L.P. VEELENTURF

Disruption Management in Passenger Railways

Models for Timetable, Rolling Stock and Crew Rescheduling

Disruption Management in Passenger Railways

Models for Timetable, Rolling Stock and Crew Rescheduling

Disruption Management in Passenger Railways

Models for Timetable, Rolling Stock and Crew Rescheduling

Bijsturing van reizigerstreinen

Modellen voor het herplannen van de dienstregeling en de materieel- en personeelsdiensten.

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

Friday 21 November 2014 at 11:30 hrs

by

LUCAS PETRUS VEELENTURF
born in Woerden, the Netherlands.

ERASMUS UNIVERSITEIT ROTTERDAM

Doctoral Committee

Promotor: Prof.dr. L.G. Kroon

Other members: Dr. D. Huisman
Prof.dr. P. Toth
Prof.dr. R.A. Zuidwijk

Copromotor: Dr. G. Maróti

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, 327
ERIM reference number: EPS-2014-327-LIS
ISBN 978-90-5892-378-3

Design: B&T Ontwerp en advies www.b-en-t.nl

Cover design: Tosca Menten

This publication (cover and interior) is printed by haveka.nl on recycled paper, Revive®.
The ink used is produced from renewable resources and alcohol free fountain solution.
Certifications for the paper and the printing production process: Recycle, EU Flower, FSC, ISO14001.
More info: http://www.haveka.nl/greening

Acknowledgments

The beginning

During the second year of my Bachelor programme at Erasmus University, I visited the "Landelijke Econometristen Dag", an event for students in Econometrics and Operations Research. At the booth of Netherlands Railways (NS), I met Dennis Huisman and Leo Kroon.

Just a short time before, NS had introduced a new timetable in which the number of trains that stopped in my hometown was reduced significantly. So I took my chance and tried to convince Dennis and Leo that the old schedule was better (for me). Nowadays I know that I am not the only one who has a suggestion for a better timetable and that it isn't as easy as it looks.

Later that year, I came in contact with Dennis again for a final assignment of the ERIM Honours Program. I did research to determine how trains should be assigned to platforms. This was the start of my long research track in railway (re-)scheduling problems and my collaboration with NS, and with Dennis, Leo and Gábor in particular: for my Bachelor thesis, Master thesis and PhD thesis.

The academic support

While I am writing this, my time at Erasmus University is coming to an end by finishing this PhD thesis. I want to use this opportunity to thank the people who have supported me during my academic career at Erasmus University.

This phase in my career started after a phone call from Jan Brinkhuis during the second year of my Bachelor studies. He asked (i.e. had to convince) me to become a student assistant at Erasmus School of Economics. Nowadays I cannot believe that I had some doubts in the beginning. I really enjoy teaching and I am very grateful that Jan Brinkhuis made this call.

I appreciate all the support I got to develop myself further as a teacher from Bas, Cock, Dick, Erwin, Martijn, Martyn, Roelof, RSM programme management, and several student assistants in later teaching collaborations.

Dennis was the one who got me interested in railway research. I have learned a lot from him during our collaborations in several projects and papers. I would like to thank him for all his support and guidance during my different study phases at Erasmus University.

It was also Dennis who introduced me to the Process quality and Innovation department of NS when I had to write my Bachelor thesis. It kicked off a long, at the moment already 7 years, collaboration with this department. I want to thank all my colleagues within this department for the constructive working environment, for the insights in the processes at NS and for the laughs during the social activities.

Leo was one of my colleagues at NS and he introduced me to Jo van Nunen of Rotterdam School of Management (RSM) to discuss the possibility of becoming a PhD candidate. I was surprised by the warm welcome of Jo and it became clear that this was the way to go. Unfortunately, our collaboration sadly came to an end one year after my start; he is still greatly missed.

At that time Gábor Maróti stepped in to strengthen my supervisory team. Gábor is someone who is available day and night for all kinds of questions. He gave me guidance and had insightful comments, and he was also patient when dealing with my programming ”style”. I am grateful for all the support I got from him.

I was a PhD student at “department 1” (also known as “departyment 1”) of RSM. Within the department of Decision and Information Sciences, there was a great atmosphere. I want to thank all my colleagues within this department. It was the best place for my PhD journey. I especially want to mention some of my fellow PhD students.

Sarita: there are many good reasons why you are called the “mum” of the PhDs. Wouter: we had the best room of the department. Xiao: thanks for all the smiles and dances you brought me. Nick: you were the creative brain of the department. Paul and Clint: thanks for being my paranymphs and for supporting me in this last stage. Paul: I hope I can be your taxi driver many more times (if my business partner Clint allows it). Joris, Evelien, Irina, Konstantina, Panos and all other PhD students: we had wonderful times, and I wish you all the best.

As a department chair, Peter van Baalen gave me opportunities to reach my full potential within the department. He made me part of the daily board of the department, gave me additional training opportunities and believed in me to take over the full responsibility of a bachelor 1 course. I really appreciated these opportunities.

My department at RSM was not the only organisation that supported me. The research schools ERIM and TRAIL organised conferences, courses, social activities and financial support for my development. I stayed in touch with the Econometric Institute (Adriana, Albert, Dennis, Ilse, Judith, Kristiaan, Lanah, Mathijn, Remy, Rommert, Twan, Wilco and Willem), and they always made me feel very welcome in the H-building. I was also warmly (let’s for-

get about all the snow) welcomed by Paolo Toth, Valentina Cacchiani and Martin Kidd when I went to the university of Bologna for a 3 month research visit; the visit to Bologna was a great experience.

The PhD thesis is now finished. This was not possible without the support of my co-authors: Albert, Daniel, Dennis, Gábor, Leo, Martin, Paolo and Valentina. Now it is up to the committee members Ingo Hansen, Dennis Huisman, Johann Hurink, Leo Kroon, Gábor Maróti, Paolo Toth, Albert Wagelmans, Tom van Woensel and Rob Zuidwijk to have a final discussion about the merits of this thesis. I want to thank them for their willingness and enthusiasm to participate in the public defense.

This whole journey would not have been possible without Leo. I think nobody could wish himself a better supervisor. You were always there for me. We have collaborated in research and in teaching. You have supported me all the time in developing myself and in preparing me for the next steps in my career. You have made me a real academic. Thank you!

The home front

Last but not least, I want to show my appreciation for the support I got from my family. My grandfather (as a former traffic controller at NS) was a big fan of me doing research in railway disruption management. My parents were always very proud and supportive of me. Yet I have to give the most credits to my wife Mariëlle. She accepted the academic work pressure. Our time together was sometimes very scarce, but she always supported me. You are the most important "co-author" of the story of my life

Woerden, September 2014
Lucas Veelenturf

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Railways: A transportation mode in which trains run on dedicated tracks to transport passengers or freight from one location to another. It looks that simple. One needs a track, a physical train (also called *rolling stock*) and a driver, and ahead it goes. However, if a railway operator has promised to serve several thousands of trains a day in its timetable, it becomes more complex. It then needs thousands of rolling stock units and thousands of drivers. To be more efficient, the operator does not hire for each train service another driver and does not buy the same amount of rolling stock as the number of train services it operates. Rolling stock can be used sequentially for different services, and the same is true for drivers.

