15	267380	1037172
	SA	222	2	332447	1781195
6	EUT	427	14	963208	6242939
	JC	-	20	272003	1040602
	SA	241	2	357582	1921032

(a) Case 4a Ut - Asd

M	Model	Time	#NO	#C	#V
1	EUT	75	0	324628	2067133
	JC	387	4	235246	1023441
	SA	152	0	232229	1229186
2	EUT	166	0	482867	3093514
	JC	388	5	239729	1026715
	SA	179	0	257335	1370023
3	EUT	281	4	590112	3810032
	JC	420	7	244215	1029989
	SA	240	1	282444	1581189
4	EUT	300	8	747882	4839291
	JC	440	10	248698	1033263
	SA	260	3	307550	1670246
5	EUT	363	11	855538	5539422
	JC	460	11	253181	1036537
	SA	290	4	332656	1813218
6	EUT	482	14	963196	6268116
	JC	-	20	257664	1039811
	SA	340	8	357782	1972034

(b) Case 4b Rtd - Gv

Table 4.7: Results with 3 regular types and 30 minutes maximum turnaround time

find feasible solutions for most cases with 6 maintenance units, but it was not able to prove optimality within the time window of 500 seconds.

To conclude, both the SA and the EUT model are not influenced heavily by the maximum turnaround time. The computation times differ little between having a maximum turnaround time of 10 minutes or having one of 30 minutes. On the other hand, the maximum turnaround time has a significant influence on the JC model. With a maximum turnaround time of 30 minutes, the computation times increase drastically. Furthermore, the SA model outperforms both the JC and the EUT model in terms of the computation time and in terms of the number of proven optimal solutions found when there are three regular types and a maximum turnaround time of 30 minutes.

4.9.4 – Evaluating the objective components

As mentioned before, the three models give the same optimal objective value on the same instance. This does not necessarily mean that the solutions are the same. In order to investigate whether the three models lead to structurally different solutions, Figure 4.15 shows a pie chart

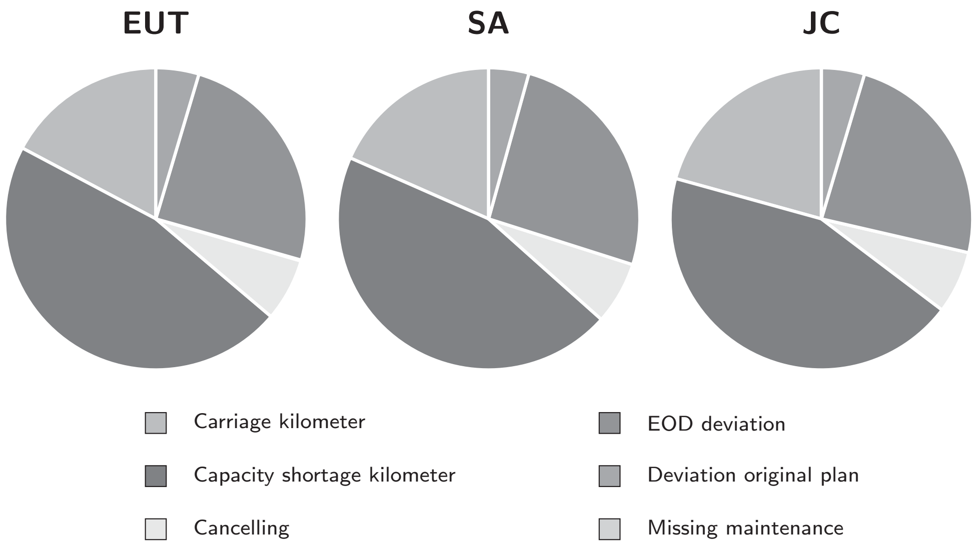


Figure 4.15: Objective function contributions per model

where the average percentage contribution of each objective aspect is shown per model. This average is calculated over all instances.

As can be seen, the contribution of cancelling a trip is the same for the three models. This is because the penalty for cancelling a trip is by far the largest penalty. Hence all models cancel as few additional trips as possible. Furthermore, the three models differ a little with respect to capacity shortages, carriage kilometers, end-of-day balance deviations, met maintenance appointments, and deviations from the original plan. There are small differences between the percentages, but the contribution of each aspect is almost the same for all models. Only in the EUT model maintenance appointments are actually missed, but this happened in just two instances. From the pie chart we can conclude that there are no structural differences between the results of the three models.

4.10 – CONCLUSIONS AND FURTHER RESEARCH

In this paper, three models are presented and compared for rescheduling the rolling stock of passenger trains during large disruptions, while taking scheduled maintenance appointments into account. The considered problem is an extension of the Rolling Stock Rescheduling Problem (RSRP). All models extend the Composition model of Fioole et al. (2006) and Nielsen (2011), which is known for rescheduling rolling stock without maintenance appointments.

The Extra Unit Type (EUT) model uses an additional rolling stock type for each train unit that require maintenance. In this way constraints can be imposed on them. This extension has the drawback that adding additional rolling stock types leads to a rapid increase in the number of possible compositions and composition changes. As a result, the EUT model tends to require more computation time than the other models when more train units require maintenance.

The second model is the Shadow-Account (SA) model. Within the SA model a shadow account for all train units is maintained, in particular for the maintenance units. In this way, maintenance constraints can be imposed on the train units that require maintenance.

The third model is the Job-Composition (JC) model. This model assigns train units to jobs. As a result, a path is created for each train unit. Specific paths can be created for train units that require maintenance leading to the corresponding maintenance locations.

The three models have been tested on a large number of instances of NS, the main Dutch operator of passenger trains. The models use the same objective function. Therefore, if the models find a proven optimal solution for an instance, then the obtained optimal objective function values are the same. Therefore we compared the models on their computation time and on the number of times they found a proven optimal solution.

The results show that the SA and the EUT model are rather insensitive to the maximum turnaround time. Their computation times differ little between instances with a maximum turnaround time of 10 minutes and instances with a maximum turnaround time of 30 minutes that are otherwise fully comparable. This is in contrast with the JC model: this model performs best on instances with a maximum turnaround time of 10 minutes. However, it is considerably slower on instances with a longer maximum turnaround time. This is due to the increase in the number of possible jobs in case of a longer maximum turnaround time.

As a consequence, the EUT model is inferior to the SA model and to the JC model. Whether the SA or the JC model performs better depends on the maximum turnaround time. The SA model performs better than the JC model on instances with a maximum turnaround time of 30 minutes. The JC model performs better than the SA model on instances with a maximum turnaround time of 10 minutes. An additional advantage of the JC model is that it is not necessary to assume that at most one train unit requires maintenance in a composition, as is necessary in the other models.

There are several directions for further research. First, a dynamic Branch & Price & Cut approach may be used to solve the JC model. In this way it may be possible to solve the instances with a longer maximum turnaround time faster. This can be incorporated with a column generation technique. The main challenge when using a column generation technique is that the order of the train units in the compositions is important. As a result, the columns are highly dependent on each other. This is in contrast with, for instance, crew rescheduling where the different crew members (driver, conductor) per train are rather independent of each other, unless it is specified that they should operate as much as possible as a *team*.

Secondly, the current models can be extended in different ways. A first possible extension is that rolling stock units requiring maintenance might *swap* maintenance appointments. This results in more flexibility while rescheduling the rolling stock. A second interesting extension would be to allow empty train movements. This can help rolling stock units reaching their maintenance appointment in time.

Thirdly, an interesting extension of the research to rolling stock rescheduling could be to include reserve rolling units which can only be used during a disruption. It would be interesting to investigate how many rolling stock units should be used as reserve, and where to locate them, especially when some rolling stock units have a maintenance appointment.

Fourthly, an integrated approach to reschedule the timetable, the rolling stock, and the crew schedules could lead to better results than rescheduling the three schedules sequentially. It would be interesting to test this in further research.

Finally, other practical aspects are important to be included in the RSRP. Especially the integration of accurate dynamic passenger demand with the maintenance appointments is an interesting topic for further research. Furthermore, station routing should be incorporated in the disruption management models. Station routing is currently neglected in the rescheduling models. For instance, determining from which platforms trains arrive or depart is an important factor during disruptions.

5 – Rolling Stock Rescheduling in Passenger Railway Transportation Using Dead-Heading Trips and Adjusted Passenger Demand

This chapter considers the paper (Wagenaar and Kroon (2016)).

Co-authors: L.G. Kroon

5.1 – INTRODUCTION AND CONTRIBUTIONS

5.1.1 – Introduction

The main focus of Netherlands Railways (NS) is to provide a good passenger service. Therefore, NS is constantly focusing on improving the quality of its services. An important measure for this quality is the ability to react to unforeseen events occurring during the day. Two kinds of unforeseen events are of interest to railway operators: disruptions and disturbances. A disruption causes the planned timetable, rolling stock circulation, and crew schedule to be infeasible. During a disturbance, however, a delay is either absorbed by the slack in the system or by rescheduling only the timetable: the rolling stock and crew schedule can absorb the disturbance and do not have to be rescheduled. The focus of this paper is on dealing with the first type of unforeseen events: disruptions.

All planned resource schedules (timetable, rolling stock circulation, and crew schedule) have to be adapted as soon as a disruption occurs in order to secure their feasibility. In practice, the first step is to update the original timetable. In the Netherlands, more than a thousand different, so called, contingency plans exist to update the timetable. These contingency plans contain a number of rules stating which trains have to be cancelled, rerouted, or delayed in case of a specific disruption. After the end of a disruption, railway operators usually want to have their timetable to be as much alike the original timetable as possible. This is already taken into account in the predescribed contingency plans. Secondly, based on the new timetable, the original rolling stock circulation has to be rescheduled. At last, with the new timetable and the new rolling stock schedule as input, the crew schedule must be modified.

In this paper we focus on the second step: rescheduling the planned rolling stock circulation given the rescheduled timetable as input. There already exist models for the Rolling Stock Rescheduling Problem (RSRP), see for instance Nielsen (2011), Nielsen et al. (2012), Haahr et al. (2014), and Sato et al. (2009). These models are currently not applied in practice yet. One of the reasons is that not all details of the real world are taken into account. In this paper we introduce an extension of the RSRP model which includes two of these practical details.

Due to the disruption, it may be that certain stations have a surplus of rolling stock units while other stations have a shortage of rolling stock units to execute the updated timetable. A shortage of rolling stock units can lead to additional cancelled trains, because there is no rolling stock unit available for at least one of the trips in the updated timetable. Furthermore, a shortage of rolling stock units possibly leads to appointing trains with too little capacity for the passenger demand on a trip. That is why NS has the possibility to schedule empty trains, called dead-heading trips, from one station to another to increase the local inventory during a disruption. By using dead-heading trips, NS wants to decrease the number of cancelled trains and to increase the customer satisfaction during and after a disruption. In this paper we will adapt the current RSRP models, by adding the possibility to use dead-heading trips during a disruption.

Secondly, the major objective of the new rolling stock circulation is to uphold a good passenger service. In other words, to cancel as little trains as possible and to use trains with enough capacity for all passenger demand. Most of the current rescheduling models assume passenger demand to be static. However, passenger demand depends on the appointed rolling stock units to trips (e.g. cancelling a trip leads to a demand increase on the next trip with the same origin and destination).

In the Netherlands, more information about passengers is currently available due to smart card data. With this data, we were able to identify the incoming and outgoing demand on a trip in the undisturbed situation. These are defined as the number of passengers that enter the railway system at the start of a trip and the number of passenger that leave the railway system after a trip. With the incoming and outgoing demand we are able to take adjusted passenger demand into account in the RSRP.

5.1.2 – Contributions and structure of the paper

The contributions of this paper can be summarized as follows. First of all, this paper is, to the best of our knowledge, the first to include unscheduled dead-heading trips in a formulation to tackle the rolling stock rescheduling problem. By including dead-heading trips, possibly less additional trains get cancelled and passenger satisfaction will increase. Secondly, an efficient preprocessing method is applied to select potential dead-heading trips from the complete set of possible dead-heading trips. In this way, dead-heading trips can adequately be included in the rolling stock rescheduling formulation. Thirdly, to the best of our knowledge, this paper is the first to include dynamic passenger flows directly in the formulation for rolling stock rescheduling in passenger railways. Finally, different boarding strategies for passengers in case the appointed capacity to a trip is less than the actual passenger demand are proposed and formulated.

Three important assumptions are taken into account while designing the models:

1. The order of train units within a composition is of importance (e.g. there is a difference between the following two compositions: ab and ba, even though both compositions exist of the same train unit types a and b).
2. The incoming passenger demand does not change due to a disruption.
3. Passengers do not leave the railway system prematurely and do not take a detour to their destination during a disruption.

The first assumption is of importance, because at Netherlands Railways it is defined at which side of an incoming train rolling stock units may be (un)coupled. As a consequence, we need to keep track which unit is in front of the incoming train and which unit resides at the back of the incoming train. The second assumption is used, because there is no information available about the change in the incoming demand due to the disruption. This assumption can be relaxed as soon as this information becomes available. The third assumption is used to simplify the problem. The objective of using adjusted passenger demand is to appoint rolling stock compositions with a large capacity to trips where the actual passenger demand is large. The capacities of the compositions differ significantly (e.g. the smallest composition has capacity for 405 passengers and the largest compositions for 1407 passengers). As a consequence, it is more important to predict the passenger demand on a global level than exactly.

The remainder of this paper is structured as follows: in the next section a literature overview is given. In Section 5.3 the models used to solve the RSRP are shown and discussed. In Section 5.4 the models are tested on real life instances of NS. Finally, in Section 5.5 conclusions and remarks on further research are given.

5.2 – LITERATURE REVIEW

Cacchiani et al. (2014) give an overview of related papers on disruption management in general. We refer to them for the reader interested in papers for timetable and crew rescheduling. In this paper we focus on rolling stock rescheduling. So, the remainder of the papers discussed are on (re)scheduling the rolling stock, given the (re)scheduled timetable as input.

5.2.1 – Scheduling

Fioole et al. (2006) formulated a model to assign rolling stock to the timetable in the scheduling phase. The model is able to handle the order of rolling stock units within compositions. Furthermore, it is able to handle complicated line structures, such as combining and splitting of trains. NS uses this model to generate the rolling stock schedules since 2004.

Cacchiani et al. (2011) investigate the problem of designing an original rolling stock circulation in such a way that the recoverability of the circulation in case of large disruptions is most effective. This model is an extension of the model used by Fioole et al. (2006). They include different disruption scenarios in their model and minimize the total number of cancelled trips, the additional shunting operations, and the deviations from the end-of-day rolling stock balances for all reallocation plans. Next to those targets, their objective focusses on the number of seat shortages, carriage kilometers, and the complexity and risk of shunting operations as well. Their model is tested on the 3000 line of NS between Den Helder and Nijmegen. They simulate different disruption scenarios and show that in their robust solution less trains have to be cancelled and the recovery costs are lower than in the originally planned schedule.

Lingaya et al. (2002) study the problem of scheduling locomotives and carriages. They describe a train as a Last-In-First-Out (LIFO) stack, where carriages can be coupled or uncoupled from the rear part of the train in LIFO order only. Their solution approach is based on a Dantzig-Wolfe reformulation, which is solved by means of column generation.

Borndörfer et al. (2015) use a hypergraph formulation to come up with a rolling stock circulation where certain practical requirements (e.g. maintenance) are taken into account. Their model is tested on real life instances of the German railway company Deutsche Bahn. Circulations for a generic week are found in between 10 minutes and 4 days of computation time by means of a column generation approach.

5.2.2 – Rescheduling

In this section we discuss papers studying the rolling stock rescheduling problem. Note that these papers do not include the possibility of scheduling dead-heading trips to reduce the number of additionally cancelled trains. Furthermore, the papers also do not take dynamic passenger flows directly into account.

Budai et al. (2010) introduce the Rolling Stock Rebalancing Problem. This problem occurs during short-term planning and during real time rescheduling. The problem they face is that a rescheduled rolling stock circulation is feasible, but there still exist off-balances in the rolling stock inventory at the end of the planning period. An off-balance is the difference between the original planned end-of-day balance and the rescheduled end-of-day balance. Their objective is to change the rolling stock circulation to solve the off-balances. The authors propose two heuristics. The first is a simple greedy approach: construct small feasible transformations to the rolling stock allocation iteratively until no more improvements are possible. The other is a two phase heuristic, where in the first phase a number of feasible transformations are selected and in the second phase an integer linear program is used to determine which transformations are selected. Their model is tested on the 3000 line of NS. It is shown that both heuristics can be used fast and effectively. The problem of this formulation is that it requires a rescheduled circulation as input and thereafter slight modifications are applied to reduce the off-balances. In our formulation, dead-heading trips might be used in order to decrease these off-balances during rescheduling.

Sato et al. (2009) propose a formulation for reallocating resources to trips in a railway network in case of large disruptions. Resources can either refer to rolling stock units or to crew members. The objective of the formulation is to produce a schedule that differs as little as possible from the original schedule. They use a two phase algorithm: in the first phase they attempt to resolve conflicts generated by the disruption. These conflicts are resolved through small changes to the original schedule. In the second phase a local search heuristic is used to improve the rescheduled plan. Their formulation is tested on a Japanese railway line. It is shown that feasible solutions arise in an acceptable amount of time for usage in practice. A difference with our approach is that this formulation does not take adjusted passenger demand or dead-heading trips into account.

Nielsen (2011) extended the model of Fioole et al. (2006) to cope with rescheduling. He formulated an integer programming problem with the adjusted timetable and the original rolling stock schedule as input and an adjusted rolling stock circulation as output. The formulation used in this paper is extended in our paper to cope with adjusted passenger demand and dead-heading trips.

Nielsen et al. (2012) propose a rolling horizon to solve the rolling stock rescheduling problem. The idea behind the rolling horizon is that at the beginning of the disruption not all information about the duration of the disruption is known: this information becomes gradually available. The rescheduling is periodically performed within a limited rolling horizon length, possibly taking new information into account. At each time instant where an updated timetable becomes available, or when a certain amount of time has passed without any update, the MIP of Nielsen (2011) is solved again for a certain time horizon. This model is tested on instances of NS. Solutions with small deviations from the original plan are found in short computation times.

Haahr et al. (2014) make use of a column generation approach to solve the RSRP. This method does not take the order of the rolling stock units within a composition into account. The method makes use of a decomposition method based on individual paths for the units. As a consequence, unit specific constraints could be applied to them. Haahr et al. (2015c) (Chapter 3) compares two different approaches for the RSRP. The first approach is based on the model of Fioole et al. (2006) and the second approach extended the model proposed by Haahr et al. (2014) by including the order of rolling stock units within a composition. The models are compared on instances from different railway operators in different countries. Results demonstrate that the model of Fioole et al. (2006) is on average faster to solve the rescheduling instances.

Wagenaar et al. (2016) (Chapter 4) introduce scheduled maintenance appointments in the RSRP. Certain rolling stock units have a maintenance appointment during the day at one of the stations. If this aspect is not included in the model for rescheduling, these rolling stock units will most likely miss their appointments. The authors introduce three different extensions of the model introduced by Fioole et al. (2006). Results demonstrate that their models are able to efficiently take maintenance appointments into account in the RSRP.

All of the above models assume passenger demand to be static. That means that every trip has a predescribed passenger demand, which is not influenced by the compositions appointed to other trips. In the extreme case that a trip gets cancelled, this means that the passenger demand for the next trip with the same origin and destination does not change, even though most passengers wait to board the next train to their destination.

Kroon et al. (2014) consider real-time rescheduling of rolling stock during large disruptions while taking dynamic passenger flows into account. They use a two-stage feedback loop, where in one stage the rolling stock allocation is optimized by using the model of Nielsen (2011) and in the other stage the effect of passenger flows on the allocation of the rolling stock is determined by means of a passenger simulation. This simulation provides feedback in terms of passenger delays due to limited capacity of the assigned rolling stock. This feedback is then used in the optimization model in order to reallocate the rolling stock again, in such a way that the passenger delay is reduced. Given the reallocation of the rolling stock, the passenger simulation is performed again and feedback is given to the optimization model and this loop continues for a number of iterations. In our model, we take changing passenger flows directly into the formulation into account.

5.3 – MODEL

In this section we introduce the model we use to reschedule the rolling stock after the occurence of a disruption given the modified timetable. In this model we take the order of the rolling stock units in a train into account. In Section 5.3.1 we start with explaining the algorithm that selects potential dead-heading trips from the set of all possible dead-heading trips. Thereafter, in Section 5.3.2 we include adjusted passenger demand in the RSRP directly, as opposed to Kroon et al. (2014) who use a two-stage feedback loop. Finally, in Section 5.3.4 we show the complete model that is used to solve the RSRP. In this model both dead-heading trips and adjusted passenger demand are included.

5.3.1 – Dead-heading trips

Dead-heading trips may be used in practice to transfer rolling stock units from stations with a surplus of train units to stations with a shortage of train units. A list of all potential dead-heading trips is required to take them into account in the formulation for rescheduling the rolling stock. It is possible to schedule a dead-heading trip at every time instance, so there is a long list of potential dead-heading trips. Therefore, it is impossible to take all possible dead-heading trips into account. We can limit this number by defining a potential departing station for a dead-heading trip to be the arrival station of a trip in the timetable, and a potential arrival station of a dead-heading trip to be the departing station of a different trip in the timetable. This can be done without loss of information, because all possibilities in between do not make a difference for the inventory registration. Therefore, all the dead-heading trips in between can be aggregated into one dead-heading trip. See, for instance, Figure 5.1 where four potential deadheading trips between the stations B and G are displayed. Note that a dead-heading trip could depart at any time instance between the departure at B and the arrival at G, these four are just shown as an example. These dead-heading trips are all aggregated into one dead-heading trip that departs the earliest directly after trip $A - B$ and arrives the latest just before the start of trip $G - H$. This means that, if this dead-heading trip is scheduled, at the start of trip $G - H$, station G has one additional rolling stock unit in its inventory, and station B has one rolling stock unit less in inventory at the start of trip $B - C$.

There are certain practical rules that potential dead-heading trips have to satisfy. In this section we propose a preprocessing module that retrieves only those dead-heading trips that

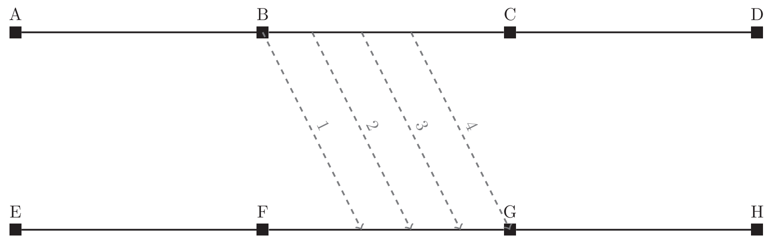


Figure 5.1: Dead-heading example

satisfy the imposed rules from the set of all possible dead-heading trips. The imposed rules are the following:

1. A dead-heading trip can only be scheduled after the disruption has occured and only until a certain amount of time after the disruption is over.
2. The travel time of the dead-heading trip may not be longer than a threshold value of time set by the operator.
3. The track that is disturbed due to the disruption may not be used by the dead-heading trip.
4. The dead-heading trip does not cause a conflict with the rescheduled timetable.

Furthermore, there is one obvious basic constraint which the empty train unit sent via a dead-heading trip has to fulfill: *The train unit can only be used in further operations after the dead-heading trip arrived at its destination.*

To this end, let T be the set of trips in the rescheduled timetable and S the set of stations in the railway network. Denote $s_t^d(s_t^a)$ as the station where trip $t \in T$ departs (arrives). Furthermore, let τ_t^d be the departure time of trip $t \in T$ and τ_t^a the arrival time. Define the predecessor of trip $t \in T$ as $\sigma(t)$ and $\lambda(t)$ as the previous trip with the same origin and destination station as trip $t \in T$. Then, we introduce $\nu_{s,s'}$ as the time it takes to transfer an empty train unit from station $s \in S$ to $s' \in S$. We assume that all scheduled trips $t \in T$ between stations s and s' have a travel time longer than or equal to $\nu_{s,s'}$. With respect to the dead-heading trips, we define D as the set of potential dead-heading trips satisfying all imposed rules. Here, $d \in D$ is defined to be a dead-heading trip scheduled to depart from station s_d^d and arriving at station s_d^a, departing the earliest at time τ_d^d and arriving the latest at time τ_d^a. The minimum travel time of a dead-heading trip $d \in D$ equals $\nu_{s_d^d, s_a^d}$.

As mentioned before, to limit the amount of potential dead-heading trips, we aggregate all possible dead-heading trips that depart from the arrival station of a trip to a departure station of a different trip in the timetable into one. We introduce a set D' containing all aggregated potential dead-heading trips. Algorithm 2 defines the set D'. Note that the set D' also contains the dead-heading trips which violate the imposed rules.

Algorithm 2: The set D'

for $t_1 \in T$ **do**
    for $t_2 \in T$ **do**
        if $s^a_{t_1} \neq s^d_{t_2} \wedge \tau^a_{t_1} + \nu_{s^a_{t_1}, s^d_{t_2}} \leq \tau^d_{t_2}$ **then**
            Create new dead-heading trip d, with:
            •$s^d_d = s^a_{t_1}$
            •$s^d_a = s^d_{t_2}$
            •$\tau^d_d = \tau^a_{t_1}$
            •$\tau^a_d = \tau^d_{t_2}$
            •Minimum traveltime $= \nu_{s^d_d, s^d_a}$
            $D' = D' \cup \{d\}$
        end if
    end for
end for

The first rule (1.) states that a dead-heading trip can only be used after the disruption has occured and until a certain amount of time after the disruption is over (denoted by ζ). The second rule (2.) imposes that a dead-heading trip in the set D' can only be added to the set D if the travel time is not larger than a threshold value γ. To check whether one of these rules is violated, denote dis^{start} as the start time of the disruption and dis^{end} as the predicted end time of the disruption. A potential dead-heading trip $d \in D$ has thus to satisfy conditions (5.1)-(5.3):

$$\tau^d_d \geq dis^{start} \tag{5.1}$$

$$\tau^a_d \leq dis^{end} + \zeta \tag{5.2}$$

$$\nu_{s^d_d, s^a_d} \leq \gamma \tag{5.3}$$

Note that the values ζ and γ are set by the operator and can obtain any possible positive value.

The third rule (3.) states that the scheduled dead-heading trip may not use the disturbed track. Denote $s(dis)$ and $s'(dis)$ as the two stations between which the disruption occurs. Then, Figure 5.2 shows the four different cases which may not occur. These cases can all be prevented by using a preprocessing method that checks which stations are passed by a dead-heading trip and removing those dead-heading trips that uses the track between stations $s(dis)$ and $s'(dis)$.

The fourth rule (4.) states that a dead-heading trip should not cause a conflict with the timetable. We define a dead-heading trip to be conflict-free if the following conditions hold:

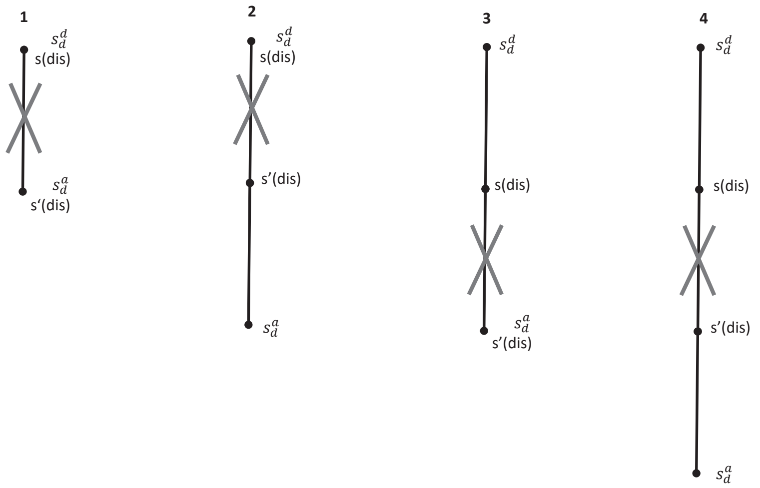


Figure 5.2: Four different forbidden dead-heading trips

1. The time between the departure of the dead-heading trip and the departure of other trips with the same origin and destination station is larger than the minimal headway time H.
2. The time between the arrival of the dead-heading trip and the arrival of other trips with the same origin and destination station is larger than the minimal headway time H.

There are many potential dead-heading trips possible between the arrival of one trip and the departure of another trip in the timetable. All those potential dead-heading trips are aggregated into a single dead-heading trip departing at τ_d^d the earliest and arriving at τ_d^a the latest. Most likely the difference between the earliest departure time and the latest arrival time is larger than the actual travel time of the dead-heading trip. Therefore, we define m_d as the amount of time slack in an aggregated dead-heading trip $d \in D'$:

$$m_d = \tau_d^a - \tau_d^d - \nu_{s_d^d, s_d^a}$$

Remember, $\lambda(t)$ is defined as the previous trip with the same origin and destination station as trip $t \in T$. Furthermore, χ_t is the time in between trip $t \in T$ and its predecessor $\lambda(t)$. This time is the smallest time between either the arrivals at station s_d^a or the departures at station s_d^d of the two trips:

$$\chi_t = \min(\tau_t^d - \tau_{\lambda(t)}^d, \tau_t^a - \tau_{\lambda(t)}^a)$$

To check whether there is room available on the track for one of the potential dead-heading trips in the aggregated dead-heading trip $d \in D'$, we define α_d as the set of potentially conflicting scheduled trips $t \in T$ with the aggregated dead-heading trip $d \in D'$. If the maximum time gap between the potentially conflicting trips (χ_t) is larger than twice the headway time H, then there is room in the network for at least one of the dead-heading trips in the aggregated dead-heading trip d to be scheduled. To this end α_d is defined as the set of trips $t \in T$ taking place on the same track as where we want to schedule the dead-heading trip $d \in D'$. See for example Figure 5.3. Here we have displayed the travel time of an aggregated dead-heading trip, indicated by a black line. This dead-heading trip can depart at τ_d^d the earliest and arrives at τ_d^a the latest. All dashed lines represent scheduled trips that are possibly in conflict with the potential dead-heading trip.

In order to check whether a trip $t \in T$ conflicts with the dead-heading trip, we first check whether they have the same departure and arrival station ($s_d^d = s_t^d$, and $s_d^a = s_t^a$). Secondly, we check whether the departure time of the scheduled trip t is later than the earliest departure time plus the minimal headway time ($\tau_d^d + H$) and before the latest possible departure time plus the minimum headway time ($\tau_d^d + m_d + H$) of the dead-heading trip. Finally, the arrival time of the trip t has to be before the latest possible arrival time plus the minimium headway time ($\tau_d^a + H$) and after the earliest possible arrival time plus the minimum headway time ($\tau_d^a - m_d + H$) of the dead-heading trip. This is also visualized in Figure 5.3.

$$\begin{aligned}\alpha_d = &\{t \in T | s_d^d = s_t^d \wedge s_d^a = s_t^a \wedge \tau_d^d + H \leq \tau_t^d \leq \tau_d^d + m_d + H \\ &\wedge \tau_d^a - m_d + H \leq \tau_t^a \leq \tau_d^a + H\}\end{aligned} \tag{5.4}$$

Thus, Equation (5.4) defines the set of trips $t \in T$ that have a potential conflict with the aggregated dead-heading trip $d \in D'$.

In order to test whether there is room for the dead-heading trip to be scheduled, we need to check whether the maximum time gap of the trips in the set α_d is larger than twice the minimum allowed headway time H. So, if inequality (5.5) is satisfied, then there is room for at least one of the dead-heading trips in the aggregated dead-heading trip $d \in D'$ to be scheduled.

$$\max_{t \in \alpha_d} \chi_t \geq 2H \tag{5.5}$$

To conclude, a dead-heading trip $d \in D'$ is added to the set of potential dead-heading trips D if and only if the dead-heading trip satisfies conditions (5.1)-(5.3) and condition (5.5), and does not use the track between stations $s(dis)$ and $s'(dis)$.

5.3.2 – Modelling passenger demand by passenger flows

In the Netherlands smart cards are used since 2012 in public transport. A passenger checks in with his or her smart card at his or her origin and checks out again at his or her destination. Therefore, smart card data gives the time and location a passenger enters and leaves the railway

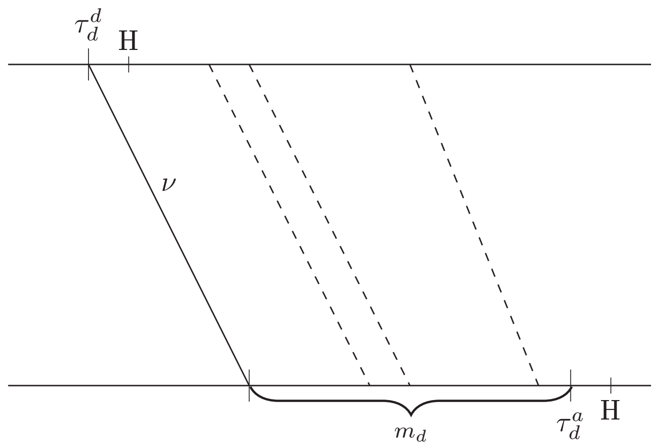


Figure 5.3: Example of the potential conflicting trips with a dead-heading trip

system. The passenger demand per trip can be estimated by tracking the route passengers use from origin to destination during the undisturbed situation. This information enables us to take changing passenger flows (e.g. due to a disruption) into account in the RSRP.

With smart card data it is also possible to estimate the incoming and outgoing demand per trip, see Van der Hurk et al. (2015). In the undisturbed situation, we define the incoming demand on trip $t \in T$, denoted by $\widetilde{ID}_t$, as the number of passengers that want to enter the railway system by taking trip t. The number of passengers that want to leave the railway system after trip $t \in T$ in the undisturbed situation is defined as the outgoing demand and is denoted by $\widetilde{OD}_t$. Additionally, the passenger flow on trip $t \in T$ (the number of passengers in the train) in the undisturbed situation is denoted by $\widetilde{F}_t$. Remember that $\sigma(t)$ is defined as the predecessor of trip $t \in T$ and $\lambda(t)$ as the previous trip with the same origin and destination station as trip $t \in T$.

Before it is possible to model the passenger flow, we introduce the following decision variables:

- $F_t \in \mathbb{R}_+$: the passenger flow sent through trip $t \in T$
- $SS_t \in \mathbb{R}_+$: the total capacity shortage for passengers on trip $t \in T$
- $OD_t \in \mathbb{R}_+$: the number of passengers that actually leave the railway system after trip $t \in T$

Figure 5.4 provides a simple example of adjusted passenger demand. In the left part there are in total 4 trips scheduled: $(a-b), (b-c), (a'-b')$, and $(b'-c')$. The two trips $(a-b)$ and $(b-c)$ are scheduled between the same two stations as the two trips $(a'-b')$ and $(b'-c')$, the only difference is that the trips $(a'-b')$ and $(b'-c')$ are scheduled later in time. The

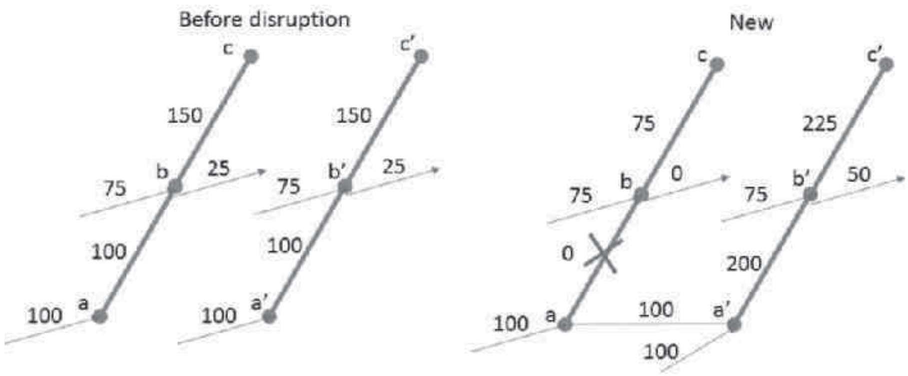


Figure 5.4: Example of taking adjusted passenger demand into account

following parameters and variables are present in the left part of Figure 5.4 for the trips $(a-b)$ and $(b-c)$:

- $\widetilde{ID}_{(a-b)} = F_{(a-b)} = \widetilde{F}_{(a-b)} = 100$
- $\widetilde{OD}_{(a-b)} = OD_{(a-b)} = 25$
- $\widetilde{ID}_{(b-c)} = 75$
- $F_{(b-c)} = \widetilde{F}_{(b-c)} = 150$

The same parameter and variable values hold for the trips $(a'-b')$ and $(b'-c')$.

In the right part of the figure we assume that a disruption occurs, leading to a complete blockage of the trip $(a-b)$. As a result, the passenger demand on all trips changes.

- First, it is assumed that the incoming demand stays the same for the trip $(a-b)$. However, due to the disruption no train can be appointed to the trip, causing a capacity shortage equal to the incoming demand: $SS_{(a-b)} = \widetilde{ID}_{(a-b)} = 100$.
- As a consequence, $F_{(a-b)} = OD_{(a-b)} = 0$.
- As a result, the passenger flow on trip $(b-c)$ only depends on its incoming demand: $F_{(b-c)} = \widetilde{ID}_{(b-c)} = 75$.
- The passenger flow on trip $(a'-b')$, however, now depends on the incoming demand and on the capacity shortage of its previous trip: $F_{(a'-b')} = \widetilde{ID}_{(a'-b')} + SS_{(a-b)} = 200$.
- That results in $OD_{(a'-b')} = \widetilde{OD}_{(a-b)} + \widetilde{OD}_{(a'-b')} = 50$.
- Finally, the passenger flow on trip $(b'-c')$ depends on its incoming demand and on the passenger flow of the previous trip: $F_{(b'-c')} = \widetilde{ID}_{(b'-c')} + F_{(a'-b')} - OD_{(a'-b')} = 225$.

We want to handle adjusted passenger demand as explained by means of the example. Remember that the following assumptions are included in the model with respect to passenger flows:

1. The incoming passenger demand does not change due to a disruption.
2. Passengers do not leave the railway system prematurely and do not take a detour to their destination. If a train is cancelled, passengers wait for the next train with the same origin and destination.

This means that we do not model passenger demand as it would precisely happen in reality. However, we want to take the adjusted passenger demand into account such that the model appoints rolling stock compositions with enough capacity to the trips. As a consequence, it does not matter whether the demand on a trip equals, for instance, 1050 or 1150 passengers. In both scenarios, the best available composition in the Netherlands to appoint to the trip is a train with capacity for 1200 passengers. Therefore, we only take adjusted passenger demand on an aggregated level into account.

To model the adjusted passenger demand in a mathematical formulation, we denote P as the set of possible compositions, where a composition is a combination of train units that can be used on a trip. Subsequently, cap_p is the capacity for passengers in composition $p \in P$. This capacity is measured as the maximum number of passengers that can possibly fit in the rolling stock units present in the composition. Denote V as the unique set of trips, meaning that V contains exactly every trip with the same origin and destination once.

Then, the decision variable $X_{t,p}$ is equal to 1 if composition $p \in P$ is appointed to trip $t \in T$. With the introduced parameters and decision variables, the following constraints are able to keep track of the passenger flows in the railway network:

$$F_t + SS_t = \widetilde{ID}_t + F_{\sigma(t)} - OD_{\sigma(t)} + SS_{\lambda(t)} \qquad \forall t \in T \tag{5.6}$$

$$F_t \leq \sum_{p \in P} X_{t,p} \cdot cap_p \qquad \forall t \in T \tag{5.7}$$

$$\sum_{\substack{t \in T \\ s_t^d = s_v^d, s_t^a = s_v^a}} \widetilde{OD}_t = \sum_{\substack{t \in T \\ s_t^d = s_v^d, s_t^a = s_v^a}} OD_t \qquad \forall v \in V \tag{5.8}$$

$$\sum_{\substack{t' \in T : \tau_{t'}^a \leq \tau_t^a \\ s_{t'}^d = s_t^d, s_{t'}^a = s_t^a}} (\widetilde{OD}_{t'} - OD_{t'}) \geq 0 \qquad \forall t \in T \tag{5.9}$$

Constraints (5.6) state that the passenger flow on trip $t \in T$ is equal to the incoming demand, plus the passenger flow on the previous trip, minus the amount of passengers that get off the train at the end of the previous trip, plus the capacity shortage of the previous trip, and finally minus the number of passengers that do not fit in the composition appointed to the trip. Constraints (5.7) make sure that the passenger flow can not exceed the appointed capacity for the trip. Constraints (5.8) denote that all passengers have to arrive at their planned destination before the end of the planning horizon. Constraints (5.9) state that none of the passengers can arrive at their destination before they originally planned to arrive, because the number of outgoing passengers cannot exceed the number of planned outgoing passengers.

Both simplifying assumptions (1. and 2.) are satisfied with these constraints. First of all assumption (1.) is satisfied, because $\widetilde{ID}_t$ is a parameter. For the second assumption, let us assume that passengers are able to leave the railway system at stations before reaching their destination. That means that the total number of passengers getting off at some station is larger than the scheduled number for at least a single station, because not only passengers with this station as destination leave the railway system, but also passengers who prematurely leave the system there. As a result it holds that:

$$\sum_{\substack{t \in T \\ s_t^d = s_v^d, s_t^a = s_v^a}} (OD_t - \widetilde{OD}_t) > 0 \qquad \exists v \in V \tag{5.10}$$

However, this causes a conflict with Constraints (5.8). So, passengers can not leave the railway system before reaching their destination. The same reasoning holds for passengers taking a detour; more passengers would get off a train after a unique trip ($v \in V$), and so Equation (5.10) would hold in this case as well, which causes a conflict with Constraints (5.8).

Next to minimizing the total number of seat-shortages, we also want to minimize the total passenger delay on an aggregated level. The variable Q_t denotes the passenger delay after trip $t \in T$ and is defined as in Equation (5.11).

$$Q_t = \sum_{\substack{t_2 \in T: \tau_{t_2}^a \leq \tau_t^a \\ s_{t_2}^d = s_t^d, s_{t_2}^a = s_t^a}} (\widetilde{OD}_{t_2} - OD_{t_2}) \cdot (\tau_{t_1}^a - \tau_t^a) \qquad \forall t, t_1 \in T : \lambda_{t_1} = t \tag{5.11}$$

The first part of the equation for the passenger delay denotes the number of passengers that wanted to leave the railway system after a trip at station s_t^a, but were not yet present at s_t^a because the capacity of the train was too little. The second part of the equation represents the minimum amount of time these passengers have to wait before they can arrive at their destination by using the next train that arrives at s_t^a.

Finally we have to note that with constraints (5.8), (5.9), and with the objective of minimizing the amount of seat-shortages and the total passenger delay, it holds that $OD_t = \widetilde{OD}_t$ as long as there is enough capacity to satisfy all passenger demand. However, when there is not enough capacity available for all passenger demand on a trip, then the model decides which passengers with which destinations will board the train and which do not. This is done in such a way that the total number of seat-shortages and the total passenger delay is minimized, but this boarding strategy is not necessarily the strategy that actually takes place in real-life. Therefore, we present two extreme boarding strategies in the next section, to compare the "optimal" boarding strategy with.

5.3.3 – Boarding strategies

In real-time it is unknown which passengers will board the train and which passengers will wait for the next train to depart to their destination, when the appointed capacity to a trip is too little for the corresponding passenger demand. By using only constraints (5.6)-(5.9) the model is allowed to decide which passengers will board the train. This will be done such that the total number of seat-shortages and the total passenger delay are minimized. In reality, however, it is not likely that precisely those passengers which minimize the global objective will board the train. Unfortunately, it is not possible to precisely predict what passengers will do. For instance, if there are two passengers waiting at a platform and both of them have a different destination: station a and b. The arriving train has only room for one of the two passengers, then there is no way to tell which of the two passengers will board the train. That is why we introduce three different boarding strategies to test their influence on the rescheduled rolling stock circulation.

The results of using only constraints (5.6)-(5.9) will be compared with models that force passengers with a certain destination to board a trip. We will compare the model with two extreme cases and an average case. The first extreme case assumes that, if the appointed capacity is too little for the passenger demand, then passengers with the nearest destination to come will board the train first. The second extreme case assumes that passengers with the furthest destination on the line to come will board the train first in case the appointed capacity for a trip is too small. The third boarding strategy is in between. It forces that half of the passengers with the nearest destination will board the train and the other half that boards the train will have a different destination. In reality, however, most likely passengers with different destinations will board the train than in any of the cases.

5.3.3.1 – Nearest destination first

In the first extreme case it is assumed that passengers with the nearest destination on the line to come board the train first in case the appointed capacity is too little. This can be done by setting constraints on the number of passengers leaving the railway system after the arrival of an incoming train. At every arrival of a train at a station, we determine the amount of passengers that have that station as their destination. This can either be the same as in the original situation ($\widetilde{OD}_t$) or larger, because the demand on the previous trip exceeded the capacity of the train. We define XO_t as this number of passengers that have station s_t^a as their destination at time τ_t^a, see Equation (5.12). Here, the first sum of the right hand side of the equation denotes all passengers that want to leave the railway system at station s_t^a *up to and including* the arrival of trip $t \in T$. The second term defines the number of passengers that have actually arrived at station s_t^a *up to* the arrival of trip t. Thus, XO_t denotes the number of passengers that have station s_t^a as their destination at time τ_t^a.

$$XO_t = \sum_{\substack{t' \in T: \\ \tau_{t'}^a \leq \tau_t^a \\ s_{t'}^d = s_t^d, s_{t'}^a = s_t^a}} \widetilde{OD}_{t'} - \sum_{\substack{t' \in T: \\ \tau_{t'}^a < \tau_t^a \\ s_{t'}^d = s_t^d, s_{t'}^a = s_t^a}} OD_{t'} \qquad \forall t \in T \tag{5.12}$$

Note that $XO_t \geq 0$ due to Constraint (5.9). In order to force passengers with the earliest destination to board the train, we want to achieve that $OD_t = \min\{F_t, XO_t\}$. Remember that OD_t represents the number of passengers getting off the train at the end of trip t. Thus we want to model that at every arrival of a train at a station, either XO_t or F_t passengers leave the train. To model this, we introduce the binary variable PO_t, which equals 1 if the passenger flow on trip $t \in T$ is larger than the number of passengers that want to get out at the end of trip t, and 0 otherwise, see Equation (5.13).

$$F_t + M(1 - PO_t) \geq XO_t \geq F_t - M \cdot PO_t \qquad \forall t \in T \tag{5.13}$$

Note that M represents a large number in this inequality.

Now, adding constraints (5.12) and (5.13), together with constraints (5.14) - (5.15), ensures that either all passengers that want to get out at the end of trip $t \in T$ will leave the train, or the complete passenger flow leaves the train:

$$OD_t \geq F_t - M \cdot PO_t \qquad \forall t \in T \tag{5.14}$$

$$OD_t \geq XO_t - M \cdot (1 - PO_t) \qquad \forall t \in T \tag{5.15}$$

At every station we force the maximum number of passengers to leave the train. As a consequence, we have forced that passengers with the earliest destination have boarded the train on the previous stations.

5.3.3.2 – Furthest destination first

In the second extreme case it is assumed that passengers with the furthest destination on the line to come board the train first in case the appointed capacity is too little. This is achieved by setting constraints on the number of passengers in an incoming train continuing their journey on the line after the arrival of an incoming train. At every arrival of a train at a station we determine the number of passengers that have a further station on the line as destination. This can either be the same as in the original situation $(\widetilde{F}_t - \widetilde{OD}_t)$ or larger, due to the fact that the demand exceeded the appointed capacity of the previous train. We define this number of passengers as XC_t, as described in Equation (5.16). Here, the first term of the right hand side of the equation denotes the total number of passengers that want to continue to a station further on the line than station s^a_t *up to and including* trip $t \in T$. The second term denotes the number of passengers that have actually continued to a station further on the line *up to* trip $t \in T$. Thus, XC_t denotes the number of passengers that have a station further on the line than station s^a_t as their destination at time τ^a_t.

$$XC_t = \sum_{\substack{t' \in T: \\ \tau^a_{t'} \leq \tau^a_t \\ s^d_{t'} = s^d_t, s^a_{t'} = s^a_t}} (\widetilde{F}_{t'} - \widetilde{OD}_{t'}) - \sum_{\substack{t' \in T: \\ \tau^a_{t'} < \tau^a_t \\ s^d_{t'} = s^d_t, s^a_{t'} = s^a_t}} (F_{t'} - OD_{t'}) \qquad \forall t \in T \tag{5.16}$$

Note that $XC_t \geq 0$ due to Constraint (5.9). In order to force passengers with the latest destination to board the train, we want to achieve that: $F_t - OD_t = \min\{F_t, XC_t\}$. Here, $F_t - OD_t$ represents the number of passengers continuing their journey after the arrival of train $t \in T$ at station s^a_t. This should either be equal to XC_t or equal to the total passenger flow F_t. To model this, we introduce a binary variable PC_t that is equal to 1 if the passenger flow on trip $t \in T$ is larger than the number of passengers that want to continue their journey after trip t, and 0 otherwise, see Equation (5.17).

$$F_t + M(1 - PC_t) \geq XC_t \geq F_t - M \cdot PC_t \qquad \forall t \in T \tag{5.17}$$

In case $PC_t = 0$, it means that the number of passengers that want to continue their journey after trip t is larger than the total passenger flow on the trip. As a result, no passenger will leave the train after trip t $(OD_t = 0)$, and so $F_t - OD_t = F_t$, see Constraint (5.18). On the other hand, if PC_t is equal to 1, it means that the passenger flow is larger than the number of passengers that want to continue after trip t. So, the actual number of passengers continuing after trip t $(F_t - OD_t)$, equals the total number of passengers that want to continue their journey after trip t: XC_t. See Constraint (5.19).

$$F_t - OD_t \geq F_t - M \cdot PC_t \qquad \forall t \in T \tag{5.18}$$

$$F_t - OD_t \geq XC_t - M \cdot (1 - PC_t) \qquad \forall t \in T \tag{5.19}$$

In this way we force passengers with the latest destination to board the train at the previous stations.

5.3.3.3 – Average boarding strategy

The above two models simulate extreme instances. It is unlikely that all passengers will leave the train at a station that is not a final station. It is also unlikely that none of the passengers will leave the train at an intermediate station. Therefore, we have included a third boarding strategy. This strategy is in between the furthest and nearest destination first strategies. We assume that at most half of the passengers with the nearest destination will board the train, and, at least, the other half will have a destination further away. Consequently, we want to achieve at the end of every trip that $OD_t = \min\{\frac{1}{2}F_t, XO_t\}$. In other words, at the end of every trip $t \in T$, the number of passengers that leave the train equals all passengers that have station s_t^a as their destination if that number is smaller than half of the passenger flow in the train, or half of the passenger flow leave the train.

To this end, the binary variable PM_t equals 1 if $\frac{1}{2}F_t > XO_t$ and 0 otherwise. The constraint (5.20) makes sure that this holds true.

$$\frac{1}{2}F_t + M \cdot (1 - PM_t) > XO_t \geq \frac{1}{2}F_t - M \cdot PM_t \tag{5.20}$$

Due to this constraint it must be that $XO_t \geq \frac{1}{2}F_t$ if $PM_t = 0$. Furthermore, in case $PM_t = 1$ it must hold that $\frac{1}{2}F_t > XO_t$. Then, Constraints (5.21)-(5.24) sets the passenger flow as in our goal. Constraints (5.21) and (5.22) forces the number of outgoing passengers to be equal to $\frac{1}{2}F_t$ if $PM_t = 0$. Note that the -1 in Constraints (5.21) and the $+1$ in Constraints (5.22) are necessary due to rounding of $\frac{1}{2}F_t$. Furthermore, Constraints (5.23) and (5.24) set the number of outgoing passengers equal to XO_t is $PM_t = 1$.

$$OD_t \geq \frac{1}{2}F_t - PM_t \cdot M - 1 \qquad \forall t \in T \tag{5.21}$$

$$OD_t \leq \frac{1}{2}F_t + PM_t \cdot M + 1 \qquad \forall t \in T \tag{5.22}$$

$$OD_t \geq XO_t - (1 - PM_t) \cdot M - 1 \qquad \forall t \in T \tag{5.23}$$

$$OD_t \leq XO_t + (1 - PM_t) \cdot M + 1 \qquad \forall t \in T \tag{5.24}$$

All three boarding strategies will be compared with the model where the passengers are guided to board the train in an optimal way with respect to the global objective.

5.3.4 – Complete model

In this subsection the complete MIP model used to reschedule the rolling stock while including dead-heading trips and adjusted passenger demand is discussed. The notation of the previous subsections still holds and is extended with the following parameters and variables. Let $\mathcal{M}$ be the set of rolling stock types, then $v_m(p)$ denotes the number of train units of type $m \in \mathcal{M}$ in composition $p \in P$. Different stations have different platform lengths, as a result not all stations can cope with all possible composition lengths. To that end denote $\eta(t)$ as the set of allowed compositions on trip $t \in T$, with respect to the platform lengths at station s_t^d, s_t^a, and all the stations in between.

At the end of a trip, the composition of a train can possibly be changed, depending on the shunting rules at the station, before departing on its successive trip. A composition change consists of the composition of the incoming and outgoing trip and which units are coupled and uncoupled during the composition change. For each trip $t \in T$, $\rho(t)$ denotes the set of allowed composition changes at the end of trip t. Furthermore, p_q denotes the composition of the incoming trip of composition change $q \in \rho(t)$ and p'_q denotes the composition of the outgoing trip in the composition change. For a given composition change $q \in \rho(t)$ at the end of trip $t \in T$, $\alpha_{q,m}$ denotes the number of uncoupled units of type $m \in \mathcal{M}$ and $\beta_{q,m}$ denotes the number of coupled units of type $m \in \mathcal{M}$ in this composition change. The time at which coupling takes place at the end of trip $t \in T$ is denoted by τ_t^+ and the time at which an uncoupled unit is available after uncoupling is denoted by τ_t^-.

The available number of units $m \in \mathcal{M}$ at station $s \in S$ at the beginning of the planning period is denoted by $i_{s,m}^0$ and the desired number of available units of type $m \in \mathcal{M}$ at station $s \in S$ at the end of the planning period is given by the parameter $i_{s,m}^{\infty}$.

Finally, the following additional decision variables are used in the model:

- $X_{t,p} \in \{0, 1\}$ denotes whether composition $p \in \eta(t)$ is used on trip $t \in T$.
- $Z_{t,q} \in \{0, 1\}$ denotes whether composition change $q \in \rho(t)$ is used at the end of trip $t \in T$.
- $C_{t,m}$ and $U_{t,m} \in \mathbb{Z}_0^+$ denote the number of units $m \in \mathcal{M}$ that are coupled and uncoupled at the start and end of trip $t \in T$, respectively.
- $I_{t,m} \in \mathbb{Z}_0^+$ denotes the number of units of type $m \in \mathcal{M}$ in the inventory at station s_t^{dep} immediately after time τ_t^+.
- $I_{s,m}^{\infty} \in \mathbb{Z}_0^+$ denotes the number of units of type $m \in \mathcal{M}$ at station $s \in S$ at the end of the planning period.
- $W_{s,m}$ denotes the deviation from the desired end-of-day balance of rolling stock type $m \in \mathcal{M}$ in station $s \in S$.

- $Y_{d,m} \in \{0,1\}$ denotes whether dead-heading trip $d \in D$ is covered by a rolling stock unit of type $m \in \mathcal{M}$.

With the above variables and parameters, and the ones introduced in the previous sections, we can form the mathematical model that is used to reschedule the rolling stock during a disruption while including dead-heading trips and adjusted passenger demand. In this model, the subsets A_t, B_t, E_t and G_t are defined as:

1. $A_t = \{t' \in T : s^d_{t'} = s^d_t, \tau^d_{t'} \leq \tau^+_t\}$
2. $B_t = \{t' \in T : s^d_{t'} = s^d_t, \tau^-_{t'} \leq \tau^+_t\}$
3. $E_t = \{d \in D : s^a_d = s^d_t, \tau^a_d \leq \tau^+_t\}$
4. $G_t = \{d \in D : s^d_d = s^d_t, \tau^d_d \leq \tau^+_t\}$

This leads to the following complete model:

$$\min f(X, Z, W, Y, SS, Q) \tag{5.25}$$

Subject to:

$$\sum_{p \in \eta(t)} X_{t,p} = 1 \qquad \forall t \in T \tag{5.26}$$

$$X_{t,p} = \sum_{q \in \rho(t): p_q = p} Z_{t,q} \qquad \forall t \in T, p \in \eta(t) \tag{5.27}$$

$$X_{t,p} = \sum_{q \in \rho(\sigma(t)): p'_q = p} Z_{\sigma(t),q} \qquad \forall t \in T, p \in \eta(t) \tag{5.28}$$

$$C_{t,m} = \sum_{q \in \rho(\sigma(t))} \beta_{q,m} Z_{\sigma(t),q} \qquad \forall t \in T, m \in \mathcal{M} \tag{5.29}$$

$$U_{t,m} = \sum_{q \in \rho(t)} \alpha_{q,m} Z_{,q} \qquad \forall t \in T, m \in \mathcal{M} \tag{5.30}$$

$$\begin{aligned} I_{t,m} = \quad & i^0_{s^d_t,m} - \sum_{t' \in A_t} C_{t',m} + \sum_{t' \in B_t} U_{t',m} \\ & - \sum_{d \in G_t} Y_{d,m} + \sum_{d \in E_t} Y_{d,m} \qquad \forall t \in T, m \in \mathcal{M} \end{aligned} \tag{5.31}$$

$$\begin{aligned} I^\infty_{s,m} = \quad & i^0_{s,m} - \sum_{t \in T, s^d_t = s} C_{t,m} + \sum_{t \in T, s^a_t = s} U_{t,m} \\ & - \sum_{d \in D, s^d_d = s} Y_{d,m} + \sum_{d \in D, s^a_d = s} Y_{d,m} \qquad \forall s \in S, m \in \mathcal{M} \end{aligned} \tag{5.32}$$

$$I^{\infty}_{s,m} = i^{\infty}_{s,m} + W_{s,m} \qquad \forall s \in S, m \in \mathcal{M} \qquad (5.33)$$

$$F_t + SS_t = \widetilde{ID}_t + F_{\sigma(t)} - OD_{\sigma(t)} + SS_{\lambda(t)} \qquad \forall t \in T \qquad (5.34)$$

$$F_t \leq \sum_{p \in P} X_{t,p} cap_p \qquad \forall t \in T \qquad (5.35)$$

$$\sum_{\substack{t \in T \\ s^d_t = s^d_v, s^a_t = s^a_v}} \widetilde{OD}_t = \sum_{\substack{t \in T \\ s^d_t = s^d_v, s^a_t = s^a_v}} OD_t \qquad \forall v \in V \qquad (5.36)$$

$$\sum_{\substack{t_1 \in T : \tau^a_{t_1} \leq \tau^a_t \\ s^d_{t_1} = s^d_v, s^a_{t_1} = s^a_v}} (\widetilde{OD}_{t_1} - OD_{t_1}) \geq 0 \qquad \forall t \in T, v \in V \qquad (5.37)$$

$$Q_t = \sum_{\substack{t_2 \in T : \tau^a_{t_2} \leq \tau^a_t \\ s^d_{t_2} = s^d_t, s^a_{t_2} = s^a_t}} (\widetilde{OD}_{t_2} - OD_{t_2}) \cdot (\tau^a_{t_1} - \tau^a_t) \qquad \forall t, t_1 \in T : \lambda_{t_1} = t \qquad (5.38)$$

$$X_{t,p} \in \{0, 1\} \qquad \forall t \in T, p \in \eta(t) \qquad (5.39)$$

$$C_{t,m}, U_{t,m}, I_{t,m} \in \mathbb{R}_+ \qquad \forall t \in T, m \in \mathcal{M} \qquad (5.40)$$

$$I^{\infty}_{s,m} \in \mathbb{R}_+, W_{s,m} \in \mathbb{R} \qquad \forall s \in S, m \in \mathcal{M} \qquad (5.41)$$

$$Z_{t,q} \in \mathbb{R}_+ \qquad \forall t \in T, q \in \rho(t) \qquad (5.42)$$

$$Y_{d,m} \in \mathbb{R}_+ \qquad \forall d \in D, m \in \mathcal{M} \qquad (5.43)$$

$$F_t, SS_t, OD_t, Q_t \in \mathbb{R}_+ \qquad \forall t \in T \qquad (5.44)$$

Constraints (5.26) specify that to each trip exactly one composition is assigned. All compositions before and at the start of the disruption are fixed, because these trips are already underway. In that case, the set of allowed compositions, $\eta(t)$, only consists of one composition. Note that a trip is cancelled if the empty composition is appointed to it. Constraints (5.27) state that if composition $p \in \eta(t)$ is assigned to trip $t \in T$, then only a composition that can originate from p can be assigned to the succeeding trip. Constraints (5.28) state that if composition p is assigned to trip t, then only a composition which can be changed into p can be assigned to the predecessor trip $\sigma(t)$.