If a crew member has to operate a certain service between two locations, it is called a *task*. For rolling stock units a task means that the rolling stock is used for a certain service between two stations. For each crew member and rolling stock unit there is a list of sequential tasks they have to perform on one day, which is called a *duty*. Over the years railway operators have made the duties for rolling stock and crew as efficient as possible to reduce hiring or purchasing costs.

A disadvantage of these efficient schedules is that, if something goes wrong, the effect on the railway system can be severe. For example, if there is a separate rolling stock unit for each train service, a broken rolling stock unit requires the operator to find rolling stock for just one service. However if the broken rolling stock unit was assigned to multiple train services in its duty, for all these services new rolling stock must be found. If there are train services for which no rolling stock can be found, these services need to be cancelled. Cancelling a train service is the last option for an operator since it will lead to large delays for passengers. The passengers of the cancelled train service have to find another route to their destination and the train services which are not cancelled become more crowded.

Another example in which the efficient schedules can lead to problems is the following. Suppose there is a crew member who first has to operate a train from *Woerden* to *Rotterdam*, then his next task is to operate a train from *Rotterdam* to *Woerden* and the crew member ends his duty by driving a train from *Woerden* to *Eindhoven*. During operations, after the crew member has arrived in *Rotterdam*, the tracks between *Rotterdam* and *Woerden* are taken out of service due to a broken overhead line. The dispatchers of the railway operator then have to solve several problems. The dispatchers have to check whether they could get the crew member to *Woerden* in time such that he could still perform the last task in his duty. If this is not possible, the dispatchers have to ensure that the crew member can still reach his final destination (*Eindhoven*) and the dispatchers have to find another crew member to operate the train between *Woerden* and *Eindhoven* in time. This example considers just one train between *Rotterdam* and *Woerden*, but, as one can imagine, if multiple trains run between *Rotterdam* and *Woerden*, multiple duties of crew members (and rolling stock units) are affected and the dispatchers of the railway operator have a hard time to find a solution for all these affected duties.

1.2 Disruption Management

Many cases exist where irregularities disturb the operations of trains. *Disruptions* are in this thesis defined as unexpected events which require that several train services need to be cancelled due to unavailability of infrastructure, rolling stock or crew. Examples include a blockage of certain tracks due to broken overhead lines, an accident with other traffic or a malfunctioning rolling stock unit. If a disruption occurs, the timetable, rolling stock schedule and crew schedule become infeasible. Since disruptions occur unexpectedly, the operator has to make these schedules feasible again within a real-time environment. This is called *disruption management*.

One goal is to make the schedules feasible again. However, it is even more important that the passengers face as little inconvenience as possible. Therefore, the focus should be to minimize the number of train services which need to be cancelled by lack of rolling stock or crew. Furthermore, the capacity of the rolling stock assigned to a train service should match the demand for that service. This means that for rolling stock rescheduling also predictions about the demand should be available.

The disruption management process consists of finding a new timetable, finding rolling stock for every train service in the new timetable, and finding crew to operate all trains in the new timetable. The best solution to handle a disruption can be found by considering these three schedules in an integrated way together with the passenger behavior. However, handling one schedule at a time is already computationally challenging. Therefore, most research in railway disruption management focuses on rescheduling one schedule at the time. An overview of existing literature in recovery models and algorithms for railway disruption management is given by Cacchiani et al. (2014).

This thesis focuses on rescheduling one or at most two railway schedules at the time. The research aims to develop tools for dispatchers of passenger railway operators which help them in finding new schedules in case of large-scale disruptions. The approaches aim at reducing the inconvenience that passengers experience from the disruption by cancelling as few train services as possible.

If the rescheduling problems are solved separately, timetable rescheduling is most of the time seen as the first step in the disruption management process. If a new timetable is ready, the next step is to find rolling stock for each train service in the new timetable. The last step of the disruption management process deals with assigning crew to each train service in the new timetable while considering which type of rolling stock will be used for it. The structure of this thesis follows this procedure. First a timetable rescheduling approach is discussed in Chapter 2, then a combined timetable and rolling stock rescheduling approach is discussed in Chapter 3, and at last, Chapters 4 and 5 deal with crew rescheduling approaches.

1.3 Contributions

In this section the content of the research described in Chapters 2 to 5 is summarized in more detail. The performance of the developed approaches is tested on instances based on the operations of Netherlands Railways, which is the major railway operator in the Netherlands.

(Chapter 2): L.P. Veelenturf, M.P. Kidd, V. Cacchiani, L.G. Kroon, and P. Toth. A railway timetable rescheduling approach for handling large scale disruptions. Under review at Transportation Science

The research of Chapter 2 focuses on timetable rescheduling for passenger trains at a macroscopic (operator) level in a railway network. An integer programming model is formulated for solving the timetable rescheduling problem, which minimizes the number of cancelled and delayed trains while adhering to infrastructure and rolling stock capacity constraints. It is also possible to reroute trains to reduce the number of cancelled and delayed trains. All stages of the disruption management process (from the start of the disruption to the time the normal situation is restored) are taken into account.

(Chapter 3): L.P. Veelenturf, L.G. Kroon, and G. Maróti. Passenger Oriented Railway Disruption Management by Adapting Timetables and Rolling Stock Schedules. Under review at Annals of Operations Research

Chapter 3 describes a real-time disruption management approach which integrates the rescheduling of the rolling stock and the timetable by taking the changes in passenger demand into account. Due to a disruption, passengers will adapt their routes to their destinations. Therefore, the operator has to increase the capacity of trains for which it expects more demand than on a regular day. Furthermore, at locations with additional demand, the frequencies of trains serving that station could be increased. Within the disruption management approach, the timetable decisions are limited to additional stops of trains at stations they normally would not serve. Several variants of the approach are suggested, differing in how they determine which additional stops should be executed. Real-time rescheduling requires fast solutions. Therefore several variants of a heuristic approach are used.

(Chapter 4): L.P. Veelenturf, D. Potthoff, D. Huisman, and L.G. Kroon. Railway crew rescheduling with retiming. Transportation Research Part C: Emerging Technologies, 20:95-110, 2012.

In Chapter 4, the crew rescheduling problem with retiming is modelled and solved. This problem extends the crew rescheduling problem by the possibility to slightly delay the departure of some trains, such that more flexibility in the crew rescheduling process is obtained. The algorithm focuses on rescheduling the duties of train drivers. It is based on column generation techniques combined with Lagrangian heuristics. In order to prevent large increases in computation time, retiming is allowed for a limited number of train services.