Constraints (5.29) specify the number of coupled train units at the beginning of a trip and Constraints (5.30) specify the number of uncoupled train units at the end of a trip.

Constraints (5.31) stipulate the inventory of rolling stock type $m \in \mathcal{M}$ at station s^d_t immediately after time τ^d_t. This is equal to the start inventory at the associated station, minus all train units that are coupled up to time τ_t^d, plus all uncoupled units that are available before time τ_t^d, plus all empty units that entered the station before time τ_t^d due to dead-heading trips, and minus all empty units that departed from the station up to time τ_t^d due to dead-heading trips.

Constraints (5.32) define that the end inventory at a station $s \in S$ is equal to the start inventory at s minus all train units that are coupled at s, plus all train units that are uncoupled at s, plus all empty units that are transferred to s, and minus all empty train units that are transferred from s to another station. Constraints (5.33) states that the end-of-day balance is equal to the planned end-of-day balance plus a deviation.

Constraints (5.34)-(5.38) set the passenger flows as explained in the previous Section 5.3.2.

The objective function (5.25) is a function that depends on which composition is assigned to a trip (X), what composition changes are made (Z), the deviation from the end-of-day balance at stations (W), the number of dead-heading trips that are scheduled (Y), the total number of seat shortages (SS), and the total passenger delay (Q). The decision maker should state, in advance, the costs of having deviations from the end-of-day balance at a station, of having not enough capacity for all passengers on a trip, of having different composition changes than in the original schedule, of sending dead-heading trips, and of having too little capacity. The objective is a trade-off between passenger service and costs.

Constraints (5.39)-(5.44) denote the variable domains.

5.4 – COMPUTATIONAL TESTS

In this section we give an overview of the computational results of the various instances on which we have tested our models. All computational tests are performed on an Intel (R) Core (ITM) i5-3210M processor with 2.50 GHz and 8GB RAM by using CPLEX 12.6.1.

5.4.1 – Variants of the model

The complete model, as presented in Section 5.3.4, was tested in seven versions:

1. Original Model (*OM*): no dead-heading trips and no adjusted passenger demand have been included.
2. Dead-Heading Model (*DHM*): dead-heading trips have been included, but no adjusted passenger demand has been included.
3. Adjusted Demand Model (*ADM*): no dead-heading trips have been included, but adjusted passenger demand has been included.

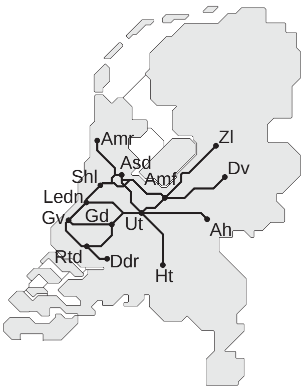
4. Earliest Destination Board Model (*ADM(E)*): no dead-heading trips have been included, but adjusted passenger demand has been included, with the addition that passengers with the nearest destination are assumed to board the train (see Section 5.3.3.1).
5. Latest Destination Board Model (*ADM(L)*): no dead-heading trips have been included, but adjusted passenger demand has been included, with the addition that passengers with the furthest destination are assumed to board the train (see Section 5.3.3.2).
6. Average Destination Board Model (*ADM(A)*): no dead-heading trips have been included, but adjusted passenger demand has been included, with the addition that half of the passengers with the nearest destination are assumed to board the train (see Section 5.3.3.3).
7. Dead-Heading Adjusted Demand Model (*DHADM*): both dead-heading trips and adjusted passenger demand are taken into account.

We have not included the three versions where boarding strategies and dead-heading trips are combined. This is because those three versions did not lead to additional conclusions: dead-heading trips are used to decrease the number of cancelled trips, and the extreme boarding strategies lead to many seat-shortages.

All seven versions are tested on real life instances from NS. In Section 5.4.2 all computational experiments used to test the models are explained in detail. Thereafter, in Section 5.4.3, the results of applying the models to the instances are discussed. The results of the models that make use of adjusted passenger demand are difficult to compare with the results of the models that do not include adjusted passenger demand. Consequently, we split the comparison of the results. In Section 5.4.3.2 we start with investigating the added value of using dead-heading trips by comparing *OM* with *DHM*. Following, in Section 5.4.3.3, we compare all models where adjusted passenger demand is taken into account: *ADM*, *ADM(L)*, *ADM(E)*, *ADM(A)*, and $DHADM$. Finally, in Section 5.4.3.4, all models are compared with each other with respect to their computation times.

5.4.2 – Case description

Figure 5.5 gives an overview of the railway lines that are included in our case study. All important and busy lines of the Western part of the Netherlands in 2012 are taken into account. In this way a large part of the complete Dutch railway network is covered by our instances. The data set consists of 2276 different trips. Two different rolling stock types are used on these trips. They differ in their number of carriages (either 4 or 6). By using these rolling stock types, in total 11 different compositions (including the empty one) are available.



Line	Stations	Frequency
500	Gv Gd Ut Amf Zl	hourly
700	Shl Amf Zl	hourly
800	Amr Zd Asd Ut Ht	half hourly
1500	Asd Amf Dv	half hourly
1600	Shl Amf Dv	hourly
1700	Gv Gd Ut Amf Dv	hourly
1900	Gv Rtd Ddr	half hourly
2000	Gv Gd Ut Ah	half hourly
2100	Asd Shl Ledn Gv Rtd Ddr	half hourly
2600	Asd Shl Ledn Gv	half hourly
2800	Rtd Gd Ut Amf	half hourly
3000	Amr Asd Ut Ah	half hourly
3500	Shl Ut Ht	half hourly
8800	Ledn Ut	half hourly
20500	Rtd Gd Ut	hourly
21700	Rtd Gd Ut	hourly

Figure 5.5: The NS network considered in the test instances.

Besides the timetable of a large part of the Dutch railway network, NS provided us with information regarding the passenger demand. For every scheduled trip we are given the expected passenger demand and the expected percentage of passengers getting off at the end of that trip. As a consequence, we can determine the incoming demand on every trip.

The models have been developed to be used in real-time. In order to test them adequately, we simulate different disruption scenarios. These scenarios are summarized in Table 5.1. As can be seen, there are 12 different disruptions simulated at different locations and different time slots on the railway network.

Case	Disruped area	Disruption Time		
1, 2, 3	Gd - Ut	07:00-10:00	16:00-19:00	11:00-15:00
4, 5	Rtd - Gv	16:00-19:00	11:00-15:00	
6	Ledn - Ut	16:00-19:00		
7, 8	Amf - Ut	16:00-19:00	11:00-15:00	
9, 10	Gv - Ledn	16:00-19:00	11:00-15:00	
11, 12	Asd - Ut	16:00-19:00	11:00-15:00	

Table 5.1: Disruption cases

Besides the simulated disruptions, also the parameter settings are of importance for testing the models. There is a trade-off between the different objective components. For instance, reducing the number of seat-shortages for passengers leads to an increase in the number of carriage kilometers. Consequently, the penalty settings will influence the results of the RSRP. We will test different penalty settings for every disruption case and every model. First of all, cancelling additional trips must be prevented, so the penalty for cancelling a trip is always the largest. We use a penalty value of 1 000 000 in half of the settings. We like to investigate the trade off between cancelling trips and the other objective components, so we use a penalty of 100 000 for the other half of the settings.

Cancel trip	End Dev	SS km	Carr km	Dead-heading	Shunting (Unplanned, Cancelled)
1000000 100000	1000	1	0,1	10000	(1000, 100)
	2000	0.1		5000	(2000, 200)
	5000	0.5		2000	(5000, 500)
		2		1000	
		5		200	

Table 5.2: Summary of the different penalty settings. The first column denotes the penalty values for cancelling an additional trip. The second column for having a negative deviation from the scheduled end of day balance. The third and fourth column for the seat-shortages per kilometer and the carriage kilometers. The fifth column shows the penalty values for using a dead-heading trip. Finally, the sixth column denotes the penalty values for having an unplanned or cancelled shunting movement.

With respect to the other objective coefficients, the following penalties are used. First of all, a penalty of 1000 is used for a single negative deviation from the scheduled end-of-day balance. Negative deviations from the scheduled end-of-day balance are solved during the night by using dead-heading trips. This is expensive for the railway operator and these operations need therefore to be minimized. However, it is more important to have a good passenger service, so the penalty is not that large. Furthermore, penalties are set upon having seat shortage kilometers and on driving a single carriage for one kilometer. For the first, a general penalty of 1 is used and for the latter a general penalty value of 0.1 is imposed. The penalties for seat shortages and carriages kilometers seem to be small, however, they are measured per kilometer. The shortest trip has a length of 16km and the longest trip has a length of 67km. As a result, having, for example, 100 seat-shortages on a trip with length 50km already leads to a penalty of $100 \cdot 50 \cdot 1 = 5000$. The same holds for the carriage kilometers, this is measured per carriage per kilometer. Note that the smallest composition already contains 4 carriages. Thirdly, the penalty for sending a single dead-heading trip during the day equals 2000 and the penalty for a single minute of passenger delay equals 1. The final two penalty values are for performing a shunting operation at a location and time where no shunting operation was scheduled, and for *not* doing a shunting operation at a location and time where a shunting operation was scheduled. An unplanned shunting movement means that a crew member must be appointed to the corresponding station to perform the shunting movement, this takes time and costs manpower. On the other hand, cancelling a shunting movement means that a crew member is unnecessarily appointed, and this has certain costs as well. We use a penalty value of 1000 for adding a new shunting operation and of 100 for cancelling a shunting operation.

In order to test the influence of the penalty settings, the models are tested on different settings as well. In every of those settings, the penalty value of one of the objective coefficients is changed, while all other penalty values are the same as in the general setting. First of all, the penalty value for a single negative deviation from the scheduled end-of-day balance has been increased to 2000 and 5000. Secondly, the penalty for having a seat shortage kilometer is changed to either 0.1, 0.5, 2, or 5. The carriage kilometer penalty is always kept the same, because only the ratio $\frac{SSkm}{Carrkm}$ is of importance. Thirdly, the penalties for scheduling a single dead-heading trip during the day have changed to 200, 1000, 5000, and 10000. Finally, the penalty settings for (unplanned, cancelled) shunting activities are changed to (2000, 200) and (5000, 500).

This leads to a total of 26 different penalty settings. A short summary of the penalty values is given in Table 5.2. Here, the two general penalty settings are shown in the top row. All other penalty coefficients are shown beneath.

5.4.3 – Results

As mentioned before, it is not correct to compare the results of the models that include adjusted passenger demand with the models that do not include adjusted passenger demand. Therefore, we start with giving an example of the difference between the models using adjusted passenger demand and the models that do not in Section 5.4.3.1. Thereafter, we compare the two versions *OM* and *DHM* with each other in Section 5.4.3.2. In Section 5.4.3.3 we compare the five versions *ADM*, *ADM(L)*, *ADM(E)*, *ADM(A)*, and $DHADM$.

5.4.3.1 – Difference example

We use disruption case 1 (see Table 5.1) as the example to explain the difference between the models that use adjusted passenger demand and the models that do not. A disruption between Gouda (*Gd*) and Utrecht (*Ut*) takes place from 07:00-11:00 and as a consequence the rolling stock circulation needs to be rescheduled for the remainder of the day.

We focus on the difference by rescheduling the rolling stock circulation with the original model (*OM*) and the adjusted demand model (*ADM*). Figure 5.6 presents the passenger demand used by the *OM* (denoted by *OM* in the Figure), and the adjusted passenger demand used during rescheduling with *ADM* (denoted by *ADM*) for the trajectory Gouda-Utrecht of the 500 line. The time during the day is displayed on the horizontal axis. Note that we only display the passenger demand in a single direction, so no passenger demand is displayed for the direction Utrecht-Gouda. Furthermore, we only show the passenger demand for the trips on the 500 line. However, this is not the only line with a connection between Gouda and Utrecht.

The other lines are taken into account when determining the adjusted demand between the stations.

As can be seen, the adjusted passenger demand and the original demand are equal as long as everything runs according to plan. However, after the disruption, there is a large peak for the adjusted demand, while the original passenger demand remains constant. In order to reduce this large peak in *ADM*, large compositions have to be appointed to the trips just after the disruption. This can be seen in Figure 5.7, where the appointed capacity after rescheduling with *ADM* and *OM* are shown for the 500 line between Gouda and Utrecht. After rescheduling with *OM* there are no larger compositions appointed to the trips just after the disruption, while after rescheduling with *ADM* larger compositions are actually appointed to these trips. As a consequence, it takes approximately eight trips on the 500 line before the adjusted demand is equal to the original passenger demand again. Note that on other trips of other lines between Gouda-Utrecht also large compositions are appointed with *ADM*.

5.4.3.2 – Dead-heading trips

By comparing the original model with the model that includes dead-heading trips we can emphasize the added value of using dead-heading trips. To this end, we will compare the two models based on their objective values and on the number of cancelled trips in this subsection. When the solution of *DHM* does not use any dead-heading trips, it means that its objective value is equal to the objective value of *OM*.

Table 5.3 gives an overview of the results of both models. As can be seen, in cases 1, 2, and 3 there is always one additional trip cancelled in *OM*, while there are no trips cancelled when rescheduling with *DHM*. This is due to the fact that dead-heading trips are used in these cases to overcome a shortage of rolling stock units at Ut just after the disruption. As a consequence, the average objective value of *DHM* is smaller than the average objective value of *OM*. In cases 4, 5, 8, 9, 11, and 12 there are no differences between *OM* and *DHM* and in the cases, 6, 7, and 10 there is only a small difference between the models. Here, dead-heading trips are used to either decrease the number of unplanned shunting movements, the negative end of day deviations, or the number of seat-shortages. These dead-heading trips are only scheduled if the benefits outweight the costs. So, when dead-heading trips are cheaper, they are used more.

There is no difference between the instances with a penalty value of 1 000 000 or 100 000 for cancelling an additional trip. Both *OM* and *DHM* always prevent trips from getting cancelled, if this is possible. Changing the other objective coefficients does not influence the results much, it slightly alters the optimal solutions.

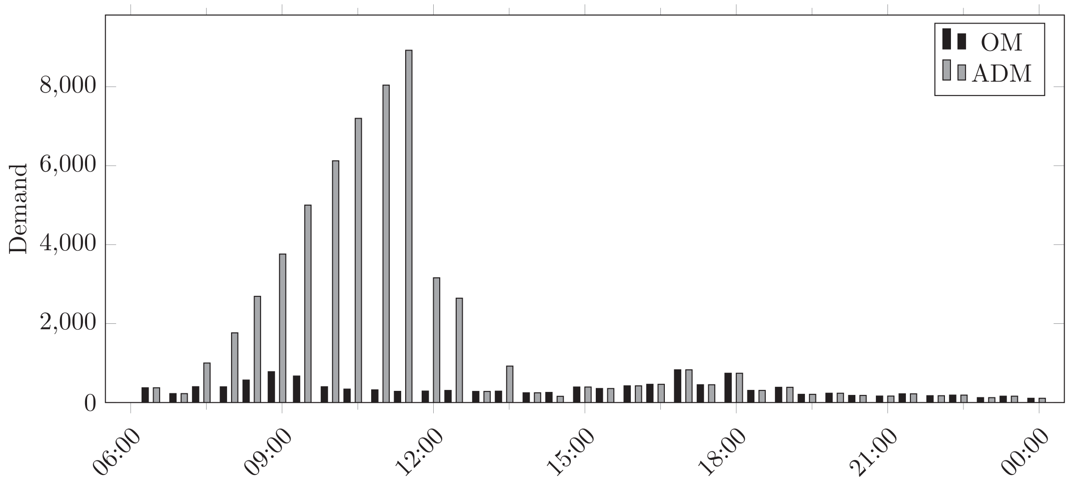


Figure 5.6: Passenger demand

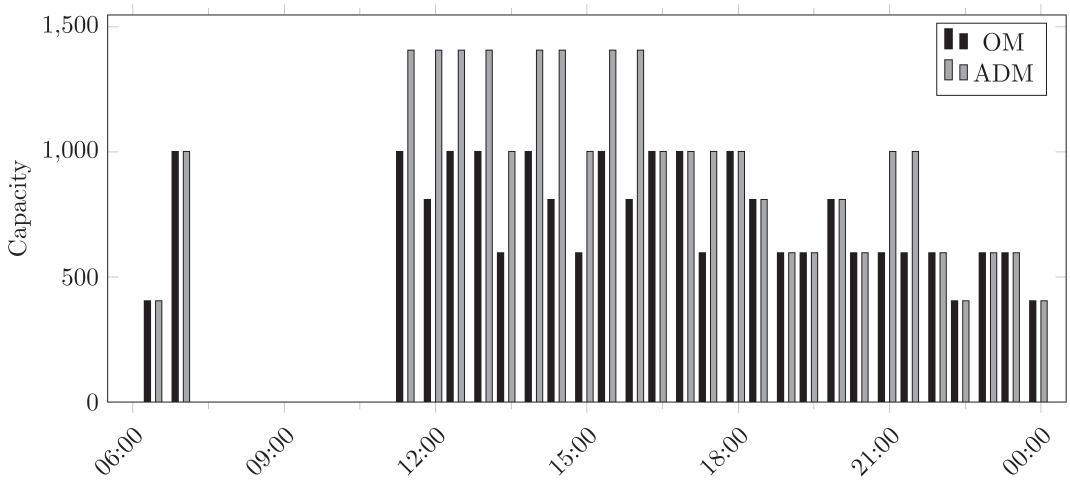


Figure 5.7: Capacity

Summarizing the results of *OM* and *DHM*, we can conclude that dead-heading trips are important to prevent additional trips from getting cancelled. In this way the passenger service increases.

5.4.3.3 – Adjusted passenger demand

In this section we present the results of the models that include adjusted passenger demand: *ADM*, *ADM(L)*, *ADM(E)*, *ADM(A)*, and *DHADM*. We compare these models based on the number of cancelled trips and on the passenger service (seat-shortages and passenger delay).

Tables 5.4, 5.5, and 5.6 give an overview of the results for these models. The first conclusion we can draw from the tables is that $DHADM$ performs best in terms of cancelling the least amount of trips, the number of seat shortages, and the total passenger delay. *DHADM* takes

	Objective		# Cancelled		
Case	*DHM*	*OM*	*DHM*	*OM*	D-H trips
1	88861	639265	0	1	1.48
2	132122	675162	0	1	1.92
3	96861	637611	0	1	1.44
4	93567	93567	0	0	0
5	90430	90430	0	0	0
6	88379	88401	0	0	0.16
7	95609	95715	0	0	0.52
8	91537	91537	0	0	0
9	113934	113934	0	0	0
10	91186	91215	0	0	0.16
11	98490	98490	0	0	0
12	90968	90968	0	0	0

Table 5.3: Results of *DHM* and *OM*. The first column denotes the disruption instance. The second and third column represent the average objective value over the 26 different penalty settings. The fourth and fifth column show the average number of cancelled trips over the 26 different penalty settings. Finally, the sixth column denotes the average number of dead-heading trips used in the 26 different penalty settings in *DHM*.

next to adjusted passenger demand also dead-heading trips into account. So, in the *DHADM*, dead-heading trips are used to either decrease the number of cancelled trips, the number of seat-shortages, or the passenger delay. The gains of scheduling a dead-heading trip are worth the costs in all instances.

Secondly, there are more seat-shortages when rescheduling with *ADM(L)* than with *ADM(E)*. *ADM(L)* boards passengers with the furthest destination away, while *ADM(E)* boards passengers with the nearest destination. Consequently, seats are sooner available if passengers leave the train earlier. So, with respect to the number of seat shortages, it is better for the railway operator if passengers who need to get off the train first also board first. The *ADM(A)* strategy is less extreme, and most likely more realistic. A strategy in between both extreme strategies leads to less seat-shortages. However, a guided boarding strategy is not optimal. The number of seat shortages is much lower after rescheduling with *ADM*. The *ADM* decides which passengers board the train depending on the global objective. To summarize, in the worst case (*ADM(L)*) there will be many more seat-shortages than necessary, because the "wrong" passengers board the train first.

Thirdly, it holds for all models that sometimes more trips are cancelled than absolutely necessary, which can be deduced from the fact that more trips get cancelled than after rescheduling the instances with *OM*. This means that some trips with small passenger demand are cancelled. As a consequence, the rolling stock units of the cancelled trips can be used on the trips with a large passenger demand. In this way, there is more capacity available on the very busy lines during a disruption. Thus, by cancelling trips with small passenger demand, the total number of seat-shortages is reduced. This might not be ideal in practice. Therefore, we introduced an additional case where the penalty for cancelling a train is increased, see Section 5.4.3.3.1.

Case	Cancelled trips					Dead-heading
	ADM	ADM(L)	ADM(E)	ADM(A)	DHADM	DHADM
1	2.68	2.64	3.08	2.57	1.56	14.60
2	2.16	2.12	2.24	2.19	1.16	9.36
3	2	2.08	2	2	0.96	12.52
4	0	0	0	0	0	2.08
5	0	0	0	0	0	3.96
6	0	0	0	0	0	2.12
7	0	0	0	0	0	7.24
8	0	0	0	0	0	2.88
9	0	0	0	0	0	2.88
10	0	0	0	0	0	1.68
11	0.44	0.08	0.12	0.46	0	5.96
12	0.48	0.52	0.39	0.59	0	7.00

Table 5.4: Results of the models taking adjusted demand into account: the first five columns denote the average number of cancelled trips for all models over the 26 parameter settings. The last column shows the average number of dead-heading trips used in $DHADM$.

Case	Seat shortages				
	ADM	ADM(L)	ADM(E)	ADM(A)	DHADM
1	168578	235812	187944	176740	166577
2	112461	158350	123798	119391	111798
3	60837	89343	69687	65827	59589
4	44496	54024	47843	45408	44161
5	24424	32031	27970	25525	22431
6	3854	3854	3854	3871	3854
7	25108	42344	35628	28236	23362
8	24823	43543	35987	29016	25086
9	52657	75380	61742	58028	52464
10	23417	32330	30101	25284	23389
11	69537	82434	78972	71702	66368
12	49559	56183	54097	53786	45345

Table 5.5: Results of the models taking adjusted demand into account. The last five columns denote the average number of seat-shortages.

Case	Passenger delay				
	ADM	ADM(L)	ADM(E)	ADM(A)	DHADM
1	119940	218253	165132	126891	118101
2	84846	151030	111267	90721	83882
3	44754	86402	59823	48226	44406
4	45196	54025	50199	45801	44846
5	25108	32031	29816	25827	23064
6	3824	3824	3824	3824	3824
7	21503	39390	30087	22475	20068
8	22776	40965	31233	23578	22759
9	53517	75380	60189	58244	53313
10	23808	32329	26781	25563	23746
11	64848	83307	78615	68681	62379
12	50211	58590	51678	54347	46203

Table 5.6: Results of the models taking adjusted demand into account. The last five columns show the average passenger delay for all models.

The penalty for cancelling a trip is important in the models with adjusted passenger demand. With a penalty value of 1 000 000 on average only 0.27 trips get cancelled with *ADM*, while with a penalty value of 100 000 on average 0.92 trips get cancelled. However, as a consequence, with a penalty value of 1 000 000 there are on average 57768 seat shortage kilometers, while there are only 52406 seat shortage kilometers on average with a penalty of 100 000. The rolling stock units that were used on the cancelled trips can be used on other trips to increase the capacity there. So, a trade off must be made between cancelling trips and the number of seat shortages.

To summarize, adjusted passenger demand has a large influence on the results. The demand does not decrease, and, as a result, the trade-off between cancelling additional trips and using larger compositions on other trips is no longer evident. If passengers board in the "worst" case scenario, as in *ADM(L)*, then the seat-shortages will be much larger than if the passenger follow the "optimal" boarding strategy as in *ADM*. Finally, dead-heading trips can be used to reduce the number of seat-shortages and the number of additional cancelled trips.

5.4.3.3.1 – Cancelling less trains

The rescheduling solutions when using adjusted passenger demand cancel more trains than necessary. This is a solution that is not ideal in practice. Therefore, we have increased the penalty value for cancelling a trip to 10 000 000 in order to test the models with adjusted passenger demand while cancelling the least amount of trips as possible. The four models ADM, $ADM(L)$, $ADM(E)$, and $ADM(A)$ cancel now as many trips as OM, while the model $DHADM$ now cancels just as many trips as DHM. Tables 5.7, 5.8, and 5.9 give an overview of the average number of cancelled trains, the average number of seat shortages, and the average passenger delays for the models when using different penalty values for cancelling a trip. As can be seen, the larger the penalty for cancelling a trip, the less trips get cancelled, but the more seat-shortages and/ or passenger delays there will be.

Cancel	Cancelled trips					Dead-heading trips
penalty	ADM	ADM(L)	ADM(E)	ADM(A)	DHADM	DHADM
10 000 000	0.25	0.25	0.25	0.25	0	6.1
1 000 000	0.27	0.31	0.28	0.27	0.03	5.9
100 000	0.92	0.94	0.94	0.93	0.56	5.8

Table 5.7: Results of the models with different penalty values for cancelling a trip. The first five columns denote the average number of cancelled trips for all models. The last column denotes the average number of dead-heading trips scheduled by $DHADM$.

Cancel	Seat shortages				
penalty	ADM	ADM(L)	ADM(E)	ADM(A)	DHADM
10 000 000	58179	79905	65955	62122	57273
1 000 000	57768	78868	65209	60320	57284
100 000	52406	72150	60088	56781	50292

Table 5.8: Results of the models with different penalty values for cancelling a trip. The last five columns denote the average number of seat-shortages.

Cancel	Passenger delay				
penalty	ADM	ADM(L)	ADM(E)	ADM(A)	DHADM
10 000 000	48700	76637	53046	49812	47346
1 000 000	48511	75828	52918	49017	47172
100 000	45021	70093	51623	46171	43926

Table 5.9: Results of the models with different penalty values for cancelling a trip. The last five columns denote the average passenger delay for all models.

5.4.3.4 – Computation times

The models need to be fast in order to be useful in real-time. Therefore, all seven models are compared based on their computation times. Figure 5.8 shows the average computation time per case per model. The computation times for both *ADM(L)*, *ADM(E)*, and *ADM(A)* are much larger than the acceptable norm of 300 seconds, so only the computation times of the other models are shown in the figure. The *OM* is fastest, however *DHM* is not much slower.

On average *OM* takes 22,8 seconds to solve an instance, *DHM* 28,3 seconds, $DHADM$ 59,2 seconds, *ADM* 73,9 seconds, *ADM(L)* 360,5 seconds, *ADM(E)* 621,2 seconds, and *ADM(A)* 490,6 seconds.

We can conclude that *OM*, *DHM*, $DHADM$, and *ADM* all have computation times that are acceptable for usage in practice.

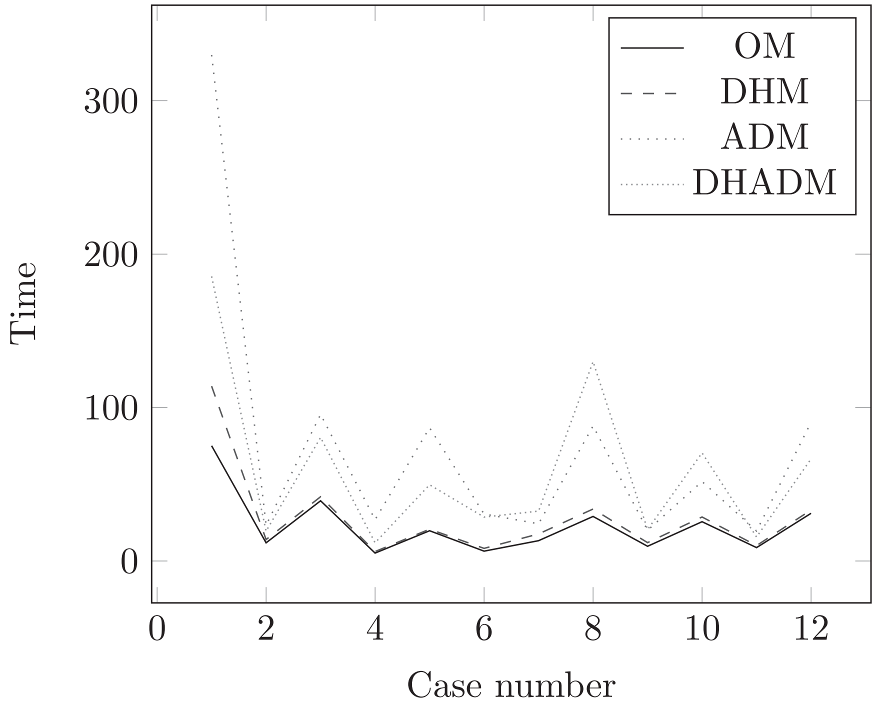


Figure 5.8: Computation times per model

5.5 – CONCLUSION

In current literature, models are developed to tackle the Rolling Stock Rescheduling Problem. However, these models have not been applied in practice yet, because not all practical aspects have been taken into account. In this paper we included two of these practical aspects in the rolling stock rescheduling model.