(Chapter 5): L.P. Veelenturf, D. Potthoff, D. Huisman, L.G. Kroon, G. Maróti, and A.P.M. Wagelmans. A quasi-robust optimization approach for crew rescheduling. Transportation Science, forthcoming

Chapter 5 presents a novel approach for crew rescheduling by considering several scenarios for the duration of the disruption. The rescheduling problem is similar to a two-stage optimization problem. In the first stage, at the start of the disruption, the plan is rescheduled based on the optimistic scenario (i.e., assuming the shortest possible duration of the disruption), while taking into account the possibility that another scenario will be realized. A prescribed number of the rescheduled crew duties is required to be *recoverable*. The true duration of the disruption is revealed in the second stage. By the recoverability of the duties, it is expected that the first stage solution can easily be turned into a schedule that is feasible for the realized scenario for the duration of the disruption.

1.4 Outline

The remainder of this thesis consists of the papers described in Section 1.3. To make the chapters self-contained, we chose to include exact copies of the papers submitted or published. This means that there can be overlap in the introductions and definitions of the problems discussed in the chapters. In Chapter 6 we summarize the concluding remarks of the different papers.

Chapter 2

A railway timetable rescheduling approach for handling large scale disruptions

This chapter considers the paper (**Veelenturf** et al. (2014a)) which is under review at Transportation Science. 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. In 2013 a preliminary version of this paper has been granted a third place in the Student Paper Award Competition of the Railway Application Section of INFORMS

Co-authors: M.P. Kidd, V. Cacchiani, L.G. Kroon and P. Toth

2.1 Introduction

The occurrence of unexpected large-scale disruptions, such as the unavailability of railway track segments due to broken overhead wires or rolling stock breakdowns, causes train delays and train cancellations with a consequent reduction of the quality of service to the passengers. Therefore, it is crucial to recover from such situations as quickly as possible in order to reduce passenger dissatisfaction and to restore the service of the railway system.

Due to its complexity, the recovery problem is usually decomposed into phases that are solved in sequence. The main phases consist of timetable rescheduling, rolling stock rescheduling and crew rescheduling. Timetable rescheduling calls for determining a feasible timetable by applying reordering, retiming and rerouting of trains and even train cancellations. The derived timetable is input to the second phase, in which it may be necessary to determine a new rolling stock allocation, due to the changes applied in the previous phase. Similarly, the new timetable and rolling stock allocation are input to the last phase that aims at obtaining a feasible crew schedule. Obviously, a feedback loop is sometimes necessary if no feasible rolling stock or crew plan can be obtained, possibly requiring the cancellation of additional trains. Solving the three phases separately may lead to sub-optimal solutions. However, solving them all in an integrated way would lead to unacceptably long computing times, as we are facing real-time problems. An overview of current models for solving these three steps is presented in Cacchiani et al. (2014).

In this paper, we focus on the timetable rescheduling phase, thereby taking into account constraints from the rolling stock rescheduling phase in order to increase the probability of obtaining a feasible rolling stock schedule during the second phase. Constraints of the crew rescheduling phase are more difficult to include, since there are much more complicated rules about (meal) breaks and durations of crew duties. Therefore these are not considered in our model.

We study timetable rescheduling at a macroscopic level, i.e. with high level constraints disregarding detailed information on signals and routes inside stations or junctions. The reason is that we want to deal with a complex real-world railway network and, at the same time, to solve the problem in very short computing times. First, at a macroscopic level it has to be determined which trains can still run with the available infrastructure capacity. Thereafter, small conflicts at the signaling level should be detected and solved by slightly delaying some trains.

We consider large-scale disruptions related to blockages of one or more railway tracks between stations for a certain period of time (e.g. two hours). Indeed, disruptions of this kind are very hard to manage by railway operators and infrastructure managers, as they cause many changes in the system and decisions need to be taken very quickly. In addition, such disruptions occur on a daily basis and the support of an automated tool for solving them is highly desirable. Currently such disruptions are handled in practice by selecting the appropriate contingency plan from a large set of such plans.

The main contribution of this paper consists of proposing an Integer Linear Programming (ILP) formulation for the timetable rescheduling problem to deal with large-scale disruptions on a real-world railway network. The formulation takes into account constraints that allow to partially integrate the timetable rescheduling problem with the rolling stock rescheduling phase. In particular, we consider a railway network with a cyclic timetable, i.e. the schedule of the trains is repeated every given time period (for example every hour).

This approach generates a new timetable from the start of a disruption until the moment at which the timetable must be back again to the normal state. This means that the approach does not only make a new reduced cyclic timetable for the steady disrupted state, but it also produces a timetable to make the transition from the original timetable to this new reduced cyclic timetable and back to the original timetable after the disruption has ended.

A useful feature of the proposed model is that it takes into account the possibility of rerouting trains along alternative geographical paths in the network in order to reduce the number of trains that are cancelled or delayed.

The model is solved to optimality by a general purpose solver on a set of real-world instances of Netherlands Railways (the major railway operator in the Netherlands) in short computing times.

This paper is organized as follows. Section 2.2 presents an overview of related research. In Section 2.3, the problem is presented and in Section 2.4 an ILP formulation is given. Section 2.5 is devoted to the computational results based on instances of Netherlands Railways, and the conclusions are discussed in Section 2.6.

2.2 Literature overview

Many works study the Train Timetabling Problem during the planning phase, i.e., when an optimal timetable is derived for a set of trains in a time horizon of six months to one year. We refer the reader to the following recent surveys on this topic: Cacchiani and Toth (2012), Caprara et al. (2007), Caprara et al. (2011) and Lusby et al. (2011). In recent years, many studies have been developed dealing with real-time timetable rescheduling. The majority of them concern train rescheduling when relatively small disturbances affect a subset of trains, instead of large-scale disruptions as is the case in our paper. We refer to Cacchiani et al. (2014) for an overview of real-time rescheduling problems and solution approaches.

For example, the real-time traffic management system ROMA (Railway traffic Optimization by Means of Alternative graphs) is presented in D'Ariano et al. (2008a) and D'Ariano et al. (2007). ROMA considers the infrastructure at a detailed level and uses a branch-and-bound algorithm for sequencing train movements combined with a local search algorithm for rerouting trains. The experiments described in these papers concern the line between Utrecht and 's Hertogenbosch, and the congested areas around Utrecht Central Station and around Schiphol Amsterdam Airport in the Netherlands.

Extensions of ROMA are presented in Corman et al. (2009), Corman et al. (2010), Corman et al. (2012), and D'Ariano et al. (2008b), taking into account different objectives (minimization of train delays and preservation of train connections), or building flexible timetables that postpone certain decisions to the operational phase. These works consider a set of instances provided by Netherlands Railways.

Other microscopic approaches for small disturbances are presented in Boccia et al. (2013), Caimi et al. (2012), Lusby et al. (2013) and Mannino and Mascis (2009).

A macroscopic level of detail of the railway network to handle disturbances is considered in Acuna-Agost et al. (2011b), Acuna-Agost et al. (2011a), Dollevoet et al. (2012), Kecman et al. (2013), Schöbel (2009) and Törnquist and Persson (2007).

Only very few works deal with large-scale disruptions. In Albrecht et al. (2013), disruptions due to unexpected track maintenance extensions are considered, i.e. longer maintenance operations are required than were planned. A meta-heuristic is used to construct an integrated timetable which includes track maintenance, and an operational tool is used to generate a new feasible schedule for the disrupted system. A case study for a single track rail network in Queensland Australia is carried out.

In Brucker et al. (2002), train rescheduling is considered in the case of a partial track blockage due to construction works. A local search algorithm is presented with the goal of minimizing lateness. This algorithm is tested on real-life instances of the German railways.

In Corman et al. (2011), the authors consider the case of double track railway lines where some block sections of one track are unavailable. Centralized and distributed approaches are presented: in the centralized approach the entire rescheduling problem is solved, while in the distributed approach a coordinator sets constraints between areas and delegates the scheduling decisions to local schedulers. Computational experiments on a large railway network in the Netherlands show that both approaches face increasing difficulty in finding feasible schedules in a short computation time for increasing time horizons of traffic prediction.

In Wiklund (2007), the author describes a simulation procedure for simulating train traffic at a microscopic level in order to determine the effectiveness of various recovery strategies in case of large-scale disruptions. A case study involving a fire at the interlocking system of a station in Stockholm is considered.

Our work extends the ILP model presented in Louwerse and Huisman (2014), in which two double track lines were considered and a partial or full track blockage was taken into account. The new feature consists of dealing with a real-world railway network where the number of tracks within the stations and between the stations is not limited to two. Furthermore, trains are allowed to take other tracks within the stations or between the stations in comparison with the tracks they are originally scheduled to, and now all tracks may be used in both directions. To prevent overtakings of trains running on the same track, additional constraints are considered.

In addition, train reroutings along different paths are allowed in order to avoid the disrupted area. An advantage of rerouting trains is that passengers do not need to reroute themselves (possibly with some transfers) and that they can experience smaller delays.

As in Louwerse and Huisman (2014), we include rolling stock constraints in order to increase the probability of getting a feasible rolling stock schedule.

We also focus on all stages of the disruption management process, i.e. from the start of the disruption to the time at which the normal situation is restored. In Louwerse and Huisman (2014) only a new cyclic timetable is made for execution during the disruption. The authors do not consider the transition from the original timetable to the new temporary timetable, nor the transition back to the original timetable when the disruption has ended. In addition, they assume that, at the time the disruption starts, the network is empty. In our research we do not have these assumptions, and so we also determine how the timetable must be modified during the transition phases.

2.3 Problem description

In this paper we consider a real-time timetable rescheduling approach for railway networks. In case of a major disruption (i.e. temporarily blocked tracks) this approach is able to determine, by taking into account the available infrastructure capacity, which train services (or parts thereof) should be cancelled and which should be delayed such that as many trains as possible can still be operated. A train service is partially cancelled if it runs only to a subset of the stations it normally runs to. In most situations this means that the train ends at a location different than the planned end location or starts at a location different than the planned start location.

The use of the available infrastructure capacity in this approach is considered from a macroscopic point of view, and by taking into account rolling stock capacity there is a high probability that the new timetable has a feasible rolling stock schedule as well.

2.3.1 The disruption

For this approach we consider disruptions where a number of tracks between stations are blocked. There can be multiple track blockages at the same time at different locations. However, for the computational results we only consider disruptions where tracks at the same geographical location are blocked.

The duration of the disruption is assumed to be known and there is a fixed limited time available after the disruption before all trains must run again according to their original schedule.

Trains which passed their last stop before the blocked segment at the moment the disruption occurs need special attention. It is not clear whether these trains did or did not pass the critical point which caused the disruption. Therefore we assume in this research that these trains just continue as planned.

2.3.2 Resource restrictions and assumptions

We consider a railway network that consists of a set of stations (each one with a given capacity) and of a set of open track sections (the parts of the railway network between two consecutive stations), that can be single tracked, double tracked or with even more parallel tracks. A given set of trains runs on the railway network, each one according to its original timetable. Trains are characterized by a type, e.g. they can be regional or intercity trains. Each train uses a rolling stock composition, i.e. a set of coupled rolling stock units, of the same type as the train.

We distinguish three types of resources which a train may occupy at any given moment, namely tracks in open track sections, tracks in stations, and rolling stock compositions.

The capacity of a station is characterised by the number of tracks it has. Each track within the station may only be occupied by one train at any given time. Furthermore, after a train has used a track in a station, a certain headway time needs to pass before another train can use the same track. All tracks in a station are assumed to have a platform next to them and some stations have a shunting yard with an infinite capacity of tracks. An extension of the model could handle shunting yards with finite capacity. However the shunting yards are not considered to be the bottlenecks since we want to run (and not store) as many trains as possible.

The capacity of an open track section between two stations is also characterised by the number of tracks it has. The tracks can be used in both directions. It is assumed that a train cannot switch tracks while running on an open track section, and a track can only be used by multiple trains at the same time if the trains run in the same direction. A certain headway time should be taken into account between two trains using a track at the same time in the same direction, or between two trains using a track consecutively in opposite directions. It is assumed that a train entering a station from an open track section is able to reach every track in the station regardless of the track of the open track section it is entering from.

At each station with a shunting yard, a limited number of rolling stock compositions is available at the start of the day. A train uses a rolling stock composition for its entire duration, after which the rolling stock composition is moved to a shunting yard or used by another train. Hence rolling stock compositions may only end their duties at stations with shunting yards, and the compositions are not split during the day. Furthermore, two trains may share rolling stock only if they are of the same type. After a train has ended, a minimum turnaround time is required before the rolling stock composition of that train may be used by another train.

Finally, the minimum running time between two stations and the minimum dwell time inside a station should be respected by all trains. The arrival or departure of a train at a station may further be delayed by only a maximum amount of time, and trains may only end at their final destination or at their last stop before the disrupted tracks.

2.4 Mathematical formulation

Each train service is represented by a set of events, which are arrivals or departures at certain stations. The aim of the rescheduling approach is to determine the times at which these events take place or to decide to cancel some events.