First of all, we introduced the possibility of scheduling dead-heading trips from a station with an excess of inventory to a station with a shortage of inventory. These trips are called dead-heading trips and can be used to reduce the number of cancelled trips.

Secondly, adjusted passenger demand is taken into account in the model. Passengers stay in the railway system until they arrive at their destination. As a consequence, trains with more capacity will be appointed to the trips where the actual passenger demand is large.

Six different rolling stock rescheduling model versions have been tested. Results show that by using dead-heading trips the number of additional cancelled trips and the number of seat-shortages are reduced in comparison with the model versions where no dead-heading trips are included. Furthermore, the model versions where adjusted passenger demand is taken into account have appointed trains with more capacity to the trips just after the end of the disruption. This is due to the fact that the adjusted passenger demand is there the largest, and appointing trains with a large capacity thus reduces the number of seat-shortages. Finally, the computation times are applicable in practice for the models where dead-heading trips and adjusted passenger demand are included.

There are interesting research possibilities for further research. First of all, results showed that if passengers with a nearer destination board the train first when there is not enough capacity for all passengers to board the train, this leads to a better global objective than if passengers with a further destination board the train first. As a consequence, it might be interesting to investigate the usage of trains skipping stations, such that passengers with a further destination board those trains instead of claiming the capacity of the other trains. In this way both passengers with a nearer destination and with a further destination might become satisfied. Secondly, the assumption that passengers do not leave the railway system prematurely and do not take a detour to their destination should be relaxed in further research. Finally, other practical aspects need to be included in the Rolling Stock Rescheduling Problem as well. For instance, rolling stock units that have a maintenance appointment somewhere during the day should be included in the model together with adjusted passenger demand and dead-heading trips.

6 – A Comparison of Optimization Methods for Solving the Depot Matching and Parking Problem

This chapter considers the paper (Haahr et al. (2015b)) which is under review at Transportation Science. In 2015 a preliminary version of this paper has won the second place in the Student Paper Award Competition of the Railway Application Section of INFORMS.

Co-authors: J.T. Haahr and R.M. Lusby

6.1 – INTRODUCTION

Passenger railways are an important mode of transportation in many countries. Travelers depend on a safe, reliable and timely service. For a rail operator, providing this service requires the careful planning of trains, personnel and infrastructure. Planning problems that must be addressed include, but are not limited to the following: *i)* determining a timetable, which stipulates the arrival and departure times of the train services to operate; *ii)* creating a rolling stock schedule, specifying a feasible fleet circulation by assigning train units to timetabled train services; and *iii)* constructing a crew plan by assigning certified train personnel to operate the trains. Due to the complexity of each of the underlying optimization problems, the planning problems are usually solved sequentially and in isolation.

In this paper, we present a comparison of optimization methods for determining whether a scheduled set of *shunting movements* at a given depot is feasible with respect to the layout of the depot. A depot (or shunting yard) is a storage facility that is usually located in the close vicinity of a railway station. It typically consists of several parallel tracks on which train units not in service can be parked. A shunting movement refers to the process of driving a unit to (or from) a depot track from (or to) a platform in the station and is induced whenever the train *composition* changes on successive train services. Multiple units can be assigned to the same train service (i.e. coupled in a convoy), and this ordered collection of units is termed a composition. Compositions are typically designed to meet the forecast passenger demand, while not using more train units than is necessary. Composition changes occur when, for instance, a train unit is either taken out of service or a train unit is brought into service. In the first case, the uncoupled unit must be shunted to the station's depot where it is parked and awaits its next service. In the second case a train unit must be retrieved from the depot and coupled to a train service.

The rolling stock scheduling problem determines which train unit *types* to assign to each timetabled train service. Two train unit types typically differ in their respective physical characteristics, e.g. length and passenger capacity. We assume train units of the same type to be interchangeable. Shunting movements are usually not considered when planning the rolling stock schedule. It is often assumed that these can be resolved in a later phase. The assumption is that the capacity as well as the infrastructure layout of any depot on the network is sufficient to cater for the necessary shunting movements implied by the rolling stock schedule. However, for railway networks where depot capacity is scarce, it may not always be possible to feasibly perform the resulting shunting movements. Furthermore, the planning of shunting movements is not a trivial problem. This is especially true at larger stations where many shunting movements may occur over the course of a day and where there can be a number of depot tracks of different lengths. Effective methods for finding feasible solutions to, or proving the infeasibility of, the so-called Train Unit Shunting Problem (TUSP) are therefore essential.

A railway network normally contains several depots, and the set of shunting movements at each of them can be deduced from a solution to the rolling stock scheduling problem. Scheduling the rolling stock is beyond the scope of this paper, and we assume a solution to this problem is available as input to the TUSP. From the given rolling stock schedule all arrivals at and departures from depots are implicitly specified; an uncoupled train unit corresponds to an arrival at a depot, while a coupled train unit corresponds to a departure from a depot. Such arrival or departure events are associated with a specific train unit type. We assume that the depots of two different stations are independent of each other. In other words, the set of shunting movements at one depot is confined to that depot and has no impact on the shunting movements of another depot. We term a feasible solution to the TUSP a *shunting plan*, and this must satisfy two constraints. First, the total length (or capacity) of each individual depot track must not be violated at any given time; and second, no train unit ordering *conflicts* occur. A conflict occurs when the arrival of a train unit at a depot track blocks the departure of a train unit from the same track. In this paper, we assume that depot tracks function as last-in first-out (LIFO) stacks. This means that the last train unit to arrive at a depot track must be the first to leave. In reality, open ended depot tracks exist; however, such cases will also have ordering restrictions that must be obeyed. In this paper, any open-ended track is operated as a LIFO stack.

The TUSP is essentially comprised of two interdependent subproblems. A *matching* problem pairs the arrivals and departures at a depot. This is necessary as only train unit types are known in the solution to the rolling stock problem. The resulting matching implicitly specifies how long individual units will spend in the depot and a track allocation problem or *parking* must be solved to identify on which track the units will be parked. Contrary to previous work in this field, we view the TUSP as a feasibility problem as the primary goal is to determine whether or not a given rolling stock schedule is feasible from a shunting perspective. Almost all of the operational cost is incurred in the rolling schedule and this schedule is unlikely to change in order to improve the combined objective of the TUSPs. It is important to simply know whether the generated shunting movements are feasible with respect to the depots. We note that after confirming feasibility of the shunting movements, the TUSPs could be resolved with an appropriate objective function. In the absence of a feasible solution, proving that no solution exists is equally important. If no solution exists, it means that the given rolling stock schedule is not feasible from a shunting perspective. As a consequence, in such situations, a new rolling stock schedule must be found. The emphasis in this paper is therefore on determining whether or not a solution to an instance of the TUSP exists.

In this paper we propose three new methods for solving the TUSP and benchmark their performance against several methods from the literature on both realistic and artificial instances. In particular we present a Constraint Program (CP) approach, a column generation procedure, and a greedy randomized heuristic approach. To our knowledge, CP has never been applied to the TUSP despite the fact that the methodology is extremely effective at solving feasibility problems. These are compared against a reference Mixed Integer Program (MIP) and a two stage approach from the literature. For the reference MIP we also experiment with a delayed constraint variant. The devised methods cover a broad range of optimization techniques. Some of the proposed approaches are exact solution methods, which find one feasible solution (if it exists) or prove that no solution exists. Others are heuristic approaches that strive to quickly find any feasible solution by searching subsets of the entire solution space and consequently cannot prove infeasibility. A comparison of all of the approaches on instances obtained from our industrial partners in The Netherlands and Denmark allows definitive conclusions regarding the strength and weaknesses of each approach, not to mention their applicability in practice, to be made. This paper therefore makes the following important contributions to the literature on train unit shunting.

1. Three new methodologies for solving the TUSP
2. Extensive computational experiments that compare the performance of these new methods with existing methods from the literature
3. A benchmarking of the methods on problem instances from multiple railway operators in different countries.

In addition to the main contributions, this paper also makes several minor ones. These include: the development of several efficient initial infeasibility checks that can be run prior to solving an instance of the TUSP; a problem decomposition approach that reduces an instance of the TUSP to a number of smaller, independent TUSP subproblems; an extension of a reference MIP, giving it the potential to dynamically add constraints; and an extension of the TUSP to also allow parking at platform tracks within the station at the end of the planning horizon. Computational tests suggest that there is no single method that dominates all others.

First, in Section 6.2 we give an overview of related literature and highlight the main differences to our contributions. In Section 6.3 we present the problem description. A number of polynomial time feasibility checks are discussed in Section 6.4 before introducing the solution methods in Section 6.5. The problem instances are presented in Section 6.8, followed by the benchmark of the solution methods. Finally, we conclude and give some remarks on further research in Section 6.9.

6.2 – LITERATURE OVERVIEW

To the best of our knowledge, the TUSP was first introduced in passenger railways by Freling et al. Freling et al. (2005). Other authors have considered different variants of the same problem, including additional constraints and decisions such as maintenance operations or station routing. In some cases, the matching of train units between arrivals and departures is given as input and not part of the problem. Otherwise, part of the problem is also to specify which compatible (arriving) train unit is matched to every departure. The remaining part of the TUSP is to find a valid parking plan for the in and out movements specified by the train matching. With the exception of Kroon et al. Kroon et al. (2008), all studies do not integrate the matching and parking problem, but solve them separately. A cost structure is often used to rank different matching and parking assignments. We, however, focus on the core matching and parking problem and do not differentiate between distinct solutions.

Freling et al. Freling et al. (2005) consider the problem of parking train units overnight at a depot in such a way that each train unit can be retrieved, without moving others, when needed during the operations of the following day. Any feasible parking must ensure that train units of different types do not block each other when departing from the yard the next day. The problem is decomposed into two smaller subproblems: a matching problem and a parking (or track allocation) problem. The first entails matching arrivals and departures at the depot under consideration. A solution to this problem hence also stipulates the train services each physical train unit will perform. This problem is formulated as a MIP and solved using a commercial solver. The parking problem, on the other hand, determines how to park the assigned matchings on each of the tracks at the yard such that the track capacity is never exceeded and such that the movements associated with the matchings are conflict-free. The authors model this as a set partitioning problem with side constraints and solve it using a heuristic column generation procedure. A corresponding MIP approach was not considered nor compared. The proposed approach was tested on one instance from Netherlands Railways and results were found within 20 to 60 minutes of computation time. In contrast to our work, the authors adopt a cost-structure to both problems in order to rank the solutions. We compare our methods with a variant of this approach in Section 6.8.

The work of Lentink et al. Lentink et al. (2006) extends the work of Freling et al. Freling et al. (2005), where a four-step approach is proposed to solve the matching and parking problem. The problems are still solved independently; however, the parking problem is extended to include more practical aspects, e.g. cleaning and maintenance of units, as well as the routing costs incurred from the depot to the station platforms. Their model is tested on small and big problems instances. The small problem instances are solved quickly, but the larger instances require at least 700 seconds of computation time.

A dynamic programming based heuristic approach for the TUSP is proposed by Haijema et al. Haijema et al. (2006). The matching and parking problems are solved sequentially and in isolation. To reduce the problem size the authors propose a rolling horizon technique to solve the problem. A realistic test case from the railway station Zwolle in the Netherlands is used to analyse the performance of the algorithm. A 24 hour period is considered in which 45 train units arrive and 55 train units depart. The depot has 19 tracks and a total capacity of 4000 meters. Solutions to the problems are found quickly and the results are promising; however, only a single instance is considered.

Solving the TUSP without some form of matching/parking separation has been proposed by Kroon et al. Kroon et al. (2008). The authors essentially extend the work of Freling et al. Freling et al. (2005) and propose a large MIP formulation that simultaneously solves the matching and parking problem. The authors try to keep the depot tracks as homogeneous (with respect to the train unit type) as possible. In addition, they also consider conflict-cliques in order to reduce the large number of conflict constraints. In contrast, in our work we accommodate this problem by adding conflict constraints on the fly in the Branch-and-Bound (B&B) framework. Practical restrictions that include how to handle depot tracks that can be approached from both sides are described. Two stations from the Dutch Railway network form the computational study, where instances with up to 125 train units of 12 different types are considered.

Jacobsen and Pisinger Jacobsen and Pisinger (2011) present three different heuristics to solve a variant of the TUSP that includes maintenance scheduling. The model is tested on small instances and the runtimes are low. Internal rearrangements are permitted if one train unit is blocking another train unit. The model is not tested on problem instances from practice.

In contrast to Freling et al. Freling et al. (2005), Lentink et al. Lentink et al. (2006), and Haijema et al. Haijema et al. (2006), we propose several new methods that integrate the matching and parking problems. Our methods are compared with the integrated and sequential methods from the literature. Furthermore, unlike any other research in this area, we test all methods on several classes of large and realistic instances from different train operating companies. In addition, we present and discuss the possibility of parking train units at platforms at the end of the planning horizon. This can also be seen in Table 6.1, where an overview of existing literature is given.

Paper	Integrated	Method	Benchmark	Platform
Freling et al. (2005)	No	MIP, CG	No	No
Lentink et al. (2006)	No	MIP, CG	No	No
Haijema et al. (2006)	No	DP	No	No
Kroon et al. (2008)	Yes	MIP	No	No
Jacobsen and Pisinger (2011)	Yes	H	No	No
This paper	Yes	CP, MIP, CH, H, (CG)	Yes	Yes

Table 6.1: Overview of the contribution of papers on the TUSP. The first column denotes the paper. The second column indicates whether the matching and parking problems are integrated. The third column states which solution methods are used, where CG stands for column generation, DP for dynamic programming, CP for CP, and H for heuristics. The fourth column states whether or not the solution methods are benchmarked on multiple practical settings from different railway operations. The final column specifies whether or not overnight platform parking is taken into account.

6.3 – PROBLEM DESCRIPTION

This section begins with a formal description of the TUSP, and this is followed by an example problem instance. We then introduce the platform extension used in this paper. General notation is introduced throughout the section, and investigation into the complexity of the problem is also provided.

6.3.1 – Problem description

The aim of the TUSP is to create a feasible shunting plan or prove that no such plan exists. A feasible shunting plan includes a feasible matching of all arrivals and departures at the depot as well as a feasible assignment of the obtained matchings to tracks. A valid matching pairs all initially parked train units at the depot, as well as all train units that arrive during the day, with compatible departures. Any unmatched train unit must remain parked at the depot. In other words, all unmatched train units reside in inventory until the end of the planning horizon. Furthermore, in a feasible matching all departures must be covered by a train unit, otherwise the TUSP is infeasible. A feasible solution to the matching problem must be assigned to depot tracks in a conflict-free manner. Without loss of generality, we can assume there is a one-to-one correspondence between arrivals and departures. Initially parked trains can be modelled as early arrivals, and train units that remain at the depot at the end of the planning horizon can be modelled as late departures.

All train units initially parked at the depot and all train units that arrive during the day are termed *arrival* events. A specific train unit type and known arrival time are associated with each arrival. The arrival time of initially parked train units is assumed to be the start of the planning horizon, denoted t_0. Likewise, departing train units are termed *departure* events. A departure has a known departure time and a required train unit type. For the sake of simplicity, a departure is defined for all train units remaining in the depot at the end of the planning horizon, which is denoted t_∞. We define a matching to be a combination of a compatible (with respect to train unit type) arrival event and departure event. Consequently, a feasible set of matchings covers all arrival events and all departure events exactly once. In other words, initially parked and arriving train units are matched to either a compatible departure or assigned to stay on some track in the depot at time t_∞.

A solution to the parking problem must satisfy two types of constraints. First, the capacity on each individual depot track may not be violated at any time. In other words, the total length of all train units residing on a track at any time may not exceed the length of the track itself. It suffices to ensure that this holds whenever a train unit arrives at the depot. Second, all tracks must be processed in a LIFO order, i.e the last parked train unit on a track must be the first train unit to leave. A train unit cannot leave a track (for a departure) if a different train unit has arrived in the meantime and is staying on the same track. Otherwise, a conflict would occur. We assume that an assigned track is occupied from (and including) the arrival time until (and including) the departure time of the corresponding matching. This is a conservative approach as an arrival will occupy its assigned depot track some time after arriving, and a departure will release the track allocation some time before departing from the associated station. By using this conservative approach, we are guaranteed to find a solution that is feasible in practice.

We make several assumptions in this paper. First, the shunting movements at different stations in the railway network are assumed to be completely independent of each other. Hence, the respective TUSPs can be solved separately for every station. Second, no maintenance operations are considered in our problem. Thus, train units of the same type are assumed to be completely interchangeable. Third, internal shunting of train units is not considered; once a train unit is parked it can only be moved once retrieved for departure. In practice, internal shunting can be performed, albeit at the cost of using shunting personnel resources. A plan without additional movements is preferred, if possible. Finally, a certain buffer period between consecutive arrivals and departures may be desired for the same train unit. We assume that a unit must be parked on a depot track for at least β minutes. The value of β is parametric. A high value of β will result in a shunting plan that is more robust to delays; however, it also reduces the combinatorial solution space. In the computational experiments we set β equal to the lowest value, $\beta = 1$.

The majority of the considered problem instances have an initial inventory of train units at the start of the planning horizon. Each depot track may therefore contain a number of train units in a specific order. The proposed approaches all adhere to this initial ordering of the train units. It is also possible for the presented approaches to determine the initial parking order of the train units on the track. Not having an initial ordering is a less restrictive problem, potentially giving a greater number of feasible solutions. We do not pursue this variant in this paper, but note that it may be relevant at a strategic or tactical planning level.

In contrast to the literature, no cost structure is defined in our work on the TUSP. The TUSP is considered a feasibility problem as this is the most important question to answer. Any of the proposed solution approaches could be used as part of a larger framework to determine whether or not a feasible solution exists before finding the most preferred one. This is highly applicable in an operational setting where time is limited and it is crucial to quickly detect feasibility. We note, however, that almost all of the presented approaches can, without much difficulty, be extended to include an objective function.

6.3.2 – Example

An example of a problem instance is given in Table 6.2. The example highlights the importance of integrating the matching and parking problems. The table lists a set of arrival and departure events. The time each event occurs and its associated train unit type are also specified. Furthermore, for this example, the lengths of the train unit types for a, b and c are 200, 100 and 150 meters, respectively. Two depot tracks are assumed to be available, one with length 550 meters and one with length 200 meters.

Event	Type	Time
Arrival	a_1	12:00
Arrival	a_2	12:30
Arrival	b_1	13:00
Arrival	c	13:30
Arrival	b_2	14:00
Departure	b	15:00
Departure	c	15:30
Departure	a	16:00

Table 6.2: Example list of events in a problem instance. Note, that a_1 and a_2 (and b_1 and b_2) denote the same train unit type.

Each arrival must be matched to a compatible departure, or remain in the depot. For the latter option the arrival is paired to an artificial event, which is denoted inv. For the sake of argument, if parking constraints are ignored, we can identify two feasible solutions to the matching problem. These are given in Table 6.3.

Matching 1	Matching 2
(b_1, b)	(b_2, b)
(c,c)	(c,c)
(a_1, inv)	(a_1, inv)
(a_2, a)	(a_2, a)
(b_2, inv)	(b_1, inv)

Table 6.3: Two feasible solutions to the matching problem for the TUSP instance in Table 6.2.

From a parking perspective, only one of the two matchings is feasible at 15:00. The arriving units are parked on the tracks in the order specified in Figure 6.1. Matching 1 is clearly infeasible as train unit b_2 blocks the departure of train unit b_1. Matching 2, on the other hand, is conflict-free: no train unit is blocking a departure of a different train unit.

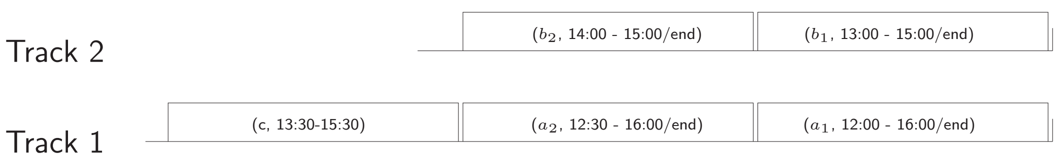


Figure 6.1: An example of train units in Table 6.2 assigned to two depot tracks.

Platform Parking

Train units can be, and are in certain situations, parked on platform tracks in practice. During the day passengers board and alight from trains at platforms. Consequently, train units should not be parked there during the day. However, during the night there is no, or limited, traffic. If train units parked on platform tracks overnight can service the first train service from the platforms on the following day, then no additional shunting is required.

According to some railway operators, e.g. Netherlands Railways (NS) and Danish State Railways (DSB), train units are, under certain circumstances, parked on platform tracks. The rolling stock activities for the following day are known, and it can be practical to park train units on a platform track overnight if the parked composition is to depart early the next day. Depending on the track layout and the number of platforms at a station, a number of platform tracks may be eligible to be used in this way. However, still, a number of platform tracks must be reserved in order to allow night trains or maintenance crews to operate.

At any station we assume that a certain number of platform tracks, N, can be used for overnight parking. In order to ensure a smooth operation, the first N departing train services (of the following day) dictate which train units can be parked on the N available platform tracks. A departing train service may consist of multiple train units, allowing more than a single train unit to be parked on a platform track. For instance, if $N = 2$, and the first two departing trains have the compositions aa and bc, then two train units of type a can be assigned to the first platform track (perhaps from different arrival services) and a single train unit of each type b and c to the second platform track. In this variation of the problem, the extension can be handled by adding the N platform tracks as normal shunting tracks with additional restrictions. More specifically, in addition to the existing track constraints, a matched arrival and departure pair can only be assigned to a platform track if the arrival takes place in an appropriate time window (close to the end of the daily operation) and if the departure corresponds to one of the first train services of the following day.

6.3.3 – General Notation

When solving an instance of the TUSP for a specific depot, a set of arrival and departure events is assumed to be given. This set of events is denoted by E. The sets E^{arr} and E^{dep} respectively contain all arrivals and departures, and together define a partition of all events; that is, $E^{arr} \cup E^{dep} = E$ and $E^{arr} \cap E^{dep} = \emptyset$. An arrival corresponds to a train unit that is uncoupled at the station and must be parked in the depot, while a departure corresponds to a train unit that is to be coupled at the station and must be retrieved from the depot. Furthermore, we are given a set, M, of train unit types and a set, S, of tracks on which units can be parked. Two subsets of S, S_d and S_p, with $S_d \cup S_p = S$ and $S_d \cap S_p = \emptyset$ contain, respectively, the set of depot tracks and the set of platform tracks. All definitions are summarized in Table 6.4. The time at which event $e \in E$ takes place is denoted by t_e, and $m_e \in M$ denotes the train unit type of the corresponding event. We let c_s denote the capacity of track $s \in S$. This capacity is equal to the maximum length that can be stored simultaneously on track $s \in S$. The length of a train unit type $m \in M$ is denoted by l_m. Table 6.5 summarizes these parameters.

Set	Definition
E	Set of all events
$E^{arr} \subset E$	Set of arrival events
$E^{dep} \subset E$	Set of departing events
M	Set of train unit types
S	Set of tracks
$S_d \subset S$	Set of tracks in the depot for parking
$S_p \subset S$	Set of platform tracks

Table 6.4: List of defined sets

Notation	Description
t_e	Time event $e \in E$ takes place
m_e	Train unit type corresponding to event $e \in E$
l_m	Length of train unit type $m \in M$
c_s	Capacity of track $s \in S$

Table 6.5: List of parameters

Complexity

The complexity of the TUSP has been addressed by multiple authors in the literature. Several variations of the problem exist, each of which is known to be $\mathcal{NP}$-hard. The shunting problem considered by Freling et al. Freling et al. (2005) is essentially a specialization of the considered TUSP. They prove the variant to be $\mathcal{NP}$-hard by a reduction from the Tram Dispatching Problem studied by Winter et al. Winter (1999), which in turn is $\mathcal{NP}$-hard.

We present a simple and informal proof, that shows that the considered TUSP is $\mathcal{NP}$-hard by a reduction from the Graph Coloring Problem (GCP). In the GCP a color must be assigned to each vertex of a graph such that no adjacent vertices have the same color. Two vertices are adjacent if an edge connects them. The problem is then to decide the fewest number of colors needed. The corresponding decision problem is to decide whether a graph can be colored using k colors. The GCP is known to be $\mathcal{NP}$-hard Garey et al. (1974).

First, the TUSP is in $\mathcal{NP}$ since a feasible solution can be verified in polynomial time. The matching constraints are verified by counting the number of assignments, the capacity constraints can be verified by looping through the events (ordered by time), and a pair-wise comparison of all matching assignments to tracks can verify that the resulting parking is conflict-free. Second, we argue that the TUSP is a generalization of the GCP. Given an instance of the GCP, the number of colors corresponds to the number of available tracks. The length of the tracks is set sufficiently high, such that it is never binding. The constructed TUSP instance is generated such that only one valid matching exists by assigning a unique train unit type to all train units. A vertex (in the GCP) corresponds to a track assignment of a matching, and the selected track represents the selected color. The time of the constructed events reflects an ordering that corresponds to the edges in the GCP, i.e., if two vertices are connected then the corresponding arrival time is in between the arrival and departure time of the other, thus conflicting. If the constructed TUSP instance contains a feasible solution, then a feasible assignment of k colors is given by the track assignments of the matchings of the TUSP instance. Thus, the TUSP can be reduced to an instance of the GCP, and we can conclude that the TUSP is $\mathcal{NP}$-hard.

6.4 – INFEASIBILITY CHECKS

A select number of efficient checks can be performed independently of any solution method to assess the feasibility of an instance of the TUSP. The problem instances considered in the benchmark testing of Section 6.8 have all passed the checks discussed in this section. There is no reason to consider an instance, which violates any of the following checks as it is inherently infeasible.

Aggregated Track Capacity

At any given time the sum of all depot track lengths must be at least the sum of the lengths of all train units that need to be parked. This aggregated constraint must hold since no feasible solution can exist if it is violated. This property is easily checked in polynomial time. Rolling stock schedules implicitly satisfy this constraint if depot capacity is modeled in the rolling stock problem.

Individual Track Capacity

The depot must have at least one feasible initial parking with respect to the capacity constraints of all tracks. At the start of the planning horizon a feasible parking must exist, otherwise no solution can exist to the TUSP. With a given set of initial train units, a feasible solution can be found by solving a Multiple Knapsack Problem (MKP). Every train unit corresponds to an item, where the capacity consumption is equal to the physical length of the train unit. Each depot track corresponds to a knapsack, where the capacity is equal to the track length. Good algorithms for solving the MKP exist (see e.g. Pisinger Pisinger (1995)).

In this paper, we assume that the initial parking is feasible with respect to the mentioned knapsack constraints. However, the same constraints must be satisfied during the whole planning horizon, and not only for the initial parking. The same check is applied every time a train unit arrives to the depot. Note that the individual track capacity can be violated even if the aggregated track capacity is satisfied.

Feasible Matching

In the common case, multiple feasible matchings exist for the same problem instance, especially when ignoring depot track capacities. The reason for integrating matching and parking in the TUSP is the fact that all feasible matchings do not necessarily have a feasible parking assignment. However, if no feasible matching exists, then the TUSP is infeasible as well. Detecting whether a feasible matching exists is equivalent to solving the Assignment Problem (AP) (Munkres Munkres (1957)), which is solvable in polynomial time as the resulting matrix of the Linear Program (LP) is totally unimodular. As the number of arrival and departure events are the same, the problem is the linear AP; this can be solved using specialized polynomial time algorithms, such as the Hungarian algorithm (Kuhn Kuhn (1955)). Again, as our problem instances are the result of a feasible rolling stock schedules at least one matching must exist.

6.5 – SOLUTION METHODS

In this section we describe three new solution methods for solving the TUSP. Section 6.5.1 is dedicated to the Constraint Programming Method (CPM), while Section 6.5.2 introduces the column generation method. An overview of the Randomized Greedy Construction Heuristic (RGCH) approach is given in Section 6.5.3. Existing methods in the literature are described in Section 6.6.

6.5.1 – The Constraint Programming Method

As the TUSP is essentially a feasibility problem, a CP approach might be effective. Our proposed CP formulation is inspired by the rolling stock composition model of Fioole et al Fioole et al. (2006). This formulation was originally used for Rolling Stock (re-)Scheduling; however, we can use the idea of compositions and composition changes for the TUSP. In this formulation we attempt to assign compositions to tracks whenever an event occurs. At such times one composition must be assigned to each track individually. Similar to the rolling stock problem, a composition consists of a number of train units in a specific order. For instance, in Figure 6.1, the composition caa is assigned to track one and the composition bb is assigned to track two. Note that the empty composition, containing no train units, is a valid composition as well. Composition changes can only occur whenever events occur. A composition change is the transition of one composition to another. For instance, if an arrival of a train unit takes place at a specific track, then a train unit is, in effect, coupled to the composition currently assigned to the track. Thus, a composition change consists of two compositions, the one prior to the event occurring and the one immediately after it has taken place.

We let C be the set of all possible compositions and Q be the set of all possible composition changes. The set $Q_{e,s}$ consists of all feasible composition changes that can take place just after event $e \in E$ on track $s \in S$. For instance, if event e stipulates that a train unit of type a is arriving, then only composition changes where a train unit of type a appears on top of the stack are included in $Q_{e,s}$. Additionally, composition changes where no train units are appended or removed are included as all unaffected tracks remain unchanged.

The CP also requires some parameters. First of all, i_s specifies the initial composition on track $s \in S$. Next, λ_e defines the predecessor event of event $e \in E$, i.e. the event that occurs just before e. Furthermore, for composition change $q \in Q$, we introduce notation $\texttt{In}[q]$ and $\texttt{Out}[q]$ to denote the index of the first and second composition in a composition change in the list of allowed composition changes, i.e. the original composition and its successor composition. Finally, $\alpha_m[q]$ and $\beta_m[q]$ specify whether a train unit of type m is appended or removed. Table 6.6 summarizes the sets and parameters used in the CP approach.

We define two families of decision variables. First, the integer variable $X_{e,s}$ specifies which composition is assigned to track $s \in S$ just after event $e \in E$. The compositions are mapped to integer values, e.g., $X_{e,s} = 3$ stipulates that the ab composition is assigned to track s after event e. Recall, a composition $c \in C$ which is assigned to track $s \in S$ just after event has occurred $e \in E$ consists of all train units parked at that moment on track s, in order of arrival time. For instance, if the composition $abcd$ is assigned to track s after event e occurs, then it means that train unit d was parked there first, followed by train units c, b, and a.

Set or parameter	Definition
C	Set of possible compositions
Q	Set of possible composition changes
$Q_{e,s}$	Set of composition changes that are allowed after event $e \in E$ at track $s \in S$
i_s	The composition belonging to the start inventory at track $s \in S$
λ_e	The predecessor event of event $e \in E$
$\texttt{In}[q]$	The index of the first composition belonging to composition change $q \in Q$
$\texttt{Out}[q]$	The index of the second composition belonging to composition change $q \in Q$
$\alpha_m[q]$	Equals 1 if a train unit of type $m \in M$ is appended to the composition on the track during composition change $q \in Q$
$\beta_m[q]$	Equals 1 if a train unit of type $m \in M$ is removed from the composition on the track during composition change $q \in Q$

Table 6.6: List of all additional sets and parameters required by the CP model

Similarly, the integer decision variable $Y_{e,s}$ indicates the composition change that takes place on track $s \in S$ just after the occurrence of event $e \in E$. An example of this is the composition change from aa to a when one train unit of type a departs track s. Again, an integer value corresponds to a transition from one composition to another.

The construction of $Q_{e,s}$ models the allowed composition changes with respect to the depot track capacity and the LIFO restrictions. Note that platform parking can also be modeled in the construction of this set. First, composition changes that exceed the length of a track are not allowed. All composition changes that involve a transition to a composition that has a total length longer than the capacity of track s are removed from the set $Q_{e,s}$. Furthermore, restrictions with respect to the train unit type of events are considered. First, if event $e \in E$ is an arrival of a train unit of type a, then only composition changes where a train unit of type a is appended and composition changes where no train units are appended or removed are allowed. Second, the LIFO constraints further restrict the set of composition changes. If, for instance, the composition $abcd$ was assigned to track s just after event λ_e occurs, then there are only two allowed composition changes after a departure event $e \in E$ on track s: $abcd \rightarrow abcd$, or $abcd \rightarrow bcd$. Train units b, c, and d cannot depart as train unit a is blocking them. Finally, with respect to the platform tracks, it is not allowed to park train units at platform tracks during the day, the only allowed composition change after events during the day is $empty \rightarrow empty$. For the last events of the day, however, platform track parking can be considered. In such cases the platform track composition is restricted to being a subset of the composition of the train service which will first depart from the platform the following day. This can be considered by only allowing composition changes that involve transitions to compositions which are a subset of the departing train service composition.

The TUSP can therefore be modeled with the following mathematical program, which is used as a basis for the CPM:

$$X_{\lambda_e,s} = \texttt{In}[Y_{e,s}] \qquad \forall e \in E, s \in S \tag{6.1}$$

$$X_{e,s} = \texttt{Out}[Y_{e,s}] \qquad \forall e \in E, s \in S \tag{6.2}$$

$$\sum_{s \in S} \beta_m[Y_{e,s}] = 1 \qquad \forall e \in E^{dep}, m \in M : m_e = m \tag{6.3}$$

$$\sum_{s \in S} \alpha_m[Y_{e,s}] = 1 \qquad \forall e \in E^{arr}, m \in M : m_e = m \tag{6.4}$$

Constraints (6.1) state that the first composition of a chosen composition change on a track has to match the actual composition that is appointed before the composition change took place on the track. This actual composition is the composition that is assigned to the track just after the previous event, λ_e has occurred. A similar composition flow conservation constraint is used for the second composition of a chosen composition change. This composition must be equal to the actual composition that is appointed to the track after the composition change took place, which equals the composition that is assigned to the track just after the occurrence of event e. This is modeled by Constraints (6.2).

Every departure event has a corresponding train unit that has to depart from precisely one of the tracks in the depot. Consequently, exactly one composition change has to be selected where a train unit of type m_e has departed; on all other tracks no shunting movements can take place. This is modeled by Constraints (6.3). A similar requirement holds for an arrival. The corresponding train unit m_e has to be appended to precisely one track; this is handled by Constraints (6.4). Finally, the start inventory is enforced by fixing the values for $X_{e,s}$ of an auxiliary source event e that occurs before the first event.

6.5.1.1 – Solution procedure

The proposed model can be solved using a CP solver, effectively solving the TUSP by assigning compositions and composition changes to the events on the tracks. Similar to the rolling stock scheduling variant in Freling et al. (2005), the model does not scale well. Long depot tracks and a high number of train unit types result in a huge number of variables in practice. A large number of variables is needed for long depot tracks as the number of possible compositions increases drastically in such cases. Many different train unit types further increase the combinatorial solution space.

Preliminary results showed that small instances with two train unit types are practical to solve. However, larger instances with four train unit types quickly become impractical to solve, primarily due to the memory footprint. As a remedy, the CPM can be solved in a more practical way. We present a variant, the Constraint Programming Method Heuristic (CPMH), that restricts the compositions on tracks to contain at most $\epsilon \geq 1$ different unit train types. Note, that the contained train unit types can change over the course of a planning horizon. At any time, the number of different train unit types is at most ϵ. In effect, fewer tracks will be mixed, i.e., contain more than one train unit type, in the solution. Keeping tracks homogeneous is beneficial as it reduces the potential LIFO conflicts. The main benefit is the drastic reduction of the number of variables in the CP model. If the CP model finds a solution, then clearly it is feasible for the TUSP. However, if it fails to find a solution we cannot conclude that the instance is infeasible, unless $\epsilon = |M|$.

The minimal value of ϵ that produces a feasible solution is initially unknown, therefore we begin with $\epsilon = 1$. The strategy is to solve the model using increasing values of ϵ. A time limit of γ minutes is set in every iteration. Otherwise, too much time is potentially spent searching an infeasible solution space. In this paper we divide the available time, T, uniformly by the number of train unit types in the problem instance, $\gamma = {}^{T}/_{|M|}$. In every step when no solution has been found, we increase ϵ with 1 and try again, until $\epsilon = |M|$.

6.5.2 – Column Generation Method

Column generation is a well-known decomposition method for solving large-scale LPs. The most attractive feature of the approach is that it only generates variables that have the potential to improve the objective function while implicitly considering all non-basic variables included in the formulation. In what follows, we present a column generation procedure for the TUSP. To apply column, we decompose the problem by track, and attempt to assign each track a set of possible matchings, termed a *matching pattern*. A matching pattern is a subset of matchings that can be feasibly assigned to a given track over the planning horizon. In particular, it is a set of matchings that satisfies the LIFO requirements as well as the available track length restriction. A large number of possible matching patterns exist. Thus the approach relies on the dynamic generation of variables that represent promising matching patterns. In this paper, the solution method is referred to as the Column Generation Method (CGM).

The proposed formulation is based on the methodology presented by Freling et al. Freling et al. (2005), with the exception that in our work the matching and parking problems are not solved separately. We present a model and solution framework that simultaneously solves both problems. For the description of the model, we introduce the set $\mathcal{P}_s$, which denotes the set of all feasible matching patterns for track $s \in S$. Note, that platform track parking and LIFO constraints are already satisfied in this set. A binary decision variable $X_{p,s}$ is defined for each $s \in S$ and $p \in \mathcal{P}_s$ and governs the inclusion of the corresponding matching pattern in the final solution. A value of one indicates that the matching pattern is chosen, while a value zero indicates otherwise. As the majority of the constraints are embedded in the column construction phase. The problem can be formulated as a large generalized set partitioning problem. In the model, at most one matching pattern is assigned to each track. Further, each arrival and departure must appear in exactly one matching pattern, otherwise it is left uncovered. Binary variables Y_e, where $e \in E^{arr}$, and Z_e, where $e \in E^{dep}$, are used to indicate whether an arrival, respectively departure, is matched or not. The objective of the model is to match as many arrivals and departures as possible. Thus, the objective function simply minimizes the number of unassigned arrivals. The binary parameter $\alpha_{e,p}$ is used to indicate whether or not event $e \in E$ is contained in matching pattern $p \in \mathcal{P}_s$. The full binary integer program is given as follows.

$$\text{Minimize:} \quad \sum_{e \in E^{arr}} Y_e \tag{6.5}$$

$$\text{subject to:} \quad \sum M y_e + \sum M z_e$$

$$\sum_{p \in \mathcal{P}_s} X_{p,s} \leq 1 \qquad \forall s \in S, \ (\mu) \tag{6.6}$$

$$\sum_{s \in S} \sum_{p \in \mathcal{P}_s} \alpha_{e,p} X_{p,s} + Y_e = 1 \qquad \forall e \in E^{arr}, \ (\pi) \tag{6.7}$$

$$\sum_{s \in S} \sum_{a \in \mathcal{P}_s} \alpha_{e,p} X_{p,s} + Z_e = 1 \qquad \forall e \in E^{dep}, \ (\gamma) \tag{6.8}$$

$$X_{p,s} \in \{0,1\} \qquad \forall s \in S, p \in \mathcal{P}_s, (\gamma) \tag{6.9}$$

$$Y_e \in \{0,1\} \qquad \forall e \in E^{arr}, (\gamma) \tag{6.10}$$

$$Z_e \in \{0,1\} \qquad \forall e \in E^{dep}, (\gamma) \tag{6.11}$$

The objective function (6.5) minimizes the number of uncovered arrivals. Constraints (6.6) ensure each depot track is assigned at most one matching pattern. Constraint sets (6.7) and (6.8) enforce the requirement that each arrival and departure appears in one of the selected matching patterns, or is left uncovered. Finally, variable domains are specified by constraints (6.9)-(6.11). We refer to Model (6.5)-(6.11) as the *master* problem.

The Master Problem

Given the exponential number of matching patterns in any real-life example it is impractical to enumerate all corresponding columns and solve this formulation. In our solution method, a subset (restricted set) of the possible matching patterns is included. We relax the integrality restrictions and associated bounds given by (6.9)-(6.11). A *relaxed restricted master problem* (RRMP) is obtained. Using the optimal dual solution vector (μ^*, π^*, γ^*) to this relaxed problem, a *pricing* problem, or subproblem, is solved to determine if any favourable matching patterns exist. Promising variables are inserted iteratively into the restricted master problem until none exists - implying that the LP solution is proven optimal. By iterating between the RRMP and several pricing problems (typically one for each track), one can limit the search for the optimal solution to model (6.5)-(6.11) to include only those matching patterns that have the potential to improve the objective value. For a general introduction to column generation the reader is referred to Lübbecke and Desrosiers (2004).

The Pricing Problem

The pricing problem requires one to find a favourable set of matchings that can feasibly be parked on a given track. In other words, given an optimal solution to the RRMP, one must solve up to $|S_d| + |S_p|$ pricing problems at any column generation iteration to determine if any improving matching pattern exists. To find such patterns we present an approach that finds shortest paths in a directed graph.

The directed, acyclic graph contains one node for every possible matching, one node for every arrival (corresponding to not parking the arrival), and a source and sink node. It is layered by matchings for each arrival (including the node corresponding to not parking the arrival) and these layers are ordered in increasing arrival time. Arcs connect matchings in one layer with those of the subsequent layer - provided the two matchings can feasibly use the same track. The source node is connected to each matching in the first layer, while each matching in the last layer is connected to the sink. The cost on an any arc entering a node is equal to the reduced cost contributions of the associated matching. For example, if events $e \in E^{arr}$ and $e' \in E^{dep}$ are matched, the cost on any arc entering the node corresponding to this matching will have a cost of $-(\pi_e + \gamma_{e'})$. An example of such a network is given in Figure 6.2

As we must observe the available track length and satisfy the LIFO requirements when generating matching patterns, a resource constrained shortest path problem must be solved. Consequently, a standard label setting algorithm is used to identify paths in this network. The algorithm is similar to that of Freling et al. Freling et al. (2005); however, as we must also simultaneously find the matchings, the proposed network is much bigger. Additionally, we

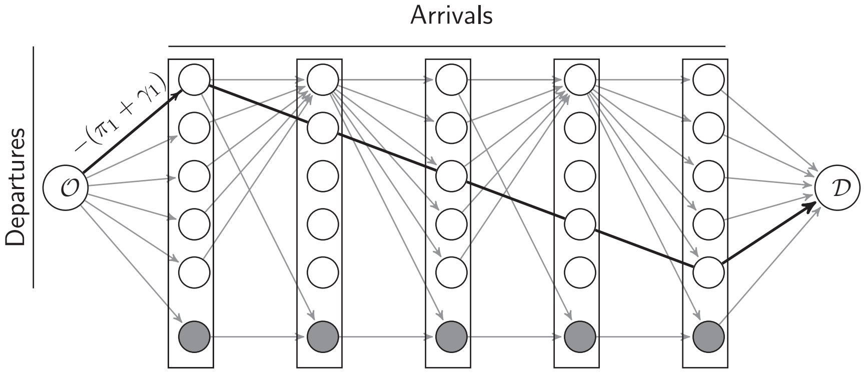


Figure 6.2: An example subproblem network with five arrivals and five departures. Every node corresponds to either the matching of an arrival and a departure or an unmatched arrival (gray nodes). All paths originating at the source ($\mathcal{O}$) and terminating at the sink ($\mathcal{D}$) represent a matching pattern, which is feasible if the resource constraints are respected. An example of such a path is given in black. Note that not all arcs and costs are shown.

must also keep track of previously matched departures. This is not nice from a dominance perspective in a label setting algorithm as it weakens the possibility of dominating partial paths away in the shortest path solve. This has a detrimental affect on the runtime of the shortest path algorithm. For large problems it is not even practical. We implement the dominance strategy described in the label setting algorithm of Freling et al. (2005) and include an extra check in the extension of a partial path to make sure that we do not visit a matching for a previously matched departure. This is a heuristic variable generation procedure, but keeps the runtimes manageable.

To ensure exactness of the column generation, the heuristic column generation approach outlined can be complemented with a MIP solve. For instance, one can resort to a MIP when the column generation fails to identify a negative reduced column. This MIP, however, would be similar in structure to that described in Section 6.6.1, with the exception that only the "best" set of matchings need to be decided for the shunting track in question. For large problems, this is expected to be slow.

6.5.3 – The Randomized Greedy Construction Heuristic

Modeling ordering constraints efficiently in integer LPs such as LIFO constraints is cumbersome. The proposed solution methods and all existing ones overcome this issue by either adding all pairwise conflicts, enumerating all possible transition states, or by generating feasible parking patterns. As such, all methods have scaling issues, whether it be in the number of constraints or the number of variables. In contrast, modelling one or multiple stacks programmatically would not have such issues.

We propose a heuristic that greedily assigns arrivals and departures to and from tracks. The important aspects of this heuristic are the efficiency of the construction and the randomization of the greedy choice. Together these characteristics allow the heuristic method to try many different track assignments and extractions within a short time. The method terminates with the first feasible solution.

Algorithm 3: Randomize Greedy Construction Heuristic

```
Input:  Track set S
Input:  Event set E, ordered by time
Output:  Matching set M and parking plan
M ← ∅
S ← InitializeEmptyTrackStacks()
for e ∈ E do
    if Type(e) =Arrival then
        s ← FindRandomCompatibleTrack(S)
        if s = ∅ then
            return ∅
        else
            S[s] ←Push(S[s], e)
        end if
    else
        s ← FindRandomCompatibleUnitType(S)
        if s = ∅ then
            return ∅
        else
            (S[s], e′) ←Pop(S[s])
            M ← M ∪ {(e, e′, s)}
        end if
    end if
end for
```

An overview of the heuristic is shown in Algorithm 3. The input to the heuristic is the set of events to process and the set of available tracks. The main loop processes events chronologically. When an arrival event occurs, a random compatible track is sought, i.e., any track that has sufficient remaining length to hold the arriving train unit. Many candidates may exist, thus the following selection criteria are used:

1. A track where the existing outmost train unit has the same train unit type
2. A track which is empty
3. Any track with sufficient capacity

The goal is to group train units of the same type and avoid stacking different train unit types on the same tracks. Recall that train units of the same type do not block each other as they are interchangeable. In order to avoid a standstill in certain instances at depots with scarce capacity, the first and second criteria are skipped with a low probability. The situation occurs when train unit types must be mixed on tracks in order to utilize the capacity fully.

For a departure event, i.e. a train unit must leave the depot, tracks are processed in a random order. The track with the correct train unit type (on top of the stack) is selected. Here it might be worthwhile considering a selection criteria approach based on the effect of removing this train unit. However, preliminary results show that this simple extraction rule is sufficiently effective.

The algorithm output is a list, where every element defines an arrival event, its matching departure event, and a specific track to which the corresponding train unit is appointed. On arrival the train unit is parked on the track, and the specified departure extracts the train unit from the same track. The heuristic is able to evaluate one construction path very quickly, thus it is embedded in an iterative loop, where the heuristic is applied with different seeds to initialize the random number generator. Every iteration thus essentially restarts the whole process. The loop continues until either a feasible solution is found or a predefined time limit is reached.

6.6 – LITERATURE METHODS

In the previous section three novel methods were introduced to solve the TUSP. In Section 6.8, we compare these methods with existing methods from literature. For completeness, we briefly describe two methods based on existing approaches. Section 6.6.1 is dedicated to the Reference MIP Method (RMM), while Section 6.6.2 focuses on theTwo-Stage Method (TSM).

6.6.1 – The Reference MIP Method

We begin by explaining the RMM. This model is based on the paper Kroon et al Kroon et al. (2008). Here, arrivals and departures are linked using matchings. The matching of an arrival and a departure event is allowed if and only if sufficient time separates the events, and the train unit types are compatible. The set of all possible matchings is denoted by $\mathcal{A}$:

$$\mathcal{A} = \{ \quad (e,f) \quad | \quad t_e + \beta \leq t_f, \quad m_e = m_f, \quad e \in E^{arr}, \quad f \in E^{dep} \quad \}$$

The set of matchings where event $e_1 \in E^{arr}$ is the arrival event is denoted by $\mathcal{A}^{arr}_{e_1}$ and the set of matchings where event $e_2 \in E^{dep}$ is the departure event is denoted by $\mathcal{A}^{dep}_{e_2}$. These sets are summarized in Table 6.7.

Set	Definition
$\mathcal{A}$	Set of all matchings
$\mathcal{A}^{arr}_{e}$	Set of matchings where event $e \in E^{arr}$ is the arrival event
$\mathcal{A}^{dep}_{e}$	Set of matchings where event $e \in E^{dep}$ is the departure event

Table 6.7: List of MIP specific sets

We change the definition of X here, such that $X_{a,s}$ takes a value of 1 if and only if matching $a \in \mathcal{A}$ is selected and parked on track $s \in S_d$.

A number of constraints need to be satisfied in order to achieve a feasible solution. First, each arrival must be assigned to exactly one departure, c.f. Constraints (6.12). Recall that a departure represents an actual departure or a stay in the depot. Similarly, each departure must be assigned to exactly one arrival, c.f. Constraints (6.13).

$$\sum_{s \in S_d} \sum_{a \in \mathcal{A}^{arr}_{e}} X_{a,s} = 1 \qquad \forall e \in E^{arr} \tag{6.12}$$

$$\sum_{s \in S_d} \sum_{a \in \mathcal{A}^{dep}_{f}} X_{a,s} = 1 \qquad \forall f \in E^{dep} \tag{6.13}$$

Constraints (6.12)-(6.13) ensure a feasible matching. For the parking, the capacity of a depot track can not be exceeded at any point in time. It is sufficient to ensure that the track capacity is not exceeded at every arrival, c.f. (6.14).

$$\begin{aligned} &\sum_{\{e' \in E^{arr} | t_{e'} \leq t_e\}} \sum_{a \in \mathcal{A}^{arr}_{e'}} X_{a,s} \cdot l_{m_{e'}} \\ &- \sum_{\{e' \in E^{dep} | t_{e'} \leq t_e\}} \sum_{a \in \mathcal{A}^{dep}_{e'}} X_{a,s} \cdot l_{m_{e'}} \leq c_s \qquad \forall e \in E^{arr}, s \in S_d \end{aligned} \tag{6.14}$$

For each arrival, Constraints (6.14) sum the contribution of past events and ensure that the used track length is less than the available track length; arrivals consume capacity, while departures release capacity.

The depot tracks are subject to LIFO restrictions. Only the out-most (top of stack) can be retrieved at any point in time. We model these restrictions by adding one constraint per pair-wise conflict to forbid such assignments, c.f. Constraints (6.15). It states that any two pairs of matchings cannot be assigned simultaneously if they block each others' movements.

$$X_{a,s} + X_{a',s} \leq 1 \qquad \forall s \in S_d, (a, a') \in \mathcal{C} \tag{6.15}$$

where

$$\begin{aligned} \mathcal{C} = \{ \quad (a, a') \quad | \quad & a = (e, f) \in \mathcal{A}, \\ & a' = (e', f') \in \mathcal{A}, \\ & t_{e'} < t_e \ \wedge \ t_{f'} < t_f \ \wedge \ t_{f'} > t_e \quad \} \end{aligned}$$

Note, that the conflict set is not track-dependent. All pair-wise conflicts are therefore repeated for every track, effectively generating very many constraints. We note that the number of constraints in the model can be reduced, possibly drastically, by replacing the pair-wise conflicts with conflict cliques, see Kroon et al Kroon et al. (2008). In general, the problem of finding maximum cliques is $\mathcal{NP}$-hard (Karp Karp (1972)). However, a number of cliques can be found heuristically in order to make the problem more tractable.

The initial inventory is not modeled directly in the above formulation. Recall that initially parked train units are modeled using arrivals and that parking (at the end of the planning horizon) is modeled using departures. Any initial train unit to track assignment can be modeled by fixing the variables corresponding to the first set of events. In the computational experiments of Section 6.8, an initial parking order is imposed. We note that in the implementation of the model, all fixed variables are removed to obtain a more compact model.

Extension

Due to the large number of conflict constraints (6.15) present in the model, we introduce in an extension of the reference MIP, termed the Delayed Constraint MIP Method (DCMM), in which Constraints (6.15) are generated on-the-fly. The DCMM is solved as a MIP model, where violated conflict constraints are added as they become violated by the optimal LP solutions. Initially, no conflict constraints are added. The success of this approach depends on the fact that most conflicts will never be violated in the B&B approach of a MIP solver.

6.6.2 – The Two-Stage Method

In the Two-Stage Method (TSM) the matching and parking problems are solved sequentially. Given a feasible matching to an instance of the TUSP, the remaining problem simply entails assigning the matchings to tracks. A problem instance may contain multiple feasible matchings for which a feasible parking exists. Solving these two problems in isolation is expected to be easier and motivates the TSM, where in the first stage a feasible matching of arriving and departing train units is generated, while in the second stage the method tries to assign the found matchings to tracks. The TSM is based on the method of Freling et al. Freling et al. (2005).

The matching and the parking problems can be solved using different methods. For the matching problem we use a MIP approach for two reasons. First, it is easy to formulate and implement a MIP for the matching problem. Second, as mentioned in Section 6.4, the resulting LP is totally unimodular. We also adopt a MIP approach for solving the parking problem. Freling et al. Freling et al. (2005) propose a column generation approach for solving this problem; however, the fast runtimes of our approach gave us no reason to pursue a more complicated framework.

For the matching problem formulation we reformulate the binary decision variable X_a; this variable indicates whether a given matching $a \in \mathcal{A}$ is selected or not. In a feasible matching, each arrival and departure should appear in exactly one matching. The resulting constraints of the MIP are given below.

$$\sum_{a \in \mathcal{A}_e^{arr}} X_a = 1 \qquad \forall e \in E^{arr}, \tag{6.16}$$

$$\sum_{a \in \mathcal{A}_e^{dep}} X_a = 1 \qquad \forall e \in E^{dep}, \tag{6.17}$$

$$X_a \in \{0,1\} \qquad \forall a \in \mathcal{A}. \tag{6.18}$$

There is no objective used in the MIP, since we are interested in feasibility only. Constraints (6.16) and (6.17) ensure, respectively, that each arrival and departure event appears in exactly one matching. The variable domain is given by (6.18). We note that several solutions (i.e. matchings) to the model may exist. It is therefore possible to guide the solution in a more advanced approach, but we do not pursue this addition in this paper.

If a solution to the matching problem problem exists, we proceed to the second stage and attempt to assign them to depot tracks. For this, we use the MIP described by Haahr et al. Haahr et al. (2014), which is identical in structure to the reference MIP approach described earlier.

The TSM is similar to what is described by Freling et al. Freling et al. (2005). However, the matching problem we propose is slightly different. In our formulation we define a decision variable for each feasible (arrival, departure) pair, while in Freling et al. Freling et al. (2005) the decision variables focus on individual units and ensure feasible matchings are obtained through constraints. Furthermore, the model of Freling et al. Freling et al. (2005) attempts to keep train units together as much as possible. We are only interested in feasibility. Finally, Freling et al. (2005) propose a heuristic column generation procedure to solve the resulting parking problem. We instead use a standard MIP solver.

6.7 – TYPE AND TRACK DECOMPOSITION

Some problem instances contain many events, many train unit types, or long tracks. This results in a large number of possible matchings or track assignments, which makes the problem impractical to solve using exact methods. For example, the CPM and RMM require too much memory, because they contain an explicit representation of the problem.

The solution space can be reduced significantly by decomposing the problem instances by train unit types and tracks. In the proposed decomposition, a train unit type is restricted to park on a select subset of tracks. The partitioning of the tracks and train unit types can be performed such that the original problem decomposes into several independent problems, which can be solved individually in sequence or in parallel. We consider such partitions where both the train unit types and tracks are partitioned into K groups, such that one group of train unit types is assigned to one group of tracks. By construction, no interaction needs to take place across the selected groups.

The decomposition divides the problem into a number of smaller independent subproblems. The primary advantage is that solving all resulting subproblems is easier than solving the full original problem. A second advantage is that the decomposition is independent of the underlying solver. The subproblems can be solved using any solution method for the TUSP. The primary disadvantage is that the resulting framework is inherently heuristic as the decomposition restricts the original solution space. Feasible solutions found using the decompositions are naturally also feasible in the original problem; however, we cannot conclude that a problem instance is infeasible if any one subproblem is infeasible. Another drawback of the proposed decomposition is the existence of multiple partitions. Some of the partitions may contain feasible solutions to all subproblems while others may not. Determining the partition is therefore another problem that must be addressed.

Due to the scope of this paper, we only propose a simple method of finding eligible partitions that will be explained and tested in Section 6.8. For the selected problem instances we generated a number of random partitions. Partitions are rejected if they do not pass the checks described in Section 6.4. In further research it could be interesting to investigate how to select good partitions.

6.8 – COMPUTATIONAL RESULTS

The presented solution methods are benchmarked on different classes of instances, which originate from three different railway networks in two different countries. Four classes, summarized in Tables 6.8, 6.9 and 6.10, are considered: STOG, DSB, NS, and NS-HARD. These instances are based on the railway networks of the Danish State Railways (DSB) and Netherlands Railways (NS) - the principal operators in Denmark and the Netherlands respectively.

All instances, except the DSB class, have been generated using a rolling stock optimizer. The events going in and out of the depots are extracted from the optimized schedule and define separate instances for each depot. Information about fleet size, train unit types, and depot track lengths are given by the railway operators. For the DSB class all arriving and departing events are explicitly given by the operator.

The STOG class consists of twelve distinct rolling stock schedules obtained by optimizing the suburban railway network in the greater Copenhagen area (DSB S-tog). This gives up to twelve different event lists per station. Identical problem instances have been eliminated resulting in a total of 96 instances for the STOG class.

The DSB class consists of real-life data for a recurring weekly schedule at the busiest station in Denmark, which is located in the center of Copenhagen. Every day in the weekly schedule is unique, thus resulting in seven instances for the DSB class.

The NS class consists of ten distinct rolling stock schedules for the whole country. This leads to ten different problem instances at eleven different stations. There are large differences between the event lists per station; some are large and some are small. Consequently, there are both difficult and relatively simple problem instances for the NS class. In total there are thus 110 instances.

The NS-HARD class is artificially constructed from the NS class, where fewer tracks are available at busy stations. These artificial cases are therefore more constrained in terms of capacity, in turn reducing the number of feasible parking plans. These problem instances have been included in an attempt to stress test the solution methods.