To do this, the timetable rescheduling approach is based on an *event-activity network* represented by a directed graph $N = (E, A)$, where E is the set of vertices (*events*) and A the set of arcs (*activities*). The graph N is associated with a set of trains T and an original timetable for these trains. The set $E = E_{\text{train}} \cup E_{\text{inv}}$ of events consists of a set E_{train} of *train events*, and a set E_{inv} of *inventory events*.

Each train event $e \in E_{\text{train}}$ represents either a departure or an arrival at a certain station. The train of which train event e is a departure or arrival of, is denoted by t_e. The scheduled time at which train event e takes places in the original timetable is given by q_e. For each train event e, d_e denotes the maximum allowed delay for the event.

Each event $e \in E_{\text{train}}$ is associated with a set of resources (tracks in open track sections, tracks in stations, and rolling stock compositions) which it uses at the moment the event takes place.

An inventory event $e \in E_{\text{inv}}$ has different characteristics. It represents the resource inventory of a certain station, open track section or shunting yard at the start of the day. The number of resources made available by inventory event e is denoted by i_e.

The use of an activity $a = (e, f) \in A$ directed from event $e \in E$ to event $f \in E$ denotes the fact that event f uses one of the resources occupied by e after e has taken place. Between two events there can be multiple activities, and for every resource type there can only be one activity between the same pair of events. Each activity $a = (e, f) \in A$ has an associated minimum duration L_a which is necessary for the specific resource used by e to become available for use by f. In summary, the activities determine the possible orders in which the resource units are used for the events.

The Timetable Rescheduling Problem consists of delaying some trains and cancelling some other trains such that maximum delay and capacity constraints are satisfied while minimising the deviation from the original timetable. The first part of the proposed ILP formulation is given in (2.1)-(2.5). The further constraints are described in subsequent sections. In the model, x_e is a decision variable denoting the time at which event $e \in E_{\text{train}}$ takes place in the new timetable and y_t is a binary decision variable such that $y_t = 1$ if train $t \in T$ is cancelled, and $y_t = 0$ otherwise.

The objective function (2.1) is a weighted sum of the number of cancelled trains and the sum of the delays of all the events from their original scheduled times. For every train t, λ_t

describes the penalty for cancelling the train (which, for example, can depend on the train type or the running time), and for each event e, μ_e is the penalty per time unit for delaying the event. Constaint sets (2.2) and (2.3) ensure that an event does not take place before its scheduled time in the original timetable and that its maximum allowed delay is not exceeded. Constraints (2.3) also ensures that if a train is cancelled, then it "virtually" runs at its original time (i.e. no delay penalty is considered).

$$\text{Minimise} \quad \sum_{t \in T} \lambda_t y_t + \sum_{e \in E_{\text{train}}} \mu_e (x_e - q_e) \tag{2.1}$$

$$\text{subject to} \quad x_e - q_e \geq 0 \qquad \forall e \in E_{\text{train}} \tag{2.2}$$

$$x_e - q_e \leq (1 - y_{t_e}) d_e \qquad \forall e \in E_{\text{train}} \tag{2.3}$$

$$y_t \in \{0, 1\} \qquad \forall t \in T \tag{2.4}$$

$$x_e \in \mathbb{N} \qquad \forall e \in E_{\text{train}} \tag{2.5}$$

2.4.1 Capacity constraints

Capacity constraints are needed in order to ensure that a resource unit is not used by more than one train at a time, and that trains which are not able to get the resources they require are cancelled.

The capacity is handled by the activities $a \in A$. For each event e we divide the activities into two groups: in-activities and out-activities. The *in-activities* of event e are all activities into event e and are denoted by the set: $A^-(e) = \{a = (f, e) \in A | f \in E\}$. An in-activity $a = (f, e)$ of event e means that event e can use the same resource as event f at least L_a minutes after event f has taken place.

For each type of resource (tracks in open track sections, tracks in stations and rolling stock compositions) and for each event e we can define a subset $C \subset A^-(e)$ of the in-activities associated with that resource type only. Then, the set $\mathcal{A}^-(e)$ denotes the collection of these in-activity subsets for event $e \in E$. This means that for each event e, $|\mathcal{A}^-(e)|$ is smaller than or equal to three (the number of resource types). So for example, the set $\mathcal{A}^-(e)$ can contain three of these in-activity subsets: *i)* one subset of activities into event e associated with open track section capacity, *ii)* one subset of activities into event e associated with station capacity, and *iii)* one subset of activities into event e associated with rolling stock availability. The details of these types of activities are discussed below.

The second group of activities of an event e are the *out-activities* which are all activities out of event e and are denoted by the set $A^+(e) = \{a = (e, f) \in A | f \in E\}$. An out-activity $a = (e, f)$ of event e means that event f can use the same resource as event e at least L_a minutes after event e has taken place. We have for each type of resource a subset $C \subset A^+(e)$ of the out-activities. These subsets are grouped in the set $\mathcal{A}^+(e)$ denoting the collection of out-activity subsets for event e.

Not every event has all three subsets of in-activities or out-activities. This is because for some events it is fixed which event will be the predecessor or successor event using the same resource and there is no minimum time involved before the resource becomes available. Therefore, we left out decision variables for these activities for which we can easily determine the values.

For example, considering the station capacity, it is sure that after an arrival of a train at a certain track in a station, the next event on that track must be the departure of that train (unless the train has reached its last station). Furthermore, since these events are related to a single train, there is no headway time involved between them.

The capacity constraints are given by constraint sets (2.6)-(2.10), where z_a is a binary decision variable such that $z_a = 1$ if activity $a \in A$ is selected, and 0 otherwise.

$$\sum_{a \in C} z_a + y_{t_e} = 1 \qquad \forall e \in E_{\text{train}}, C \in \mathcal{A}^-(e) \tag{2.6}$$

$$\sum_{a \in C} z_a + y_{t_e} \leq 1 \qquad \forall e \in E_{\text{train}}, C \in \mathcal{A}^+(e) \tag{2.7}$$

$$\sum_{a \in C} z_a \leq i_e \qquad \forall e \in E_{\text{inv}}, C \in \mathcal{A}^+(e) \tag{2.8}$$

$$x_f - x_e + M(1 - z_a) \geq L_a \qquad \forall a = (e, f) \in A \tag{2.9}$$

$$z_a \in \{0, 1\} \qquad \forall a \in A \tag{2.10}$$

For a given event $e \in E_{\text{train}}$ and a subset of activities $C \in \mathcal{A}^-(e)$, constraint set (2.6) ensures that exactly one unit of the resource associated with subset C must be made available to event e or that the corresponding train must be cancelled. In other words, these contraints ensure that the train is cancelled if at least one resource is not available for an event, and that it is not cancelled if one unit of all resource types is available.

Similarly, for a given event $e \in E_{\text{train}}$ and subset of activities $C \in \mathcal{A}^+(e)$, constraint set (2.7) ensures that at most one unit of the associated resource is made available by event e to a successor event using the same resource unit. Finally constraint set (2.8) ensures that the available inventory for a specific type of resources is not exceeded, and constraint set (2.9) (where M is a large positive value) ensures that the minimum duration of an activity, if selected, is maintained.

Train service

The minimum running and dwell times of trains should be respected. This can be modelled by the capacity constraints of Section 2.4.1. However, in contrast with the rolling stock and track usage, for each train the order of the running and dwell events is fixed. Therefore we do not need constraint sets (2.6), (2.7) and (2.8).

Let A_{train} be the set of all train activities $a = (e, f)$ which represent running or dwelling of a train between consecutive events e and f of the same train. Then for train activities $a \in A_{\text{train}}$ constraints (2.9) can be modified into:

$$x_f - x_e \geq L_a \qquad \forall a = (e, f) \in A_{\text{train}} \tag{2.11}$$

Open track section capacities

An open track section between two stations has a limited number of tracks which may be used in both directions. Multiple trains may use a track at the same time only if they run into the same direction, but a minimum headway between the two trains must be considered.

To model this, let $e^{k\ell}_{\text{track}}$ denote the initial open track section inventory event for open track section (k, ℓ) (the open track section between stations k and ℓ), let $E^{k\ell}_{\text{arr}} \subset E_{\text{train}}$ denote the set of events corresponding to an arrival at station ℓ of a train that departed from station k, and let $E^{k\ell}_{\text{dep}} \subset E_{\text{train}}$ denote the set of events corresponding to a departure from station k of a train in the direction of station ℓ.

In what follows the *open track section activities* for the open track section (k, ℓ) are described. This class of activities represents the sequential use of a track of the open track section or the very first use of a track of the open track section. We only describe the events at the side of station k. At the side of station ℓ the activities are constructed similarly.

- For an event $e \in E^{k\ell}_{\text{dep}}$ representing the departure of train t_e from station k onto the open track section (k, ℓ):
  - The set $\mathcal{A}^-(e)$ contains a subset of open track section activities from each event $f \in E^{k\ell}_{\text{dep}} \backslash \{e\}$ to e, from each event $f \in E^{\ell k}_{\text{arr}}$ to e, and from the inventory event $e^{k\ell}_{\text{track}}$ to e. Thus for these activities, constraint set (2.6) implies that a train can only depart on a track of an open track section if a train has departed on the same track in the same direction, if a track used in the opposite direction became empty after an arrival, or if there is an empty track available from the inventory.

- The set $\mathcal{A}^+(e)$ contains a subset of open track section activities to each event $f \in E^{k\ell}_{\text{dep}} \backslash \{e\}$. Here constraint set (2.7) implies that, if a train departs onto an open track section, then at most one other train can depart directly after this train on the same track in the same direction.

- For an event $e \in E^{\ell k}_{\text{arr}}$ representing the arrival of train t_e at station k from the open track section (ℓ, k):
  - There is no set $\mathcal{A}^-(e)$, since there is no capacity restriction for the arrival event as the train is already running on the track. We only have to take care that trains do not overtake each other which will be discussed in constraint set (2.12).
  - The set $\mathcal{A}^+(e)$ contains a subset of open track section activities to each event $f \in E^{\ell k}_{\text{arr}} \backslash \{e\}$, and to each event $f \in E^{k\ell}_{\text{dep}}$. Here constraint set (2.7) implies that, if a train arrives from a track of the open track section, then either another train arrives directly after it from the same track, or the track will be used in another direction by a train departing from station k.
- The set $\mathcal{A}^+(e^{k\ell}_{\text{track}})$ contains a subset of open track section activities from the inventory event $e^{k\ell}_{\text{track}}$ to each event $e \in E^{k\ell}_{\text{dep}} \cup E^{\ell k}_{\text{dep}}$. Here constraint set (2.8) implies that at most $i_{e^{k\ell}_{\text{track}}}$ trains (equal to the number of tracks between stations k and ℓ) may depart onto a track which has not been used before.

For these classes of activities, constraint set (2.9) models the headway time which has to be taken into account between two consecutive trains using the same track.

Figure 2.1 shows an example of a graph of open track section activities. Here we consider an open track between stations A and B. The events are represented by nodes and placed in a time-space plot where the vertical direction represents space and the horizontal direction represents time. For a better understanding of the activities, we assume in this example the time at which an event will take place as fixed. This time is stated within the nodes together with info about whether it is an arrival (*Arr*) or departure (*Dep*). The capacity of an inventory event is given between brackets. The open track section activities are represented by arcs. To make the graph more intuitive, the train activities are also included and represented by dotted lines. This means that two nodes linked with a dotted line, represent events of the same train.

Additional constraints are needed to prevent overtaking of trains on the same track. Therefore, *track activity pairs* are introduced. For open track section (k, ℓ) and events $e, f \in E^k_{\text{dep}}$ and $e', f' \in E^\ell_{\text{arr}}$ such that $t_e = t_{e'}$ and $t_f = t_{f'}$, let $a = (e, f)$ be the activity presenting the consecutive departures of trains t_e and t_f from station k on a track in (k, ℓ), while $a' = (e', f')$

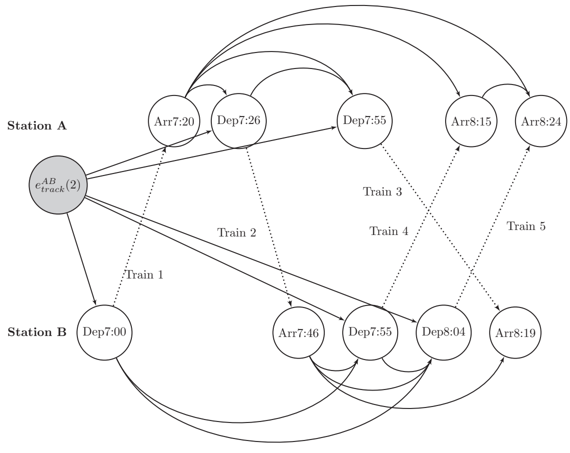

Figure 2.1: Example of Event-Activity graph for open track section capacity

is the activity corresponding to the consecutive arrivals of the two trains at station ℓ (as defined above). Then the pair (a, a') is a *track activity pair*, and B is defined as the set of all track activity pairs. In order to ensure that no overtaking on tracks takes place, constraint set (2.12) is required.

$$z_a = z_{a'} \qquad \forall (a, a') \in B \tag{2.12}$$

Hence for a track activity pair (a, a'), if a is selected then so must be a'. This ensures that the order in which two trains arrive from a track is the same as the order in which they departed.