All computational experiments are performed on a dedicated machine equipped with two Intel(R) Xeon(R) CPU X5550 (2.67GHz) processors and 24 gigabytes of main memory. Version 12.6 of the commercial solver CPLEX is used to solve the MIP and CP based approaches. A time limit of 900 seconds is set for all experiments.

The following is a short summary of the solution methods proposed in Section 6.5 benchmarked in this section.

CPM A CP formulation inspired by the composition model in Fioole et al. (2006). The formulation is solved using the CPLEX constraint program solver.

Class	Depot	Min	Max	Tracks	Length	Types
STOG	BA	12	14	4	936	2
STOG	FM	16	66	4	727	2
STOG	FS	26	58	6	1 020	2
STOG	HI	20	54	6	1 635	2
STOG	HOT	2	22	1	173	2
STOG	HTAA	20	68	35	3 272	2
STOG	KH	36	78	9	2 753	2
STOG	KJ	24	60	6	1 115	2
STOG	KL	6	66	3	558	2
STOG	UND	24	46	6	1 670	2
DSB	KK	326	518	10	3 878	12

Table 6.8: Summary of the instances: The first column indicates to which class the problem belongs. The second column defines the station. The third and fourth columns show the minimum and maximum number of events taking place at the station. The fifth column presents the number of depot tracks available within the station and the sixth column defines the total length of all depot tracks combined. Finally, the seventh column defines the number of different train unit types that need to be parked within the station.

CPMH A variant of the CPM where the number of different train unit types assigned to the same track is limited.

CGM A column generation approach that assigns matching patterns to tracks.

RGCH A randomized greedy construction heuristic that is executed multiple times with different initial seeds.

RMM A reference MIP approach solved using the CPLEX MIP solver.

Class	Depot	Min	Max	Tracks	Length	Types
NS	AMR	157	159	9	2 267	4
NS	DDR	162	162	4	939	4
NS	EHV	153	179	20	7 061	4
NS	EKZ	97	97	5	1 590	4
NS	GVC	742	744	17	5 690	4
NS	HDR	82	82	3	1 143	4
NS	HFDO	561	561	8	3 020	4
NS	HN	75	75	12	2 023	4
NS	NM	268	268	25	6 495	4
NS	RTD	378	380	22	5 384	4
NS	ZP	87	87	9	4 127	4

Table 6.9: Continued summary of the instances.

Class	Depot	Min	Max	Tracks	Length	Types
NS-HARD	GVC14	742	744	14	4 712	4
NS-HARD	HFDO5A	561	561	5	1 934	4
NS-HARD	HFDO5B	561	561	5	1 898	4
NS-HARD	HFDO6	561	561	6	2 197	4
NS-HARD	NM10	268	268	10	2 457	4
NS-HARD	NM11	268	268	11	2 657	4
NS-HARD	RTD11	378	380	11	3 085	4
NS-HARD	RTD12	378	380	12	3 410	4
NS-HARD	RTD13	378	380	13	3 669	4

Table 6.10: Continued summary of the instances.

DCMM A variant of the RMM where the pairwise order conflict constraints are generated *on-the-fly*.

TSM A two-stage decomposition method that solves the matching and parking problem in sequence using MIP approaches.

Before presenting the results in detail, we first note that the CGM is discarded from further analysis. The performance of this method on all instances was always inferior in comparison to the other methods. For small cases the time it took to produce an optimal solution to the LP relaxation was significantly greater than the time the MIP based approaches took to produce a feasible solution. For the larger instances, CGM was unable to solve the root node relaxation in a Branch-and-Price (B&P) framework to LP optimality within the time limit in most cases. These results are consistent with Haahr et al. (2014), where a similar column generation approach was outperformed by a MIP approach for the parking problem only.

Tables 6.11 and 6.12 show a comparison overview. Table 6.11 shows the number of instances for which a feasible solution is found per problem class per method within the time limit. Table 6.12 shows the average runtimes per problem class per method.

In the STOG class, which contains the smallest problem instances, 94 out of the 96 instances are feasible. All methods, except the TSM, were able to find a feasible solution for those 94 instances. The TSM was unable to find a feasible solution for one of the instances. The average solution time is less than one second for all methods.

Solving the problem instances of the other classes directly proves to be impractical. We observe that the RMM fails to solve all but the relatively small STOG class problem instances. Significantly more cases can be solved by using the more efficient DCMM, CPM and CPMH variants. The DCMM can solve all DSB class instances and a large portion of the NS class instances, but only a few of the NS-HARD class instances. The time limit becomes a prohibiting factor when using the DCMM. The CPMH is more successful in solving the NS and NS-HARD class instances, but unable to solve the DSB class instances due to the large number of train unit types. The CPM performs relatively well compared to the RMM, but is clearly dominated by the CPMH in terms of solutions found and average runtimes.

The CPMH, RGCH and TSM perform well on all realistic instances as they are able to find the same number of feasible solutions. Further, RGCH and TSM are able to identify a feasible solution within a few seconds on average. However, for the NS-HARD class of problem instances TSM proves to be the most efficient heuristic. The CPMH was unable to solve some of the larger instances (GVC), while the RGCH was unable to solve the more constrained instances (HFDO5A and HFDO5B).

Class	No	CPM	CPMH	RGCH	RMM	DCMM	TSM
STOG	96	94	94	94	94	94	93
DSB	7	0	0	7	0	7	7
NS	110	84	101	110	0	93	110
NS-HARD	90	70	83	70	0	27	90

Table 6.11: Number of feasible instances found by the methods.

Table 6.13 shows the number of instances for which infeasibility is proven per method within the time limit. First, we note that the heuristic method RGCH is by definition not able to prove infeasibility. The two infeasible instances in the benchmark were detected by all exact approaches and the CPMH. The TSM was also unable to prove infeasibility as at least one feasible matching exists.

Class	CPM	CPMH	RGCH	RMM	DCMM	TSM
STOG	1.3	0.6	0.0	0.5	0.4	0.0
DSB			0.1		255.8	1.3
NS	20.5	6.0	0.0		148.0	5.8
NS-HARD	78.2	33.3	0.3		267.5	1.1

Table 6.12: Average runtimes for finding solutions grouped by method

Based on these results we can conclude that both the TSM and the RGCH method are very fast and efficient in finding feasible solutions. The CPMH is somewhat slower, but also successful in identifying feasible solutions in many cases. The RMM is clearly dominated by the DCMM, and the CPM by the CPMH. The DCMM solves fewer instances than the CPMH and requires more runtime; however, it can, in contrast, solve some instances with a higher number of train unit types.

Class	CPM	CPMH	RGCH	RMM	DCMM	TSM
STOG	2	2	0	2	2	0
DSB	0	0	0	0	0	0
NS	0	0	0	0	0	0
NS-HARD	0	0	0	0	0	0

Table 6.13: Number of instances proved to be infeasible by the methods.

Track Splitting

As explained in Section 6.7, the TUSP can be decomposed into several independent problems by partitioning the full problem by train unit types and tracks. In this section we investigate whether this decomposition technique can improve the tractability of the exact methods.

We have randomly generated a number of partitions of a selected set of problem instances with the following procedure. First, a random number of groups is selected, where this number is smaller than the number of train unit types and/ or tracks available. Second, tracks and train unit types are assigned randomly to the available groups. If any group has an empty set of train unit types or tracks, then the whole generation is rejected. Further, it is ensured that the maximum depot capacity required by the train unit types is less than the capacity of the tracks in the group. Finally, if train units are positioned initially in the depot, then this naturally adds constraints to the generation of the groups (e.g. if train units of different types are on the same track initially in the depot, then those two train unit types and the track where they are parked on are in the same group).

				CPMH			RGCH			DCMM		
Instance	#	F	Sub.	F	I	T	F	I	T	F	I	T
HFDO	10	10	20	20	0	0	20	0	0	20	0	0
GVC	10	10	20	20	0	0	20	0	0	4	0	16
RTD	10	10	30	30	0	0	30	0	0	30	0	0
DSB1	25	19	71	43	2	26	50	0	21	65	6	0
DSB2	25	18	71	40	4	27	50	0	21	64	6	1

Table 6.14: Summary of results achieved when running different methods on problem instances decomposed by splitting tracks and train unit types. The columns respectively show the instance considered, the number of decompositions, the number of feasible decompositions, the number of generated subproblems, and finally the number of **F***easible,* **I***nfeasible and* **T***imed-out instances for every method.*

The DCMM, CPMH and RGCH have been considered in this benchmark as they were unable to solve several instances in the previous section. The benchmark consists of large instances of the NS class that were unsolved by the RMM, DCMM and CPM. Decomposing this class of problems reduces the size of the underlying mathematical models significantly. Further, we consider two cases of the DSB class that were unsolved by the RMM, CPM and CPMH. A decomposition of these instances drastically reduces the number of variables and constraints in the CP model since the resulting number of train unit types is decreased.

An overview of the generated instances and results is shown in Table 6.14. The average runtimes are listed in Table 6.15. The HFDO, GVC and RTD instances were successfully decomposed, as all the resulting subproblems were feasible. The considered solution methods were able to solve these instances efficiently, except for the DCMM which was unable to produce a feasible solution for the subproblems of the largest instance. Nevertheless, DCMM is able to solve more instances using this decomposition technique. The DSB1 and DSB2 instances were on the contrary decomposed into both feasible and infeasible subproblems. The DCMM is able to solve all subproblems efficiently, except for one. The CPMH is able to solve more than half of the subproblems but several remained unresolved. This is an improvement compared to the non-decomposed results. Interestingly, decomposing the problem proved not to be efficient for the RGCH as many feasible subproblems were left unsolved. The RGCH was able to solve the original instances of the problems. A reduction in computation time is observed for both the DCMM and the RGCH in Table 6.15.

Instance	No	CPMH	RGCH	DCMM
HFDO	10	13.3	0.1	83.4
GVC	10	9.9	0.0	3.2
RTD	10	1.7	0.0	18.0
DSB1	25	26.4	0.1	42.2
DSB2	25	30.8	0.1	32.6

Table 6.15: Summary of average runtimes achieved when running different methods on problem instances decomposed by splitting tracks and train unit types. The columns respectively show the instance considered, the number of decompositions and finally the average runtime for all found solutions.

Using a decomposition of track and train unit types is shown to be worthwhile investigating. It is out of the scope of this paper to look further into different decomposition techniques. We do acknowledge that this could be an interesting future research direction.

Overnight Parking

In the common case all train units leave the depots during the early hours and enter the depot at the end of the day. Some train units enter and leave the depots during the day, e.g. before and after the rush hour periods. The capacity of the depots is therefore not very limiting during the day, which makes it easy to plan this intermediate period.

The considered problem instances do not stipulate any particular parking order at the end of the day. Consequently, no ordering conflicts arise regardless of the final track assignment, when the depots are close to being at capacity. Realistically, a smooth transition from one day to another is desirable, making sure that train units can leave the depot in a conflict free manner the following day. In our final benchmark, we combine the planning instances to include the events for two days of operation, thus forcing the solution methods to consider the overnight parking at platforms tracks.

Instance	Cases	CPMH	RGCH	DCMM	TSM
KH	3	3	3	2	1
FM	3	3	3	3	0
EHV	3	3	3	3	3
HDR	3	3	3	3	0

Table 6.16: Summary of results achieved when running different methods on problem instances with overnight parking. The columns respectively show the instance considered, the number of problem instances, and the number of solved instances for every method.

Tables 6.16 and 6.17 show the results of the overnight parking instances. The instances are naturally larger than the original ones, and require more time to solve as two days of events have been combined. The results show that the considered methods, except the TSM and DCMM, can efficiently solve all instances. Remember that all methods except the TSM integrate matching and parking. The TSM first finds one feasible matching and tries to assign the resulting matchings to tracks. Fixing the matching in a early stage does, however, restrict the flexibility when resolving the ordering conflicts. In contrast to the other benchmarks, these problem instances contain at least one very busy period, where the depots are close to being at capacity. Evidently, an approach that fixes, i.e., only considers one matching, may very well fail to find a feasible solution. The average runtimes shown in Table 6.17 reveal that the CPMH is faster than the DCMM in general when considering these extended instances. We note, however that these instances only contain 2-4 different train unit types. The performance of the CPMH is expected to decrease with higher numbers of different train unit types.

Instance	Cases	CPMH	RGCH	DCMM	TSM
KH	3	223.0	0.1	449.5	12.3
FM	3	1.1	0.0	8.2	0.0
EHV	3	11.9	0.1	244.4	0.4
HDR	3	1.0	0.0	17.7	0.1

Table 6.17: Summary of the average runtimes achieved when running different methods on problem instances with overnight parking. The columns respectively show the instance considered, the number of problem instances, and the average runtimes.

6.9 – CONCLUSION

In this paper we develop and benchmark different models and solution approaches to solve the Train Unit Shunting Problem. Given a feasible rolling stock circulation, the objective of the solution approaches is to find a feasible shunting plan. Three novel solution approaches are proposed: a CP formulation, a column generation approach, and a greedy construction heuristic. These new methods were compared with two methods from the literature. Computational experiments have been performed on multiple problem instances from three different railway operators. The benchmark highlights strengths and weaknesses of the considered approaches. A platform track parking extension is described, as platform tracks are currently used for overnight parking in some railway operations.

The main benchmark, consisting of multiple daily problem instances, demonstrated that the CGM was outperformed by all other methods. Furthermore, it revealed the shortcomings of exact models (RMM and CPM). The resulting mathematical models quickly consumed more than 24 gigabytes of memory due to the large number of constraints and/or variables required. Using delayed constraint generation (for the RMM) the DCMM is able to solve significantly more instances. The heuristic extension CPMH method (of CPM) further outperforms the DCMM when considering the instances with a small number of different train unit types. In turn, the DCMM can solve the instances with a high number of train unit types. On average, the solved instances were solved in a few minutes by these methods.

A randomized construction heuristic, the RGCH, was able to solve almost all instances within one second. However, some harder and artificially generated instances were left unsolved by the RGCH. In the main benchmark the non-integrated method TSM proved to be most successful, solving all but one of the feasible instances within a few seconds. Ironically, the unsolved instance was a relatively small problem instance.

The TUSP can be decomposed by splitting the available tracks and train unit types into several independent and smaller subproblems. A subset of the larger instances were decomposed in a second benchmark. In general, most of the resulting partitions were feasible. Using this decomposition the DCMM and CPMH were able to solve more problem instances than before, but not all. In fact, the RGCH performed worse than before as it was unable to find solutions for some of the problems that were solved originally.

In a final benchmark a number of instances were combined in order to solve two days of operation. The results show that the integrated methods are still able to solve these problems in reasonable time. However, the TSM was unable to solve most of the problem instances.

The considered solution approaches have both strengths and weaknesses. The results show that no method is superior. Solving the full mathematical formulations, i.e. RMM, CPM, directly proves to be ineffective. The DCMM is able to prove infeasibility in most cases. In addition, note that the proposed feasibility checks in Section 6.4 are very efficient - few instances were infeasible in general. Very fast solutions can be found using the RGCH but it proves to be inefficient for the more constrained instances. Finally, the TSM solves more instances using a few seconds, but has the drawback of using a fixed matching, which can be short sighted, leading to infeasibility. In conclusion, given the low runtime requirement of the RGCH and the TSM, these approaches form a reasonable choice as a first step in any solution framework. If no solution is found, then the DCMM and the CPMH can be adopted in a subsequent phase.

This paper focuses only on finding a feasible solution. No distinction is made between any feasible solution. In future research it might be interesting to extend the models by considering an objective in order to find a solution with, for instance, homogeneous tracks. Furthermore, several important restrictions have not been considered. If, for instance, the rolling stock circulation passes our feasibility check, it might still be infeasible with respect to the available crew members present for shunting operations at a station. Consequently, the crew has to be taken into account in future work on the TUSP. Other practical aspects need to be taken into account as well, e.g. detailed routing of the train units through the depot, parking whole compositions instead of single train units, train units might require maintenance at the station, and cyclic rolling stock circulations. In future research it would be interesting to include some or all of these aspects in the TUSP.

7 – Summary and conclusion

In this thesis, models and algorithms to support railway operators during and after a disruption are developed, compared, and benchmarked. As soon as a disruption occurs, the railway operator has to reschedule its timetable, rolling stock circulation, and crew schedule. This has to be done fast and in such a way that passengers are faced with the least amount of inconvenience. Current models in the existing literature are still lacking important real-life aspects. As a result, these models have not been applicable in practice yet. Therefore, this thesis aimed to answer the following research question:

"Which relevant operational details have not yet been included in the models for real-time disruption management processes, and how can these white spots in the existing models be filled?"

Five of the relevant operational details are included in the models for disruption management in this thesis. However, there are still operational details left that have to be included. Some of these white spots that are still left will be discussed in Section 7.2. The papers in this thesis are a next step in bridging the gap between the existing models in disruption management in theory and their usage in practice.

7.1 – MAIN FINDINGS

Chapter 2 proposes a framework to solve the three different disruption management models (timetable, rolling stock, and crew rescheduling) in an iterative way. This framework guarantees to find an overall feasible solution, thus a solution that is feasible for all three resource schedules. Three different settings of the iterative framework have been extensively tested on instances from Netherlands Railways. Results demonstrate that the setting where the crew is rescheduled before the rolling stock performs best. This is due to the fact that rolling stock specific constraints are already taken into account while rescheduling the timetable. As a consequence, there are usually no additionally cancelled train services after rescheduling the rolling stock. Thus, the rolling stock step can best be performed last. However, this is not always applicable in practice due to rolling stock dependent constraints for crew members. Furthermore, the results demonstrate that in all three settings a solution can be found within an amount of time that is reasonable for usage in real-time. As a consequence, the framework and the individual rescheduling modules can be of great importance in practice.

In the other chapters, the focus is on an individual step of the disruption management process: Rescheduling of rolling stock. First, Chapter 3 introduces and discusses two different models for rescheduling of rolling stock. The first model is a path-based formulation solved by means of column generation. In this formulation delayed transition contraint generation has been applied: Some of the constraints are removed initially and added only when violated. This improved the computation time significantly. The second model is an edge-based formulation based on a multi-commodity flow model. Both formulations are compared with each other on scheduling and rescheduling instances from Netherlands Railways and the Copenhagen Suburban Railway Operator DSB S-tog. Results demonstrate that the two formulations are not limited to operator specific networks. Both models can be applied within different countries. Furthermore, the results show that the formulation based on the multi-commodity flow model is faster. However, both models have acceptable computation times for usage in practice.

Chapter 4 includes a practical aspect required when solving the rolling stock rescheduling problem: Scheduled maintenance appointments. Some of the rolling stock units have a maintenance appointment at a station during the day. This has to be considered while rescheduling, otherwise the appointment will most likely be missed. Three different models are introduced to solve this problem: the Extra Unit Type (EUT), the Shadow Account (SA), and the Job Composition (JC) model. The EUT model is an extension of the known Composition model. By adding an additional rolling stock type for every unit requiring maintenance, constraints can be imposed upon the rolling stock units requiring maintenance such that they are in time for their appointment after rescheduling. The SA model keeps track of a shadow account for all units requiring maintenance. As a consequence, constraints can be imposed on the units with a maintenance appointment. Finally, the JC model assigns rolling stock units to jobs, thus creating specific paths for all rolling stock units. Constraints can be imposed on those paths, making sure that the maintenance units are on time for their appointment. In case the optimal solution has been found by all three models, they find a solution with the same optimal objective value. The results show that either the SA or JC model outperform the EUT model with respect to their computation times. It depends on the characteristics of the instance whether the SA or JC model performs best.

In Chapter 5, two other practical aspects that are required when solving the rolling stock rescheduling problem have been discussed. First of all, in practice operators can use dead-heading trips to increase the local inventory at certain stations. In this way less train services may be cancelled due to lack of rolling stock. Secondly, the forecasted passenger demand has to be adapted during a disruption. The passenger demand for a trip depends on which compositions are appointed to the (predecessor) trips (e.g. cancelling a trip increases the demand for the next trip with the same origin and destination). Six different versions of the model, depending on whether dead-heading trips and adjusted passenger demand are incorpared or not, are tested on multiple disruption scenarios from Netherlands Railways. Results demonstrate, first of all, that dead-heading trips are useful to reduce the number of cancelled train services and the amount of seat shortages for passengers. Furthermore, adjusted passenger demand influences the results heavily. The resulting rolling stock circulation after rescheduling with adjusted passenger demand appoints compositions with large capacity to the trips between the stations where the disruption took place. In this way, the passengers still waiting on a platform are transported to their destination sooner. Finally, including dead-heading trips and adjusted passenger demand have an influence on the computation times, but the computation times are still acceptable for usage in practice.

Finally, Chapter 6 discusses the last practical aspect that is included in the disruption management process in this thesis. Current rolling stock (re)scheduling models assume that the resulting rolling stock circulation is feasible with respect to the available depots within stations. However, in practice this is not always true. It is of utmost importance to the railway operator to know in short time whether the rolling stock circulation is feasible. Therefore, we have developed a Constraint Programming formulation, a Column Generation approach, and a randomized greedy heuristic to test the feasibility of a rolling stock circulation with respect to the available depots at stations. These methods are compared and benchmarked against slightly adjusted existing methods which are based on a Mixed Integer Programming formulation and a two-stage heuristic. The results demonstrate first of all that the method based on column generation is always outperformed by the other methods. Furthermore, the results show that all other solution approaches have both strengths and weaknesses. None of the methods is superior to all other methods. Very fast feasible solutions can be found by using the randomized greedy heuristic or the two-stage approach, however both methods are unable to prove infeasibility if a circulation is actually infeasible. We conclude that it is best to first use either the randomized greedy or the two-stage heuristic in an initial step and, if no solution is found, to continue with either the method based on Constraint Programming or the method based on the Mixed Integer Programming formulation.

Scientific contributions

This thesis focussed on developing algorithmic support for disruption management while including important practical aspects. In order to do so, a wide variety of models from the field of Operations Research has been applied. This had led to complex mathematical programming formulations which can be solved within acceptable time for usage in practice either by a commercial solver like Cplex, by means of a Column Generation technique, or by heuristic approaches. In this section we briefly highlight the contributions of each chapter made to the scientific community.

Chapter 2 is the first to develop an iterative framework that can be used to solve the whole macroscopic disruption management process. It guarantees to find a feasible solution for all three macroscopic resource schedules.

Chapter 3 introduces a new mathematical formulation for the rolling stock rescheduling problem, which is solved by means of column and row generation. This column and row generation approach is new due to the fact that the order of train units is taken into account.

Chapter 4 includes scheduled maintenance appointments in the mathematical formulations for rescheduling the rolling stock. To this end, three formulations are proposed and an extensive comparison is made between the models.

Chapter 5 is the first to take dead-heading trips and adjusted passenger demand into account in a model for rescheduling the rolling stock. Furthermore, this chapter presents an efficient preprocessing model to select potentially useful dead-heading trips from the huge set of available dead-heading trips.

Chapter 6 proposes a constraint program, a column generation technique, and a greedy construction heuristic for solving the Train Unit Shunting Problem (TUSP). Furthermore, several efficient infeasibility checks are used in a preprocessing step for the TUSP. Finally, a track and unit type decomposition approach for all models to reduce a TUSP instance to several smaller and independent TUSP instances are proposed.

With these contributions this thesis shows the applicability of Operations Research to solve relevant practical aspects for railway operators in various countries.

Social relevance

This thesis demonstrates that both passengers and railway operators can benefit from effective models during the disruption management processes. The research in this thesis is a next step in bridging the gap between models and algorithms in theory and their usage in practice. The proposed approaches in this thesis were tested on instances from Netherlands Railways and from Danish State Railways. The results demonstrate that the models are not limited to a specific operator only and that they can be of great value to the railway operators in practice. They can support dispatchers during disruptions to find new feasible resource schedules.

The models developed in this thesis can support the dispatchers either for the disruption management process as a whole, for the rolling stock rescheduling problem, or for the train unit shunting problem. As a consequence, using these models in practice will reduce the time it takes before the new resource schedules are operational and communicated to all people involved. This results in less inconvenience for the passengers and less time stress for the railway operators.

The objective of the developed methods is, among others, to minimize the number of cancelled train services and to appoint compositions with sufficient capacity for all passenger demand to the trips. As a result, more trains will be operated and passengers have to wait a shorter amount of time before arriving at their destination.

Furthermore, the railway operators benefit from the inclusion of certain practical aspects (e.g. scheduled maintenance appointments in the RSRP and feasibility tests for testing the rolling stock circulation with respect to the depots). As a consequence, railway dispatchers can be aided by the methods during rescheduling. This will lead to a reduction of the time it takes to reschedule the resource schedules. Moreover, the quality of the resource schedules will most likely increase.

7.2 – FUTURE RESEARCH

The results in this thesis demonstrate the applicability of our models for railway operators. However, there is still much work to be done in the field of disruption management. First of all, before the presented models in this thesis can actually be applied, they first have to be integrated within the existing systems at the railway operators. As a consequence, the models are supplied with the necessary input and the resulting output can immediately be applied. Furthermore, there is a certain trade-off between costs and passenger service. The railway operators should decide on the objective coefficients for all models in order to balance this trade-off. Moreover, the presented methods in this thesis should be combined into one model, such that all presented practical aspects are taken into account at once.

Finally, several important practical aspects are still missing in the presented models. First of all, rescheduling the timetable on a microscopic level is not considered simultaneously with rescheduling the timetable on a macroscopic level. However, a timetable should be feasible both on a microscopic and on a macroscopic level. Otherwise the resulting timetable cannot be used in practice. Therefore, either an integrated model or an iterative framework should be developed for creating a completely feasible timetable.

Secondly, station routing has not been included in any of the disruption management models yet. The departure and arrival times of a subset of trains have changed due to the disruption. As a result, the originally planned schedule for the routing of trains through the involved stations are no longer feasible. Models and algorithms should be developed in order to secure a feasible routing for every train through every station.

Thirdly, the duration of a disruption is unknown at the start of the disruption. Currently, the duration is estimated based on the experience of the dispatchers. Railway operators have a lot of historical data at their disposal. As a consequence, it should be possible to investigate whether a better estimate of the duration of a disruption can be found.

Fourthly, because the duration of a disruption is unknown, robust rolling stock schedules after rescheduling should be created. If a disruption turns out to take longer than expected, the railway operator wants to create a new feasibile circulation as soon as possible. As a consequence, it would be best if this can be done without rescheduling the rolling stock circulation completely as done after the occurence of the disruption. This should be taken into account when rescheduling by creating robust rolling stock circulations.

Finally, the researchers within the railway sector should keep an eye on the developments to come. For instance, the railway sector will develop rolling stock units which drive on their own in the near future. As a consequence, this will bring new challenges to the field.

REFERENCES

E. Abbink, D. Mobach, P. Fioole, L. Kroon, E. van der Heijden, and N. Wijngaards. Actor-agent application for train driver rescheduling. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems*, pages 513–520, May 2009.

R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice-Hall Englewood Cliffs, 1993.

R. K. Ahuja, J. Liu, J. B. Orlin, D. Sharma, and L. A. Shughart. Solving real-life locomotive-scheduling problems. *Transportation Science*, 39(4):503–517, 2005.

A. Alfieri, R. Groot, L. Kroon, and A. Schrijver. Efficient circulation of railway rolling stock. *Transportation Science*, 40(3):378–391, 2006.

C. Barnhart, N. Boland, L. Clarke, E. Johnson, G. Nemhauser, and R. Shenoi. Flight string models for aircraft fleeting and routing. *Transportation Science*, 32(3):208–220, 1998.

R. Borndörfer, M. Reuther, T. Schlechte, and S. Weider. Integrated optimization of rolling stock rotations for intercity railways. *To appear in: Transportation Science*, 2015. doi: 10.1287/trsc.2015.0633.

B. Bouzaiene-Ayari, C. Cheng, S. Das, R. Fiorillo, and W. B. Powell. From single commodity to multiattribute models for locomotive optimization: A comparison of optimal integer programming and approximate dynamic programming. *To appear in: Transportation Science*, 2014. doi: 10.1287/trsc.2014.0536.

C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.

P. Brucker, J. Hurink, and T. Rolfes. Routing of railway carriages. *Journal of Global Optimization*, 27(2-3):313–332, 2003.

G. Budai, G. Maróti, R. Dekker, D. Huisman, and L. Kroon. Rescheduling in passenger railways: the rolling stock rebalancing problem. *Journal of Scheduling*, 13:281–297, 2010.

V. Cacchiani, A. Caprara, and P. Toth. Solving a real-world train-unit assignment problem. *Mathematical Programming*, 124(1-2):207–231, 2010.

V. Cacchiani, A. Caprara, L. Galli, L. Kroon, G. Maróti, and P. Toth. Railway rolling stock planning: Robustness against large disruptions. *To apear in Transportation Science*, 2011.

V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, and J. Wagenaar. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, 63:15 – 37, 2014.

L. Cadarso and A. Marín. Robust rolling stock in rapid transit networks. *Computers & Operations Research*, 38(8):1131 – 1142, 2011.

L. Cadarso and A. Marín. Improving robustness of rolling stock circulations in rapid transit networks. *Computers & Operations Research*, 51:146 – 159, 2014.

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

L. Cadarso, G. Maróti, and Á. Marín. Smooth and controlled recovery planning of disruptions in rapid transit networks. *Intelligent Transportation Systems, IEEE Transactions on*, 16(4): 2192–2202, 2015.

L. Clarke, E. Johnson, and G. Nemhauser. The aircraft rotation problem. *Annals of Operations Research*, 69:139–151, 1997.

J.-F. Cordeau, F. Soumis, and J. Desrosiers. Simultaneous assignment of locomotives and cars to passenger trains. *Operations Research*, 49(4):531–548, 2001.

J.-F. Cordeau, G. Desaulniers, N. Lingaya, F. Soumis, and J. Desrosiers. Simultaneous locomotive and car assignment at via rail canada. *Transportation Research Part B*, 35(8): 767–787, 2002.