Note that forcing tracks to be used in only one direction can easily be achieved by not including activities from arrival events to departure events. Then the only in-activities for departure events come from other departure events. Also, we should construct an inventory event for each direction. The result will be that there are two disjoint graphs: one for each direction.

Figures 2.2-2.5 show, with bold arcs for the selected activities, all feasible solutions for the open track section capacity problem considered in Figure 2.1 with fixed event times. Note that the dashed arcs are train activities instead of open track section activities. These arcs are included such that one can easily see how events use the same resource unit. A selected path, in bold arcs, starting from the inventory event $e^{k\ell}_{track}$ indicates in which order events take place on a single track in the open track section. For example in Figure 2.2 each track is used in only one direction. Trains 1, 4 and 5 use one track and Trains 2 and 3 use the other track.

Furthermore, Figures 2.2 and 2.5 show how constraints (2.12) work between Trains 2 and 3. Due to constraint set (2.12) both the arc between the departures of these trains and the arc between the arrivals of these trains are used. The same holds for Trains 4 and 5 in all Figures 2.2-2.5.

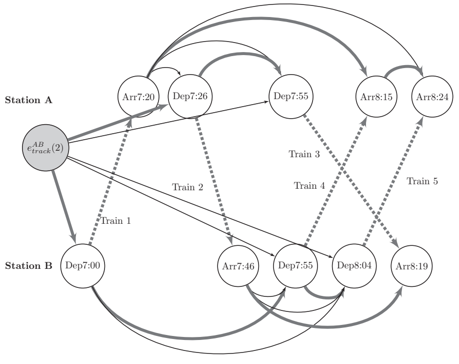

Figure 2.2: Solution 1

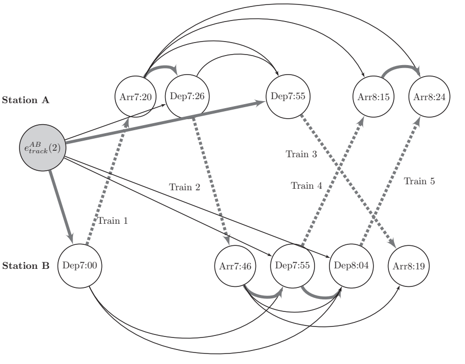

Figure 2.3: Solution 2

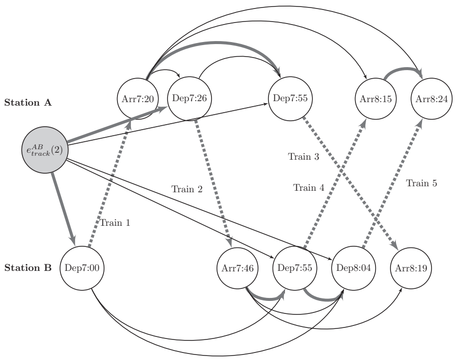

Figure 2.4: Solution 3

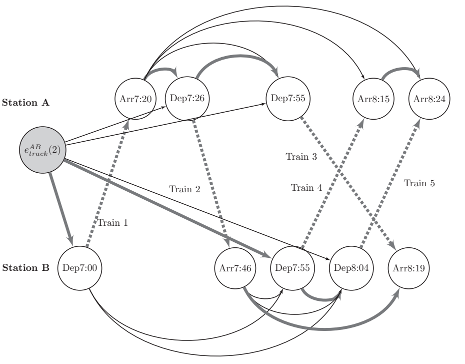

Figure 2.5: Solution 4

Station capacities

In a station a train needs to be assigned to a track with a platform to dwell or to pass if it does not have a scheduled stop. There cannot be two trains at the same track at the same time. After a train has left the track in the station, another train can arrive on that track.

To model this, we need the sets E^k_{arr} and E^k_{dep} which are all arrival and departure events at station k, respectively. Furthermore, let e^k_{stat} denote the track inventory event at station k.

In the following, the *station activities* are defined. These activities represent the sequential use of the same track in a station or the very first use of a track in a station.

- For an event $e \in E^k_{\text{arr}}$, representing the arrival of train t_e at station k:
  - The set $\mathcal{A}^-(e)$ contains a subset of station activities from each event $f \in E^k_{\text{dep}}$ to e, and from the inventory event e^k_{stat} to e. For this class of activities, constraint set (2.6) implies that a train can only arrive at station k if there is a track available. This means that a previous train has departed from a track in the station, or that a track has not been used before.
  - The set $\mathcal{A}^+(e)$ is not considered. It is sure that the next event using the same track is the departure of train t_e at station k since one train at the time is allowed on one track. We make use of this information and do not make it a decision variable in the model.
- For an event $e \in E^k_{\text{dep}}$, representing the departure of train t_e from station k:
  - The set $\mathcal{A}^-(e)$ is not considered for the same reason as above. It is sure that the previous event using the same track is the arrival of train t_e at station k.
  - The set $\mathcal{A}^+(e)$ contains a subset of station activities to each event $f \in E^k_{\text{arr}}$. For this class of activities, constraint set (2.7) implies that, if a train has departed from a station, then the track may be assigned to at most one new arrival.
- The set $\mathcal{A}^+(e^k_{\text{stat}})$ contains a subset of station activities from the inventory event e^k_{stat} to each event $e \in E^k_{\text{arr}}$. Here constraint set (2.8) implies that at most $i_{e^k_{\text{stat}}}$ trains (equal to the number of tracks at that station) may arrive at a track of station k which has not been used before.
- Some events in $e \in E^k_{\text{dep}}$ represent a departure from station k which is the first departure of a train. For these events it is not clear if the rolling stock for the train arrives from the shunting yard or whether the train uses rolling stock which was waiting at a station track after it arrived servicing a train which ended in station k. For a departure event e which represents a first departure of a train, we construct the set $\mathcal{A}^-(e)$ in the same way as if e is an arrival event and the set $\mathcal{A}^+(e)$ in the same way as if event e is a departure event. This way it is modeled that train t_e needs to have a track available at the time it departs. The occupation of the tracks by rolling stock assigned to train t_e is handled correctly by the rolling stock constraints which are discussed later.

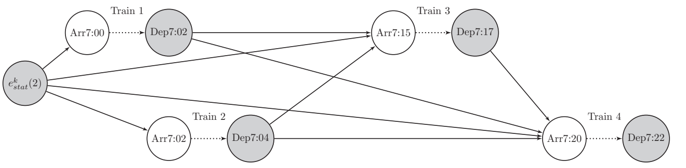

Figure 2.6: Example of Event-Activity graph for station capacity

- Some events in $e \in E^k_{\text{arr}}$ represent an arrival at station k which is the last arrival of a train. For these events it is not clear whether the rolling stock goes to the shunting yard or whether it stays at the station track to be used by another train. For an arrival event e which represents a first arrival of a train, we construct the set $\mathcal{A}^-(e)$ in the same way as if event e is an arrival and the set $\mathcal{A}^+(e)$ in the same way as if event e is a departure. This way it is modeled that train t_e needs an unoccupied track to arrive, and that the track is released for the arrival of another train directly (minimum L_a minutes) afterwards.