F. Corman, A. D'Ariano, I. Hansen, D. Pacciarelli, and M. Pranzo. Dispatching trains during seriously disrupted traffic situations. In *Proceedings of the IEEE International Conference on Networking, Sensing and Control*, pages 323–328, 2011.

A. D'Ariano, D. Pacciarelli, and M. Pranzo. A branch and bound algorithm for scheduling trains on a railway network. *European Journal of Operational Research*, 183(2):643–657, 2007.

G. Desaulniers, J. Desrosiers, and M. M. Solomon. *Column generation*, volume 5. Springer, 2005.

T. Dollevoet, D. Huisman, L. Kroon, L. Veelenturf, and J. Wagenaar. An iterative framework for real-time railway rescheduling. Technical report, Erasmus University Rotterdam, 2015.

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*, 174 (2):1281–1297, 2006.

R. Freling, R. M. Lentink, L. G. Kroon, and D. Huisman. Shunting of passenger train units in a railway station. *Transportation Science*, 39(2):261–272, 2005.

A. Fügenschuh, H. Homfeld, A. Huck, A. Martin, and Z. Yuan. Scheduling locomotives and car transfers in freight transport. *Transportation Science*, 42(4):478–491, 2008.

M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63. ACM, 1974.

G. Giacco, A. DAriano, and D. Pacciarelli. Rolling stock rostering optimization under maintenance constraints. *Journal of Intelligent Transportation Systems*, 18(1):95–105, 2014.

J. Haahr, R. Lusby, J. Larsen, and D. Pisinger. A branch-and-price framework for railway rolling stock rescheduling during disruptions. Technical report, DTU Management engineering, 2014. URL `http://findit.dtu.dk/en/catalog/2203007514`.

J. Haahr, R. Lusby, J. Larsen, and D. Pisinger. *Simultaneously recovering rolling stock schedules and depot plans under disruption*. Proceedings of the 13th Conference on Advanced Systems in Public Transport (CASPT), 2015a.

J. Haahr, R. Lusby, and J. Wagenaar. A comparison of optimization methods for solving the depot matching and parking problem. Technical report, 2015b.

J. Haahr, J. Wagenaar, L. Veelenturf, and L. Kroon. A comparison of two exact methods for passenger railway rolling stock (re)scheduling. Technical Report ERS-2015-007-LIS, ERIM Report Series Research in Management, 2015c. URL `http://hdl.handle.net/1765/78317`.

R. Haijema, C. Duin, and N. Van Dijk. Train shunting: A practical heuristic inspired by dynamic programming. *Planning in Intelligent Systems: Aspects, Motivations, and Methods*, pages 437–475, 2006.

E. van der Hurk, G. Maróti, L. Kroon, and P. Vervest. Deduction of passengers' route choice from smart card data. *IEEE Transactions on ITS*, 16:430–440, 2015.

P. M. Jacobsen and D. Pisinger. Train shunting at a workshop area. *Flexible services and manufacturing journal*, 23(2):156–180, 2011.

J. Jespersen-Groth, D. Potthoff, J. Clausen, D. Huisman, L. Kroon, G. Maróti, and M. N. Nielsen. *Disruption management in passenger railway transportation*. Springer, 2009.

R. M. Karp. *Reducibility among combinatorial problems*. Springer, 1972.

L. Kroon, G. Maróti, and L. Nielsen. Rescheduling of railway rolling stock with dynamic passenger flows. *Transportation Science*, 49(2):165–184, 2014.

L. G. Kroon, R. M. Lentink, and A. Schrijver. Shunting of passenger train units: an integrated approach. *Transportation Science*, 42(4):436–449, 2008.

H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

L. Lamorgese and C. Mannino. An exact decomposition approach for the real-time train dispatching problem. *Operations Research*, 63(1):48–64, 2015.

R. M. Lentink, P.-J. Fioole, L. G. Kroon, and C. van't Woudt. Applying operations research techniques to planning of train shunting. *Planning in Intelligent Systems: Aspects, Motivations, and Methods*, pages 415–436, 2006.

N. Lingaya, J. Cordeau, G. Desaulniers, J. Desrosiers, and F. Soumis. Operational car assignment at VIA Rail Canada. *Transportation Research Part B: Methodological*, 36(9):755–778, 2002.

I. Louwerse and D. Huisman. Adjusting a railway timetable in case of partial or complete blockades. *European Journal of Operational Research*, 235(3):583–593, 2014.

M. Lübbecke and J. Desrosiers. Selected topics in column generation. *Operations Research*, 53:1007–1023, 2004.

G. Maróti. Operations research models for railway rolling stock planning. *PhD thesis, Eindhoven University of Technology, The Netherlands*, 2006.

G. Maróti and L. Kroon. Maintenance routing for train units: the transition model. *Transportation Science*, 39(4):518–525, 2005.

G. Maróti and L. Kroon. Maintenance routing for train units: the interchange model. *Computers & Operations Research*, 34(4):1121–1140, 2007.

T. Mellouli and L. Suhl. Rotation planning of locomotive and carriage groups with shared capacities. In *Algorithmic Methods for Railway Optimization*, pages 276–294. Springer, 2007.

J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

L. Nielsen. Rolling stock rescheduling in passenger railways. *Ph.D thesis, Erasmus University, Rotterdam, The Netherlands*, 2011.

L. Nielsen, L. Kroon, and G. Maróti. A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, 220(2):496–509, 2012.

M. Peeters and L. Kroon. Circulation of railway rolling stock: a branch-and-price approach. *Computers and Operations Research*, 35(2):538 – 556, 2008.

D. Pisinger. *Algorithms for Knapsack Problems*. PhD thesis, DIKU, University of Copenhagen, Denmark, 1995. Technical Report 95-1.

F. Piu and M. G. Speranza. The locomotive assignment problem: a survey on optimization models. *International Transactions in Operational Research*, 21(3):327–352, 2014.

D. Potthoff, D. Huisman, and G. Desaulniers. Column generation with dynamic duty selection for railway crew rescheduling. *Transportation Science*, 44(4):493–505, 2010.

N. Rezanova and D. Ryan. The train driver recovery problem- a set partitioning based model and solution method. *Computers & Operations Research*, 37(5):845–856, 2009.

S. Rouillon, G. Desaulniers, and F. Soumis. An extended branch-and-bound method for locomotive assignment. *Transportation Research Part B: Methodological*, 40(5):404 – 423, 2006.

K. Sato and N. Fukumura. Real-time freight locomotive rescheduling and uncovered train detection during disruptions. *European Journal of Operational Research*, 221:636–648, 2012.

K. Sato, S. Sakikawa, T. Morita, N. Ueki, and T. Murata. Crew and vehicle rescheduling based on a network flow model and its application to a railway train operation. *IAENG International Journal of Applied Mathematics*, 39(3), 2009.

K. Talluri. The four day aircraft maintenance routing problem. *Transportation Science*, 32(1): 43–53, 1998.

B. Vaidyanathan, R. K. Ahuja, and J. B. Orlin. The locomotive routing problem. *Transportation Science*, 42(4):492–507, 2008.

E. van der Hurk. *Passengers, Information and Disruptions*. PhD thesis, Erasmus University Rotterdam, 2015.

L. Veelenturf, D. Potthoff, D. Huisman, and L. Kroon. Railway Crew Rescheduling with Retiming. *Transportation Research Part C: Emerging Technologies*, 20(1):95 – 110, 2012.

L. Veelenturf, D. Potthoff, D. Huisman, L. Kroon, G. Maróti, and A. Wagelmans. A quasi-robust optimization approach for crew rescheduling. *Transportation Science*, Available Online, 2014. doi: 10.1287/trsc.2014.0545.

L. Veelenturf, M. Kidd, V. Cacchiani, L. Kroon, and P. Toth. A railway timetable rescheduling approach for handling large-scale disruptions. *Transportation Science*, Available Online, 2015. doi: 10.1287/trsc.2015.0618.

J. Wagenaar and L. Kroon. Rolling stock rescheduling in passenger railway transportation using dead-heading trips and adjusted passenger demand. Technical report, 2016.

J. Wagenaar, L. G. Kroon, and M. Schmidt. Maintenance appointments in railway rolling stock rescheduling. *Available at SSRN 2712465*, 2016.

C. Walker, J. Snowdon, and D. Ryan. Simultaneous disruption recovery of a train timetable and crew roster in real time. *Computers & Operations Research*, 32:2077–2094, 2005.

T. Winter. *Online and Real-Time Dispatching Problems*. PhD thesis, Technical University, Braunschweig, Germany, 1999.

S. Zhan, L. Kroon, L. Veelenturf, and J. Wagenaar. Real-time high-speed train rescheduling in case of a complete blockage. *Transportation Research Part B: Methodological*, 78:182 – 201, 2015.

Nederlandse Samenvatting (Summary in Dutch)

Verstoringen zijn helaas een dagelijks probleem binnen spoorwegnetwerken. Tijdens een verstoring moeten de dienstregeling, de materieel planning en de personeelsplanning opnieuw gepland worden. Dit moet zo snel mogelijk gedaan worden en zodanig dat de passagiers zo min mogelijk hinder ondervinden van de verstoring. Voor dit proefschrift zijn bijsturingsmodellen en algoritmes ontwikkeld om de spoorwegvervoerders te assisteren gedurende het herplannen. Deze modellen en algoritmes zijn zowel met elkaar, als met de modellen uit de bestaande literatuur vergeleken. Daarnaast zijn ze gerangschikt naar kwaliteit. De modellen in de bestaande literatuur missen belangrijke practische aspecten, waardoor deze modellen niet toepasbaar zijn in de praktijk. Daarom richt dit proefschrift zich op het beantwoorden van de volgende onderzoeksvraag:

"Welke relevante operationele details bevinden zich nog niet in de bestaande bijsturingsmodellen en hoe kunnen deze ontbrekende aspecten wel meegenomen worden?"

Vijf van deze operationele details zijn meegenomen in de in dit proefschrift ontwikkelde bijsturingsmodellen. Dat betekent echter niet dat alle operationele details op dit moment meegenomen zijn. In hoofdstuk 7.2 worden suggesties voor verdere verbeteringen behandeld. Hopelijk is dit proefschrift een volgende stap naar een groter gebruik van bijsturingsmodellen in de praktijk.

In hoofdstuk 2 wordt een framework voorgesteld om de dienstregeling, materieelplanning en personeelsplanning op een iteratieve manier op te lossen. Dit framework vindt altijd een toegestane oplossing voor zowel een nieuwe dienstregeling als voor een nieuw materieel- en personeelsplan. Drie verschillende versies van het framework zijn uitgebreid getest op instanties van de Nederlandse Spoorwegen. De resultaten demonstreren dat het effectiever is om eerst de dienstregeling, dan het personeel en als laatste het materieel te herplannen. Dit komt doordat er materieel specifieke constraints zijn meegenomen tijdens het herplannen van de dienstregeling, daardoor worden er vrijwel geen extra ritten meer gecanceld tijdens het bijsturen van het materieel. Helaas is het niet altijd toepasbaar in de praktijk om eerst het personeel en

daarna het materieel te herplannen. Dit komt doordat er soms materieel restricties zijn voor personeel. De resultaten laten verder zien dat het mogelijk is om snel genoeg een toegelaten dienstregeling en materieel-personeelsplan te vinden voor gebruik in real-time. Concluderend kan gesteld worden dat het framework van grote waarde kan zijn in de praktijk.

De focus ligt op een van de individuele bijsturingsmodellen in de andere hoofdstukken: het bijsturen van het materieel. In hoofdstuk 3 worden hier twee verschillende modellen voor geïntroduceerd. Het eerste model is gebaseerd op formulatie met routes voor een individueel treinstel. Deze formulatie wordt opgelost met behulp van kolom generatie. In deze formulatie worden sommige restricties in het begin verwijderd en alleen later toegevoegd als ze geschonden blijken te zijn. Dit heeft de rekentijd aanzienlijk verbeterd. Het tweede model is gebaseerd op het multi-commodity flow model. Beide formulaties worden met elkaar vergeleken op zowel plannings als bijsturings instanties van de Nederlandse Spoorwegen en van de Kopenhaagse Spoorwegen DSB S-TOG. De resultaten laten zien dat beide formulaties toepasbaar zijn op spoorwegnetwerken van verschillende landen. Daarnaast laten de resultaten zien dat de formulatie die gebaseerd is op het multi-commodity flow model sneller een oplossing vindt. Beide modellen hebben een rekentijd die toepasbaar is in de praktijk.

In hoofdstuk 4 wordt een operationeel aspect toegevoegd aan het bijsturingsmodel voor het materieel: geplande onderhoudsafspraken. Sommige treinstellen hebben een onderhoudsafspraak op een specifiek station gedurende de dag. Dit moet meegenomen worden tijdens het bijsturen, anders mist het treinstel zijn afspraak hoogstwaarschijnlijk. Drie verschillende modellen zijn hiervoor ontwikkeld: het Extra Unit Type (EUT) model, het Shadow Account (SA) model, en het Job Composition (JC) model. Het EUT model is een uitbreiding van het al bestaande Composition model. Door middel van het toevoegen van extra materieel types voor elk treinstel dat onderhoud vereist is het mogelijk om specifieke restricties op die treinstellen te zetten. Op deze manier zijn de juiste treinstellen nog altijd op tijd voor hun afspraak. Het SA model houdt een schaduwboekhouding bij voor alle treinstellen die een onderhoudsafspraak hebben. Daardoor kunnen er restricties op de juiste treinstellen gezet worden. Het JC model wijst treinstellen toe aan taken in plaats van aan ritten. Op deze manier worden specifieke routes gecreëerd voor alle treinstellen en kunnen er restricties gezet worden op de treinstellen met een onderhoudsafspraak. Alle drie de modellen vinden dezelfde optimale oplossing. De resultaten laten zien dat het SA en het JC model het best presteren met betrekking tot de rekentijden. Het hangt van de karakteristieken van de instanties af of het SA model of het JC model het beste presteert.

In hoofdstuk 5 worden twee andere operationele aspecten toegevoegd aan het bijsturingsmodel voor het materieel. Ten eerste worden dead-heading treinen toegevoegd. In de praktijk kunnen de spoorvervoerders gebruik maken van dead-heading treinen om de voorraad op bepaalde stations te verhogen. Op deze manier vallen er minder ritten uit doordat er geen treinstellen beschikbaar zijn. Ten tweede moet de passagiersvraag aangepast worden gedurende een verstoring. De passagiersvraag voor een rit hangt af van welke trein er wordt toegewezen aan de voorgaande ritten. Bijvoorbeeld, na het uitvallen van een rit zal er een grotere passagiersvraag zijn op de volgende rit. Zes verschillende versies van het model zijn er getest. Deze versies verschillen in of dead-heading treinen en/of een veranderde passagiersvraag is meegenomen in het model. Alle versies zijn getest op verschillende instanties van de Nederlandse Spoorwegen. De resultaten laten ten eerste zien dat dead-heading treinen het aantal uitgevallen treinen reduceren en daarnaast ook het aantal zitplaatstekorten voor passagiers verminderen. Daarnaast laten de resultaten zien dat een veranderde passagiersvraag een grote invloed uitoefent op de resultaten. De nieuwe materieel planning zet treinen met grote capaciteit in op de ritten vlak na een verstoring om het aantal zitplaatstekorten voor passagiers te minimaliseren. Het toevoegen van dead-heading treinen en een veranderde passagiersvraag heeft invloed op de rekentijd, maar de rekentijd is nog altijd goed genoeg voor gebruik in de praktijk.

In hoofdstuk 6 wordt het laatste operationele aspect in dit proefschrift meegenomen. De huidige bijsturingsmodellen voor het materieel nemen aan dat de gevonden materieelplanning toegestaan is met betrekking tot de rangeerterreinen binnen de stations. In de praktijk daarentegen is dit niet altijd waar. Het is van uitermate belang voor een spoorvervoerder om in korte tijd erachter te komen of een materieelplanning daadwerkelijk realiseerbaar is. Daarom worden er in hoofdstuk 6 verschillende modellen voorgesteld om dit te testen: een constraint programming formulatie, een kolom generatie methode, en een randomized greedy heuristic. Deze nieuwe methoden zijn met elkaar en met (licht aangepaste) bestaande modellen (een mixed integer programming formulatie en een two-stage heuristic) vergeleken. De resultaten laten ten eerste zien dat de kolom generatie methode altijd slechter werkt dan de andere methodes. Daarnaast blijkt dat alle andere methoden zowel sterke als zwakke kanten hebben. Geen van de andere methoden is superieur. Toegestane oplossingen kunnen heel snel gevonden worden door de randomized greedy heuristic en de two-stage heuristic, maar beide modellen zijn niet in staat om aan te tonen dat een materieelplanning niet realiseerbaar is. Concluderend kan gesteld worden dat het het beste is om eerst de randomized greedy heuristic of de two-stage heuristic te gebruiken en na na verloop van tijd verder te gaan met de methode gebaseerd op constraint programming of de mixed integer programming formulatie.

Het onderzoek in dit proefschrift heeft laten zien dat operationele aspecten effectief meegenomen kunnen worden in de bijsturingsmodellen. De modellen zijn getest op instanties van zowel de Nederlandse Spoorwegen als de Deense Spoorwegen. De modellen zijn toepasbaar in de praktijk, omdat de oplossingen in een korte tijd gevonden kunnen worden.

Curriculum Vitae

Joris Camiel Wagenaar was born February 11th 1989 in Rotterdam, the Netherlands. In 2006 he graduated from highschool at College het Loo in Voorburg. He first obtained a B.Sc in Econometrics and Operations Research at the Erasmus School of Economics, Erasmus University Rotterdam. Thereafter he got a master's degree in Econometrics and Management Science, specialized in Operations Research and Quantitative Logistics. He started his PhD research in practice oriented algorithmic disruption management in passenger railways in 2012 at the Rotterdam School of Management. Furthermore, he spent three months at the Technical Univerisity of Denmark to work closely together with the research group of Professor David Pissinger.

The research described in this thesis has been presented at many international conferences, such as TRISTAN, CASPT, INFORMS, and IFORS. Furthermore, his research has led to several awards and publications. At the moment of writing, he has 2 publications in Transportation Research Part B: Methodological, two publications in Transportation Science, and one publication in Transportation Research Part E: Logistics and Transportation Review. Furthermore, three of his papers are in reviewing process. He got the third and second prize in the Student Paper Award competition of the Railway Application Section of Informs in 2014 and 2015. He also won the best student paper award at the TRAIL conference in 2013.

Author portfolio

Publications

Published:

L. Kroon, L. Peeters, **J. Wagenaar**, R. Zuidwijk,
"Flexible Connections in PESP Models for Cyclic Passenger Railway Timetabling",
Transportation Science, 48 (1) (2013), pp 136 - 154

V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, **J. Wagenaar**,
"An Overview of Recovery Models for Real-Time Railway Rescheduling",
Transportation Research Part B: Methodological 63 (2014), pp 15-37

S. Zhan, L. Kroon, L. Veelenturf, **J. Wagenaar**
"Real-Time High-Speed Train Rescheduling in Case of a Complete Blockage",
Transportation Research Part B: Methodological 78 (2015), pp 182-201

J. Wagenaar, L. Kroon, M. Schmidt
"Maintenance Appointments in Railway Rolling Stock Rescheduling",
Submitted to Transportation Science

J. Haahr, L. Kroon, L. Veelenturf, **J. Wagenaar**
"A Comparison of Two Exact Methods for Passenger Railway Rolling Stock (Re)Scheduling
Submitted to Transportation Research Part E: Logistics and Transportation Review

Submitted:

T. Dollevoet, D. Huisman, L. Kroon, L. Veelenturf, **J. Wagenaar**
"An Iterative Framework for Real-Time Railway Rescheduling"
Computers and Operations Research

J.T. Haahr, R.M. Lusby, **J. Wagenaar**
"A Comparison of Optimization Methods for Solving
the Depot Matching and Parking Problem"
Submitted to Transportation Research Part E: Logistics and Transportation Review

J. Wagenaar, L. Kroon
"Rolling Stock Rescheduling in Passenger Railway Transportation Using
Dead-Heading Trips and Adjusted Passenger Demand",
Submitted to Transportation Research Part B: Methodological

Teaching Experience

Mathematics	**First year bachelor students in Business Administration**
Coordinator 2014-2016	*Tasks include:* Coordinating all student assistants, setting up the course
Teaching assistant 2012-2014	lecturing exercise lectures to groups of 40 students, running the back office of both dutch and the international Mathematics course.
Introduction to Statistics	**First year bachelor students in Econometrics**
Teaching assistant 2010-2012	*Tasks include:* Giving the computer exercise lectures to groups of 80 students.
Methods & Techniques	**Third year bachelor students in Economics**
Teaching assistant 2009-2012	*Tasks include:* Giving the computer exercise lectures to groups of 80 students.

Introduction to Econometrics	**Master students in Economics**
Teaching assistant 2010-2012	*Tasks include:* Giving the computer exercise lecures to groups of 80 students.
Mathematics 1 & 2	**First year bachelor students in Economics**
Teaching assistant 2010-2012	*Tasks include:* Leading workshops with 40 students.
Applied Statistics 1	**First year bachelor students in Economics**
Teaching assistant 2010-2012	*Tasks include:* Leading workshops with 40 students.

Conference Presentations

2015:	**INFORMS**, Philadelphia, USA
2015:	**CASPT**, Rotterdam, The Netherlands
2015:	**RailTokyo**, Tokyo, Japen
2014:	**INFORMS**, San Francisco, USA
2014:	**IFORS**, Barcelona, Spain
2013:	**TRAIL congres**, Delft, The Netherlands
2013:	**OR2013**, Rotterdam, The Netherlands
2013:	**TRISTAN VIII**, San Pedro de Atacama, Chile
2013:	**RailCopenhagen**, Copenhagen, Denmark

Awards

2013:	**TRAIL congres best paper award**	first place
2014:	**RAS student paper award**	third place
2014:	**RAS problem solving competition**	third place
2015:	**RAS student paper award**	second place

Selected skills and languages

Computer skills:	**Java, Matlab, OPL studio, SPSS**
Languages:	**Dutch (native), English (near native)**

The ERIM PhD Series

The ERIM PhD Series contains PhD dissertations in the field of Research in Management defended at Erasmus University Rotterdam and supervised by senior researchers affiliated to the Erasmus Research Institute of Management (ERIM). All dissertations in the ERIM PhD Series are available in full text through the ERIM Electronic Series Portal: http://repub.eur.nl/pub. ERIM is the joint research institute of the Rotterdam School of Management (RSM) and the Erasmus School of Economics at the Erasmus University Rotterdam (EUR).

Dissertations in the last five years

Abbink, E.J., *Crew Management in Passenger Rail Transport*, Promotors: Prof. L.G. Kroon & Prof. A.P.M. Wagelmans, EPS-2014-325-LIS, http://repub.eur.nl/pub/76927

Acar, O.A., *Crowdsourcing for Innovation: Unpacking Motivational, Knowledge and Relational Mechanisms of Innovative Behavior in Crowdsourcing Platforms*, Promotor: Prof. J.C.M. van den Ende, EPS-2014-321-LIS, http://repub.eur.nl/pub/76076

Akin Ates, M., *Purchasing and Supply Management at the Purchase Category Level: strategy, structure and performance*, Promotors: Prof. J.Y.F. Wynstra & Dr E.M. van Raaij, EPS-2014-300-LIS, http://repub.eur.nl/pub/50283

Akpinar, E., *Consumer Information Sharing*, Promotor: Prof. A. Smidts, EPS-2013-297-MKT, http://repub.eur.nl/pub/50140

Alexander, L., *People, Politics, and Innovation: A Process Perspective*, Promotors: Prof. H.G. Barkema & Prof. D.L. van Knippenberg, EPS-2014-331-S&E, http://repub.eur.nl/pub/77209

Almeida e Santos Nogueira, R.J. de, *Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty*, Promotors: Prof. U. Kaymak & Prof. J.M.C. Sousa, EPS-2014-310-LIS, http://repub.eur.nl/pub/51560

Bannouh, K., *Measuring and Forecasting Financial Market Volatility using High-frequency Data*, Promotor: Prof. D.J.C. van Dijk, EPS-2013-273-F&A, http://repub.eur.nl/pub/38240

Ben-Menahem, S.M., *Strategic Timing and Proactiveness of Organizations*, Promotors: Prof. H.W. Volberda & Prof. F.A.J. van den Bosch, EPS-2013-278-S&E, http://repub.eur.nl/pub/39128

Benning, T.M., *A Consumer Perspective on Flexibility in Health Care: Priority Access Pricing and Customized Care*, Promotor: Prof. B.G.C. Dellaert, EPS-2011-241-MKT, http://repub.eur.nl/pub/23670

Benschop, N, *Biases in Project Escalation: Names, frames & construal levels*, Promotors: Prof. K.I.M. Rhode, Prof. H.R. Commandeur, Prof. M.Keil & Dr A.L.P. Nuijten, EPS-2015-375-S&E, hdl.handle.net/1765/79408

Berg, W.E. van den, *Understanding Salesforce Behavior using Genetic Association Studies*, Promotor: Prof. W.J.M.I. Verbeke, EPS-2014-311-MKT, http://repub.eur.nl/pub/51440

Betancourt, N.E., *Typical Atypicality: Formal and Informal Institutional Conformity, Deviance, and Dynamics*, Promotor: Prof. B. Krug, EPS-2012-262-ORG, http://repub.eur.nl/pub/32345

Bliek, R. de, *Empirical Studies on the Economic Impact of Trust*, Promotor: Prof. J. Veenman & Prof. Ph.H.B.F. Franses, EPS-2015-324-ORG, http://repub.eur.nl/pub/78159

Blitz, D.C., *Benchmarking Benchmarks*, Promotors: Prof. A.G.Z. Kemna & Prof. W.F.C. Verschoor, EPS-2011-225-F&A, http://repub.eur.nl/pub/22624

Boons, M., *Working Together Alone in the Online Crowd: The Effects of Social Motivationsand Individual Knowledge Backgrounds on the Participation and Performance of Members of Online Crowdsourcing Platforms*, Promotors: Prof. H.G. Barkema & Dr D.A. Stam, EPS-2014-306-S&E, http://repub.eur.nl/pub/50711

Brazys, J., *Aggregated Marcoeconomic News and Price Discovery*, Promotor: Prof. W.F.C. Verschoor, EPS-2015-351-F&A, http://repub.eur.nl/pub/78243

Burger, M.J., *Structure and Cooptition in Urban Networks*, Promotors: Prof. G.A. van der Knaap & Prof. H.R. Commandeur, EPS-2011-243-ORG, http://repub.eur.nl/pub/26178

Byington, E., *Exploring Coworker Relationships: Antecedents and Dimensions of Interpersonal Fit,Coworker Satisfaction, and Relational Models*, Promotor: Prof. D.L. van Knippenberg, EPS-2013-292-ORG, http://repub.eur.nl/pub/41508

Camacho, N.M., *Health and Marketing: Essays on Physician and Patient Decision-Making*, Promotor: Prof. S. Stremersch, EPS-2011-237-MKT, http://repub.eur.nl/pub/23604

Cancurtaran, P., *Essays on Accelerated Product Development*, Promotors: Prof. F. Langerak & Prof.G.H. van Bruggen, EPS-2014-317-MKT, http://repub.eur.nl/pub/76074

Caron, E.A.M., *Explanation of Exceptional Values in Multi-dimensional Business Databases*, Promotors: Prof. H.A.M. Daniels & Prof. G.W.J. Hendrikse, EPS-2013-296-LIS, http://repub.eur.nl/pub/50005

Carvalho, L. de, *Knowledge Locations in Cities: Emergence and Development Dynamics*, Promotor: Prof. L. Berg, EPS-2013-274-S&E, http://repub.eur.nl/pub/38449

Consiglio, I., *Others: Essays on Interpersonal and Consumer Behavior*, Promotor: Prof. S.M.J. van Osselaer, EPS-2016-366-MKT, http://repub.eur.nl/pub/79820

Cox, R.H.G.M., *To Own, To Finance, and To Insure - Residential Real Estate Revealed*, Promotor: Prof. D. Brounen, EPS-2013-290-F&A, http://repub.eur.nl/pub/40964

Deichmann, D., *Idea Management: Perspectives from Leadership, Learning, and Network Theory*, Promotor: Prof. J.C.M. van den Ende, EPS-2012-255-ORG, http://repub.eur.nl/pub/31174

Deng, W., *Social Capital and Diversification of Cooperatives*, Promotor: Prof. G.W.J. Hendrikse, EPS-2015-341-ORG, http://repub.eur.nl/pub/77449

Desmet, P.T.M., *In Money we Trust? Trust Repair and the Psychology of Financial Compensations*, Promotor: Prof. D. de Cremer, EPS-2011-232-ORG, http://repub.eur.nl/pub/23268

Dollevoet, T.A.B., *Delay Management and Dispatching in Railways*, Promotor: Prof. A.P.M. Wagelmans, EPS-2013-272-LIS, http://repub.eur.nl/pub/38241

Doorn, S. van, *Managing Entrepreneurial Orientation*, Promotors: Prof. J.J.P. Jansen, Prof. F.A.J. van den Bosch, & Prof. H.W. Volberda, EPS-2012-258-STR, http://repub.eur.nl/pub/32166

Douwens-Zonneveld, M.G., *Animal Spirits and Extreme Confidence: No Guts, No Glory?* Promotor: Prof. W.F.C. Verschoor, EPS-2012-257-F&A, http://repub.eur.nl/pub/31914

Duca, E., *The Impact of Investor Demand on Security Offerings*, Promotor: Prof. A. de Jong, EPS-2011-240-F&A, http://repub.eur.nl/pub/26041

Duyvesteyn, J.G. *Empirical Studies on Sovereign Fixed Income Markets*, Promotors: Prof. P.Verwijmeren & Prof. M.P.E. Martens, EPS-2015-361-F&A, hdl.handle.net/1765/79033

Duursema, H., *Strategic Leadership: Moving Beyond the Leader-Follower Dyad*, Promotor: Prof. R.J.M. van Tulder, EPS-2013-279-ORG, http://repub.eur.nl/pub/39129

Eck, N.J. van, *Methodological Advances in Bibliometric Mapping of Science*, Promotor: Prof. R. Dekker, EPS-2011-247-LIS, http://repub.eur.nl/pub/26509

Elemes, A, *Studies on Determinants and Consequences of Financial Reporting Quality*, Promotor: Prof. E.Peek, EPS-2015-354-F&A, http://hdl.handle.net/1765/79037

Ellen, S. ter, *Measurement, Dynamics, and Implications of Heterogeneous Beliefs in Financial Markets*, Promotor: Prof. W.F.C. Verschoor, EPS-2015-343-F&A, http://repub.eur.nl/pub/78191

Erlemann, C., *Gender and Leadership Aspiration: The Impact of the Organizational Environment*, Promotor: Prof. D.L. van Knippenberg, EPS-2016-376-ORG, http://repub.eur.nl/pub/79409

Eskenazi, P.I., *The Accountable Animal*, Promotor: Prof. F.G.H. Hartmann, EPS-2015-355-F&A, http://repub.eur.nl/pub/78300

Essen, M. van, *An Institution-Based View of Ownership*, Promotors: Prof. J. van Oosterhout & Prof. G.M.H. Mertens, EPS-2011-226-ORG, http://repub.eur.nl/pub/22643

Evangelidis, I., *Preference Construction under Prominence*, Promotor: Prof. S.M.J. van Osselaer, EPS-2015-340-MKT, http://repub.eur.nl/pub/78202

Faber, N., *Structuring Warehouse Management*, Promotors: Prof. MB.M. de Koster & Prof. Ale Smidts, EPS-2015-336-LIS, http://repub.eur.nl/pub/78603

Fernald, K., *The Waves of Biotechnological Innovation in Medicine: Interfirm Cooperation Effects and a Venture Capital Perspective*, Promotors: Prof. E.Claassen, Prof. H.P.G. Pennings & Prof. H.R. Commandeur, EPS-2015-371-S&E, http://hdl.handle.net/1765/79120

Fourne, S.P., *Managing Organizational Tensions: A Multi-Level Perspective on Exploration, Exploitation and Ambidexterity*, Promotors: Prof. J.J.P. Jansen & Prof. S.J. Magala, EPS-2014-318-S&E, http://repub.eur.nl/pub/76075

Gharehgozli, A.H., *Developing New Methods for Efficient Container Stacking Operations*, Promotor: Prof. M.B.M. de Koster, EPS-2012-269-LIS, http://repub.eur.nl/pub/37779

Gils, S. van, *Morality in Interactions: On the Display of Moral Behavior by Leaders and Employees*, Promotor: Prof. D.L. van Knippenberg, EPS-2012-270-ORG, http://repub.eur.nl/pub/38027

Ginkel-Bieshaar, M.N.G. van, *The Impact of Abstract versus Concrete Product Communications on Consumer Decision-making Processes*, Promotor: Prof. B.G.C. Dellaert, EPS-2012-256-MKT, http://repub.eur.nl/pub/31913

Gkougkousi, X., *Empirical Studies in Financial Accounting*, Promotors: Prof. G.M.H. Mertens & Prof. E. Peek, EPS-2012-264-F&A, http://repub.eur.nl/pub/37170

Glorie, K.M., *Clearing Barter Exchange Markets: Kidney Exchange and Beyond*, Promotors: Prof. A.P.M. Wagelmans & Prof. J.J. van de Klundert, EPS-2014-329-LIS, http://repub.eur.nl/pub/77183

Hekimoglu, M., *Spare Parts Management of Aging Capital Products*, Promotor: Prof. R. Dekker, EPS-2015-368-LIS, http://hdl.handle.net/1765/79092

Heij, C.V., *Innovating beyond Technology. Studies on how management innovation, co-creation and business model innovation contribute to firm's (innovation) performance,* Promotors: Prof. F.A.J. van den Bosch & Prof. H.W. Volberda, EPS-2012-370-STR, http://repub.eur.nl/pub/78651

Heyde Fernandes, D. von der, *The Functions and Dysfunctions of Reminders*, Promotor: Prof. S.M.J. van Osselaer, EPS-2013-295-MKT, http://repub.eur.nl/pub/41514

Heyden, M.L.M., *Essays on Upper Echelons & Strategic Renewal: A Multilevel Contingency Approach*, Promotors: Prof. F.A.J. van den Bosch & Prof. H.W. Volberda, EPS-2012-259-STR, http://repub.eur.nl/pub/32167

Hoever, I.J., *Diversity and Creativity*, Promotor: Prof. D.L. van Knippenberg, EPS-2012-267-ORG, http://repub.eur.nl/pub/37392

Hogenboom, A.C., *Sentiment Analysis of Text Guided by Semantics and Structure*, Promotors: Prof. U. Kaymak & Prof. F.M.G. de Jong, EPS-2015-369-LIS, http://hdl.handle.net/1765/79034

Hogenboom, F.P., *Automated Detection of Financial Events in News Text*, Promotors: Prof. U. Kaymak & Prof. F.M.G. de Jong, EPS-2014-326-LIS, http://repub.eur.nl/pub/77237

Hollen, R.M.A., *Exploratory Studies into Strategies to Enhance Innovation-Driven International Competitiveness in a Port Context: Toward Ambidextrous Ports,* Promotors: Prof. F.A.J. Van Den Bosch & Prof. H.W.Volberda, EPS-2015-372-S&E, hdl.handle.net/1765/78881

Hoogendoorn, B., *Social Entrepreneurship in the Modern Economy: Warm Glow, Cold Feet,* Promotors: Prof. H.P.G. Pennings & Prof. A.R. Thurik, EPS-2011-246-STR, http://repub.eur.nl/pub/26447

Hoogervorst, N., *On The Psychology of Displaying Ethical Leadership: A Behavioral Ethics Approach*, Promotors: Prof. D. de Cremer & Dr M. van Dijke, EPS-2011-244-ORG, http://repub.eur.nl/pub/26228

Hout, D.H. van, *Measuring Meaningful Differences: Sensory Testing Based Decision Making in an Industrial Context; Applications of Signal Detection Theory and Thurstonian Modelling*, Promotors: Prof. P.J.F. Groenen & Prof. G.B. Dijksterhuis, EPS-2014-304-MKT, http://repub.eur.nl/pub/50387

Houwelingen, G.G. van, *Something To Rely On*, Promotors: Prof. D. de Cremer & Prof. M.H. van Dijke, EPS-2014-335-ORG, http://repub.eur.nl/pub/77320

Hurk, E. van der, *Passengers, Information, and Disruptions*, Promotors: Prof. L.G. Kroon & Prof. P.H.M. Vervest, EPS-2015-345-LIS, http://repub.eur.nl/pub/78275

Hytonen, K.A., *Context Effects in Valuation, Judgment and Choice: A Neuroscientific Approach*, Promotor: Prof. A. Smidts, EPS-2011-252-MKT, http://repub.eur.nl/pub/30668

Iseger, P. den, *Fourier and Laplace Transform Inversion with Applications in Finance*, Promotor: Prof. R. Dekker, EPS-2014-322-LIS, http://repub.eur.nl/pub/76954

Jaarsveld, W.L. van, *Maintenance Centered Service Parts Inventory Control*, Promotor: Prof. R. Dekker, EPS-2013-288-LIS, http://repub.eur.nl/pub/39933

Jalil, M.N., *Customer Information Driven After Sales Service Management: Lessons from Spare Parts Logistics*, Promotor: Prof. L.G. Kroon, EPS-2011-222-LIS, http://repub.eur.nl/pub/22156

Kappe, E.R., *The Effectiveness of Pharmaceutical Marketing*, Promotor: Prof. S. Stremersch, EPS-2011-239-MKT, http://repub.eur.nl/pub/23610

Karreman, B., *Financial Services and Emerging Markets*, Promotors: Prof. G.A. van der Knaap & Prof. H.P.G. Pennings, EPS-2011-223-ORG, http://repub.eur.nl/pub/22280

Khanagha, S., *Dynamic Capabilities for Managing Emerging Technologies*, Promotor: Prof. H.W. Volberda, EPS-2014-339-S&E, http://repub.eur.nl/pub/77319

Kil, J., *Acquisitions Through a Behavioral and Real Options Lens*, Promotor: Prof. H.T.J. Smit, EPS-2013-298-F&A, http://repub.eur.nl/pub/50142

Klooster, E. van 't, *Travel to Learn: the Influence of Cultural Distance on Competence Development in Educational Travel*, Promotors: Prof. F.M. Go & Prof. P.J. van Baalen, EPS-2014-312-MKT, http://repub.eur.nl/pub/51462

Koendjbiharie, S.R., *The Information-Based View on Business Network Performance: Revealing the Performance of Interorganizational Networks*, Promotors: Prof. H.W.G.M. van Heck & Prof. P.H.M. Vervest, EPS-2014-315-LIS, http://repub.eur.nl/pub/51751

Koning, M., *The Financial Reporting Environment: The Role of the Media, Regulators and Auditors*, Promotors: Prof. G.M.H. Mertens & Prof. P.G.J. Roosenboom, EPS-2014-330-F&A, http://repub.eur.nl/pub/77154

Konter, D.J., *Crossing Borders with HRM: An Inquiry of the Influence of Contextual Differences in the Adoption and Effectiveness of HRM*, Promotors: Prof. J. Paauwe & Dr L.H. Hoeksema, EPS-2014-305-ORG, http://repub.eur.nl/pub/50388

Korkmaz, E., *Bridging Models and Business: Understanding Heterogeneity in Hidden Drivers of Customer Purchase Behavior*, Promotors: Prof. S.L. van de Velde & Prof. D. Fok, EPS-2014-316-LIS, http://repub.eur.nl/pub/76008

Kroezen, J.J., *The Renewal of Mature Industries: An Examination of the Revival of the Dutch Beer Brewing Industry*, Promotor: Prof. P.P.M.A.R. Heugens, EPS-2014-333-S&E, http://repub.eur.nl/pub/77042

Kysucky, V., *Access to Finance in a Cros-Country Context*, Promotor: Prof.dr. L. Norden, EPS-2015-350-F&A, http://repub.eur.nl/pub/78225

Lam, K.Y., *Reliability and Rankings*, Promotor: Prof. Ph.H.B.F. Franses, EPS-2011-230-MKT, http://repub.eur.nl/pub/22977

Lander, M.W., *Profits or Professionalism? On Designing Professional Service Firms*, Promotors: Prof. J. van Oosterhout & Prof. P.P.M.A.R. Heugens, EPS-2012-253-ORG, http://repub.eur.nl/pub/30682

Langhe, B. de, *Contingencies: Learning Numerical and Emotional Associations in an Uncertain World*, Promotors: Prof. B. Wierenga & Prof. S.M.J. van Osselaer, EPS-2011-236-MKT, http://repub.eur.nl/pub/23504

Lee, C.I.S.G, *Big Data in Management Research: Exploring New Avenues*, Promotors: Prof. S.J. Magala & Dr W.A. Felps, EPS-2016-365-ORG, http://repub.eur.nl/pub/79818

Legault-Tremblay, P.O., Corporate Governance During Market Transition: Heterogeneous responses to Institution Tensions in China, Promotor: Prof. B. Krug, EPS-2015-362-ORG, http://repub.eur.nl/pub/78649

Lenoir, A.S. *Are You Talking to Me? Addressing Consumers in a Globalised World,* Promotors: Prof. S. Puntoni & Prof. S.M.J. van Osselaer, EPS-2015-363-MKT, http://hdl.handle.net/1765/79036

Leunissen, J.M., *All Apologies: On the Willingness of Perpetrators to Apologize*, Promotors: Prof. D. de Cremer & Dr. M. van Dijke, EPS-2014-301-ORG, http://repub.eur.nl/pub/50318

Li, D., *Supply Chain Contracting for After-sales Service and Product Support*, Promotor: Prof. M.B.M. de Koster, EPS-2015-347-LIS, http://repub.eur.nl/pub/78526

Li, Z., *Irrationality: What, Why and How*, Promotor(s): Prof. H. Bleichrodt, Prof. P.P. Wakker, & Prof. K.I.M. Rohde, EPS-2014-338-MKT, http://repub.eur.nl/pub/77205

Liang, Q.X., *Governance, CEO Identity, and Quality Provision of Farmer Cooperatives*, Promotor: Prof. G.W.J. Hendrikse, EPS-2013-281-ORG, http://repub.eur.nl/pub/39253

Liket, K., *Why 'Doing Good' is not Good Enough: Essays on Social Impact Measurement*, Promotors: Prof. H.R. Commandeur & Dr K.E.H. Maas, EPS-2014-307-STR, http://repub.eur.nl/pub/51130

Loos, M.J.H.M. van der, *Molecular Genetics and Hormones: New Frontiers in Entrepreneurship Research*, Promotors: Prof. A.R. Thurik, Prof. P.J.F. Groenen, & Prof. A. Hofman, EPS-2013-287-S&E, http://repub.eur.nl/pub/40081

Lovric, M., *Behavioral Finance and Agent-Based Artificial Markets*, Promotors: Prof. J. Spronk & Prof. U. Kaymak, EPS-2011-229-F&A, http://repub.eur.nl/pub/22814

Lu, Y., *Data-Driven Decision Making in Auction Markets*, Promotors: Prof. H.W.G.M. van Heck & Prof. W. Ketter, EPS-2014-314-LIS, http://repub.eur.nl/pub/51543

Ma, Y., *The Use of Advanced Transportation Monitoring Data for Official Statistics*, Promotors: Prof. L.G. Kroon and Dr Jan van Dalen, EPS-2016-391-LIS, hdl.handle.net/1765/80174

Manders, B., *Implementation and Impact of ISO 9001*, Promotor: Prof. K. Blind, EPS-2014-337-LIS, http://repub.eur.nl/pub/77412

Markwat, T.D., *Extreme Dependence in Asset Markets Around the Globe*, Promotor: Prof. D.J.C. van Dijk, EPS-2011-227-F&A, http://repub.eur.nl/pub/22744

Mees, H., *Changing Fortunes: How China's Boom Caused the Financial Crisis*, Promotor: Prof. Ph.H.B.F. Franses, EPS-2012-266-MKT, http://repub.eur.nl/pub/34930

Mell, J.N., *Connecting Minds: On The Role of Metaknowledge in Knowledge Coordination*, Promotor: Prof. D.L. van Knippenberg, EPS-2015-359-ORG, http://hdl.handle.net/1765/78951

Meuer, J., *Configurations of Inter-firm Relations in Management Innovation: A Study in China's Biopharmaceutical Industry*, Promotor: Prof. B. Krug, EPS-2011-228-ORG, http://repub.eur.nl/pub/22745

Micheli, M.R., *Business Model Innovation: A Journey across Managers' Attention and Inter-Organizational Networks*, Promotor: Prof. J.J.P. Jansen, EPS-2015-344-S&E, http://repub.eur.nl/pub/78241

Mihalache, O.R., *Stimulating Firm Innovativeness: Probing the Interrelations between Managerial and Organizational Determinants*, Promotors: Prof. J.J.P. Jansen, Prof. F.A.J. van den Bosch, & Prof. H.W. Volberda, EPS-2012-260-S&E, http://repub.eur.nl/pub/32343

Milea, V., *News Analytics for Financial Decision Support*, Promotor: Prof. U. Kaymak, EPS-2013-275-LIS, http://repub.eur.nl/pub/38673

Naumovska, I., *Socially Situated Financial Markets: A Neo-Behavioral Perspective on Firms, Investors and Practices*, Promotors: Prof. P.P.M.A.R. Heugens & Prof. A. de Jong, EPS-2014-319-S&E, http://repub.eur.nl/pub/76084

Nielsen, L.K., *Rolling Stock Rescheduling in Passenger Railways: Applications in short term planning and in disruption management*, Promotor: Prof. L.G. Kroon, EPS-2011-224-LIS, http://repub.eur.nl/pub/22444

Nuijten, A.L.P., *Deaf Effect for Risk Warnings: A Causal Examination applied to Information Systems Projects*, Promotors: Prof. G.J. van der Pijl, Prof. H.R. Commandeur & Prof. M. Keil, EPS-2012-263-S&E, http://repub.eur.nl/pub/34928

Oord, J.A. van, *Essays on Momentum Strategies in Finance*, Promotor: Prof. H.K. van Dijk, EPS-2016-380-F&A, hdl.handle.net/1765/80036

Osadchiy, S.E., *The Dynamics of Formal Organization: Essays on bureaucracy and formal rules*, Promotor: Prof. P.P.M.A.R. Heugens, EPS-2011-231-ORG, http://repub.eur.nl/pub/23250

Ozdemir, M.N., *Project-level Governance, Monetary Incentives, and Performance in Strategic R&D Alliances*, Promotor: Prof. J.C.M. van den Ende, EPS-2011-235-LIS, http://repub.eur.nl/pub/23550

Peers, Y., *Econometric Advances in Diffusion Models*, Promotor: Prof. Ph.H.B.F. Franses, EPS-2011-251-MKT, http://repub.eur.nl/pub/30586

Peters, M., *Machine Learning Algorithms for Smart Electricity Markets*, Promotor: Prof. W. Ketter, EPS-2014-332-LIS, http://repub.eur.nl/pub/77413

Porck, J., *No Team is an Island: An Integrative View of Strategic Consensus between Groups*, Promotors: Prof. P.J.F. Groenen & Prof. D.L. van Knippenberg, EPS-2013-299-ORG, http://repub.eur.nl/pub/50141

Porras Prado, M., *The Long and Short Side of Real Estate, Real Estate Stocks, and Equity*, Promotor: Prof. M.J.C.M. Verbeek, EPS-2012-254-F&A, http://repub.eur.nl/pub/30848

Poruthiyil, P.V., *Steering Through: How organizations negotiate permanent uncertainty and unresolvable choices*, Promotors: Prof. P.P.M.A.R. Heugens & Prof. S.J. Magala, EPS-2011-245-ORG, http://repub.eur.nl/pub/26392

Pourakbar, M., *End-of-Life Inventory Decisions of Service Parts*, Promotor: Prof. R. Dekker, EPS-2011-249-LIS, http://repub.eur.nl/pub/30584

Pronker, E.S., *Innovation Paradox in Vaccine Target Selection*, Promotors: Prof. H.J.H.M. Claassen & Prof. H.R. Commandeur, EPS-2013-282-S&E, http://repub.eur.nl/pub/39654

Protzner, S. *Mind the gap between demand and supply: A behavioral perspective on demand forecasting*, Promotors: Prof. S.L. van de Velde & Dr L. Rook, EPS-2015-364-LIS, *http*://repub.eur.nl/pub/79355

Pruijssers, J.K., *An Organizational Perspective on Auditor Conduct*, Promotors: Prof. J. van Oosterhout & Prof. P.P.M.A.R. Heugens, EPS-2015-342-S&E, http://repub.eur.nl/pub/78192

Retel Helmrich, M.J., *Green Lot-Sizing*, Promotor: Prof. A.P.M. Wagelmans, EPS-2013-291-LIS, http://repub.eur.nl/pub/41330

Rietdijk, W.J.R. *The Use of Cognitive Factors for Explaining Entrepreneurship*, Promotors: Prof. A.R. Thurik & Prof. I.H.A. Franken, EPS-2015-356-S&E, http://repub.eur.nl/pub/79817

Rietveld, N., *Essays on the Intersection of Economics and Biology*, Promotors: Prof. A.R. Thurik, Prof. Ph.D. Koellinger, Prof. P.J.F. Groenen, & Prof. A. Hofman, EPS-2014-320-S&E, http://repub.eur.nl/pub/76907

Rijsenbilt, J.A., *CEO Narcissism: Measurement and Impact*, Promotors: Prof. A.G.Z. Kemna & Prof. H.R. Commandeur, EPS-2011-238-STR, http://repub.eur.nl/pub/23554

Rösch, D. *Market Efficiency and Liquidity*, Promotor: Prof. M.A. van Dijk, EPS-2015-353-F&A, http://hdl.handle.net/1765/79121

Roza-van Vuren, M.W., *The Relationship between Offshoring Strategies and Firm Performance: Impact of innovation, absorptive capacity and firm size*, Promotors: Prof. H.W. Volberda & Prof. F.A.J. van den Bosch, EPS-2011-214-STR, http://repub.eur.nl/pub/22155

Rubbaniy, G., *Investment Behaviour of Institutional Investors*, Promotor: Prof. W.F.C. Verschoor, EPS-2013-284-F&A, http://repub.eur.nl/pub/40068

Schoonees, P. *Methods for Modelling Response Styles*, Promotor: Prof.dr P.J.F. Groenen, EPS-2015-348-MKT, http://repub.eur.nl/pub/79327

Schouten, M.E., *The Ups and Downs of Hierarchy: the causes and consequences of hierarchy struggles and positional loss*, Promotors; Prof. D.L. van Knippenberg & Dr L.L. Greer, EPS-2016-386-ORG, http://repub.eur.nl/pub/80059

Shahzad, K., *Credit Rating Agencies, Financial Regulations and the Capital Markets*, Promotor: Prof. G.M.H. Mertens, EPS-2013-283-F&A, http://repub.eur.nl/pub/39655

Sousa, M.J.C. de, *Servant Leadership to the Test: New Perspectives and Insights*, Promotors: Prof. D.L. van Knippenberg & Dr D. van Dierendonck, EPS-2014-313-ORG, http://repub.eur.nl/pub/51537

Spliet, R., *Vehicle Routing with Uncertain Demand*, Promotor: Prof. R. Dekker, EPS-2013-293-LIS, http://repub.eur.nl/pub/41513

Staadt, J.L., *Leading Public Housing Organisation in a Problematic Situation: A Critical Soft Systems Methodology Approach*, Promotor: Prof. S.J. Magala, EPS-2014-308-ORG, http://repub.eur.nl/pub/50712

Stallen, M., *Social Context Effects on Decision-Making: A Neurobiological Approach*, Promotor: Prof. A. Smidts, EPS-2013-285-MKT, http://repub.eur.nl/pub/39931

Tarakci, M., *Behavioral Strategy: Strategic Consensus, Power and Networks*, Promotor: Prof. D.L. van Knippenberg & Prof. P.J.F. Groenen, EPS-2013-280-ORG, http://repub.eur.nl/pub/39130

Teixeira de Vasconcelos, M., *Agency Costs, Firm Value, and Corporate Investment*, Promotor: Prof. P.G.J. Roosenboom, EPS-2012-265-F&A, http://repub.eur.nl/pub/37265

Troster, C., *Nationality Heterogeneity and Interpersonal Relationships at Work*, Promotor: Prof. D.L. van Knippenberg, EPS-2011-233-ORG, http://repub.eur.nl/pub/23298

Tsekouras, D., *No Pain No Gain: The Beneficial Role of Consumer Effort in Decision-Making*, Promotor: Prof. B.G.C. Dellaert, EPS-2012-268-MKT, http://repub.eur.nl/pub/37542

Tuijl, E. van, *Upgrading across Organisational and Geographical Configurations*, Promotor: Prof. L. van den Berg, EPS-2015-349-S&E, http://repub.eur.nl/pub/78224

Tuncdogan, A., *Decision Making and Behavioral Strategy: The Role of Regulatory Focus in Corporate Innovation Processes*, Promotors: Prof. F.A.J. van den Bosch, Prof. H.W. Volberda, & Prof. T.J.M. Mom, EPS-2014-334-S&E, http://repub.eur.nl/pub/76978

Uijl, S. den, *The Emergence of De-facto Standards*, Promotor: Prof. K. Blind, EPS-2014-328-LIS, http://repub.eur.nl/pub/77382

Vagias, D., *Liquidity, Investors and International Capital Markets*, Promotor: Prof. M.A. van Dijk, EPS-2013-294-F&A, http://repub.eur.nl/pub/41511

Veelenturf, L.P., *Disruption Management in Passenger Railways: Models for Timetable, Rolling Stock and Crew Rescheduling*, Promotor: Prof. L.G. Kroon, EPS-2014-327-LIS, http://repub.eur.nl/pub/77155

Venus, M., *Demystifying Visionary Leadership: In search of the essence of effective vision communication*, Promotor: Prof. D.L. van Knippenberg, EPS-2013-289-ORG, http://repub.eur.nl/pub/40079

Vermeer, W., *Propagation in Networks:The impact of information processing at the actor level on system-wide propagation dynamics,* Promotor: Prof. P.H.M.Vervest, EPS-2015-373-LIS, http://repub.eur.nl/pub/79325

Versluis, I., *Prevention of the Portion Size Effect*, Promotors: Prof. Ph.H.B.F. Franses & Dr E.K. Papies, EPS-2016-382-MKT, http://repub.eur.nl/pub/79880

Visser, V.A., *Leader Affect and Leadership Effectiveness: How leader affective displays influence follower outcomes*, Promotor: Prof. D.L. van Knippenberg, EPS-2013-286-ORG, http://repub.eur.nl/pub/40076

Vlam, A.J., *Customer First? The Relationship between Advisors and Consumers of Financial Products*, Promotor: Prof. Ph.H.B.F. Franses, EPS-2011-250-MKT, http://repub.eur.nl/pub/30585

Vries, J. de, *Behavioral Operations in Logistics,* Promotors: Prof. M.B.M de Koster & Prof. D.A. Stam, EPS-2015-374-LIS, http://repub.eur.nl/pub/79705

Waltman, L., *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*, Promotors: Prof. R. Dekker & Prof. U. Kaymak, EPS-2011-248-LIS, http://repub.eur.nl/pub/26564

Wang, T., *Essays in Banking and Corporate Finance*, Promotors: Prof. L. Norden & Prof. P.G.J. Roosenboom, EPS-2015-352-F&A, http://repub.eur.nl/pub/78301

Wang, Y., *Information Content of Mutual Fund Portfolio Disclosure*, Promotor: Prof. M.J.C.M. Verbeek, EPS-2011-242-F&A, http://repub.eur.nl/pub/26066

Wang, Y., *Corporate Reputation Management: Reaching Out to Financial Stakeholders*, Promotor: Prof. C.B.M. van Riel, EPS-2013-271-ORG, http://repub.eur.nl/pub/38675

Weenen, T.C., *On the Origin and Development of the Medical Nutrition Industry*, Promotors: Prof. H.R. Commandeur & Prof. H.J.H.M. Claassen, EPS-2014-309-S&E, http://repub.eur.nl/pub/51134

Wolfswinkel, M., *Corporate Governance, Firm Risk and Shareholder Value*, Promotor: Prof. A. de Jong, EPS-2013-277-F&A, http://repub.eur.nl/pub/39127

Yang, S., *Information Aggregation Efficiency of Prediction Markets*, Promotor: Prof. H.W.G.M. van Heck, EPS-2014-323-LIS, http://repub.eur.nl/pub/77184

Zaerpour, N., *Efficient Management of Compact Storage Systems*, Promotor: Prof. M.B.M. de Koster, EPS-2013-276-LIS, http://repub.eur.nl/pub/38766

Zhang, D., *Essays in Executive Compensation*, Promotor: Prof. I. Dittmann, EPS-2012-261-F&A, http://repub.eur.nl/pub/32344

Zwan, P.W. van der, *The Entrepreneurial Process: An International Analysis of Entry and Exit*, Promotors: Prof. A.R. Thurik & Prof. P.J.F. Groenen, EPS-2011-234-ORG, http://repub.eur.nl/pub/23422

How to deal with a disruption is a question railway companies face on a daily basis. This thesis focusses on the subject how to handle a disruption such that the passenger serviec is upheld as much as possible. The current mathematical models for disruption management can not yet be applied in practice, because several important practical considerations are not taken into account. In this thesis several models are presented which take important practical details into account: (1) creating a macroscopic global feasible solution for all three resource schedules, instead of focussing on one individual resource schedule. (2) Scheduled maintenance appointments required by certain rolling stock units are included while rescheduling. (3) Dead-heading trips to transfer rolling stock units from stations with a surplus of inventory to stations with a shortage of inventory. (4) Adjusted passenger demand, the passenger demand is not static, but depends on the capacity appointed to the previous trips. Finally, (5) checking whether a rolling stock circulation is feasible with respect to the available depot tracks (the shunting yard) within a station. We make use of different techniques to solve the models, for instance, mixed integer linear programming, column generation, constraint programming, and heuristic models are used in this thesis. The results demonstrate that these five practical considerations can be taken into account in the disruption management models.

ERiM

The Erasmus Research Institute of Management (ERIM) is the Research School (Onderzoek school) in the field of management of the Erasmus University Rotterdam. The founding participants of ERIM are the Rotterdam School of Management (RSM), and the Erasmus School of Economics (ESE). ERIM was founded in 1999 and is officially accredited by the Royal Netherlands Academy of Arts and Sciences (KNAW). The research under taken by ERIM is focused on the management of the firm in its environment, its intra- and interfirm relations, and its business processes in their interdependent connections.

The objective of ERIM is to carry out first rate research in management, and to offer an advanced doctoral programme in Research in Management. Within ERIM, over three hundred senior researchers and PhD candidates are active in the different research programmes. From a variety of academic backgrounds and expertises, the ERIM community is united in striving for excellence and working at the fore front of creating new business knowledge.

ERiM

ERIM PhD Series
Research in Management

Erasmus University Rotterdam (EUR)
Erasmus Research Institute of Management
Mandeville (T) Building
Burgemeester Oudlaan 50
3062 PA Rotterdam, The Netherlands

P.O. Box 1738
3000 DR Rotterdam, The Netherlands
T +31 10 408 1182
E info@erim.eur.nl
W www.erim.eur.nl