For these classes of activities, constraint set (2.9) models the headway time which has to be taken into account between a departure from a track in a station and an arrival on the same track.

Figure 2.6 shows an example of a graph of station activities. The events are represented by nodes and are considered to have a fixed time which is stated within the node. In this figure, the horizontal direction represents time. Furthermore, the capacity of an inventory event is given between brackets. The station activities are represented by arcs and the train activities are also included and represented by dotted lines.

Figures 2.7-2.10 show, with bold arcs for the selected activities, all feasible solutions for the station capacity problem shown in Figure 2.6, given that the times the events take place are fixed. Note that the dashed arcs are train activities instead of station activities. A selected path, in bold arcs, from the inventory event e^k_{stat} indicates in which order events take place on a single track within the station. For example in Figure 2.7, Trains 1, 3 and 4 use one track and Train 2 uses the other track.

Rolling stock capacities

Every train needs rolling stock units to run. In our model we can have different types of rolling stock: rolling stock for intercity trains and rolling stock for regional trains. We assume that starting trains can use rolling stock of ending trains which are of the same type, or they can use rolling stock from the shunting yard, if available.

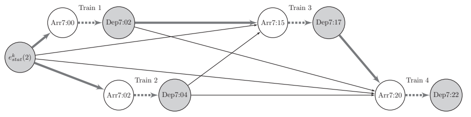

Figure 2.7: Solution 1

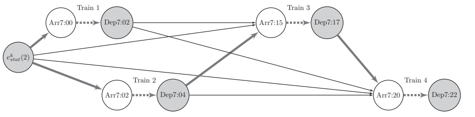

Figure 2.8: Solution 2

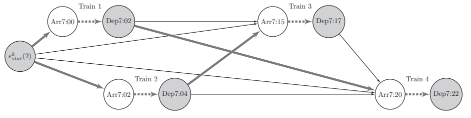

Figure 2.9: Solution 3

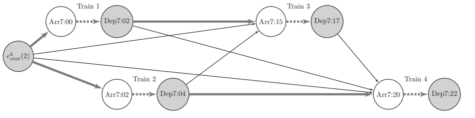

Figure 2.10: Solution 4

To model the assumptions in the structure of Section 2.4.1, let $E^k_{\text{start}} \subset E_{\text{train}}$ denote the set of departure events corresponding to the start of a train from station k, let $E^k_{\text{end}} \subset E_{\text{train}}$ denote the set of arrival events corresponding to the end of a train at station k, and let e^k_{rol} denote the initial rolling stock inventory event at station k.

In the following, the *rolling stock activities* are defined. These activities represent the transfer of rolling stock from an ending train to a starting train, or the very first use of a rolling stock composition.

- For an event $e \in E^k_{\text{start}}$, denoting a start of a train from station k:
  - The set $\mathcal{A}^-(e)$ contains a subset of rolling stock activities from each event $f \in E^k_{\text{end}}$ to event e if events e and f are using the same type of rolling stock, and from the inventory event e^k_{rol} to event e. For this class of activities, constraint set (2.6) implies that a train can only depart if an earlier train has ended at that station, or if there is still a rolling stock composition available in the inventory.
  - The set $\mathcal{A}^+(e)$ is not constructed since the next event using the same rolling stock is known (the arrival event of train t_e at the next station) and so no decision variables are needed.
- For an event $e \in E^k_{\text{end}}$,
  - The set $\mathcal{A}^-(e)$ is not constructed since the previous event using the same rolling stock is known (the departure event of train t_e from the previous station) and so no decision variables are needed.
  - The set $\mathcal{A}^+(e)$ contains a subset of rolling stock activities to each event $f \in E^k_{\text{start}}$. Here constraint set (2.7) implies that, if a train has ended at a station, its rolling stock can be assigned to at most one other train.

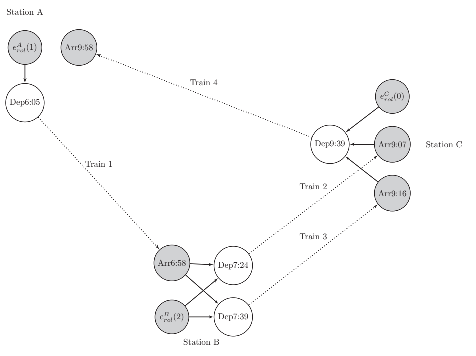

Figure 2.11: Example of Event-Activity graph for rolling stock capacity

- The set $\mathcal{A}^+(e^k_{\text{rol}})$ contains a subset of rolling stock activities from the inventory event e^k_{rol} to each event $f \in E^k_{\text{start}}$. Here constraint set (2.8) implies that at most $i_{e^k_{\text{rol}}}$ trains may start from station k by using rolling stock compositions available at station k.

- For arrival and departure events which do not represent the start of a train nor the end of a train, no rolling stock activities are constructed. Because we do not allow trains to switch rolling stock during one train service, all events of a train service use the same rolling stock. We use this information and do not need to model it by decision variables.

Furthermore, constraint set (2.9) models the turnaround time required to transfer rolling stock from an ending train to a starting train, and the time required for shunting activities if a starting train uses rolling stock from the inventory.

Figure 2.11 shows an example of a graph of rolling stock activities. The events are represented by nodes and ordered by location. The time at which an event takes place is assumed to be fixed and stated within the node. The capacity of an inventory event is given between brackets. The rolling stock activities are represented by arcs and the train activities are also included and represented by dotted lines.

Figures 2.12 and 2.13 show with bold arcs two of the feasible solutions for the rolling stock capacity problem shown in Figure 2.11. Note that again the dashed arcs are train activities. A selected path from one of the inventory events (e^A_{rol}, e^B_{rol} and e^C_{rol}) indicates in which order events use a rolling stock composition. For example in Figure 2.12, Train 1 uses a rolling stock composition from the inventory of Station A which will be left in Station B. Trains 2 and 3 both get their rolling stock composition from the inventory at Station B and Train 4 uses the same rolling stock composition as Train 2.

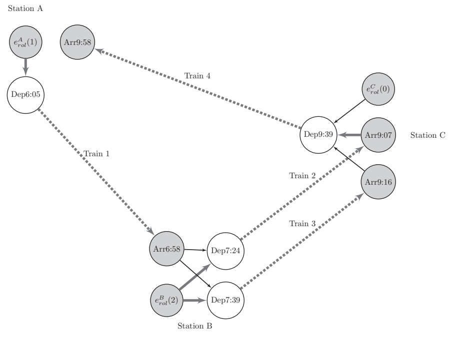

Figure 2.12: Solution 1

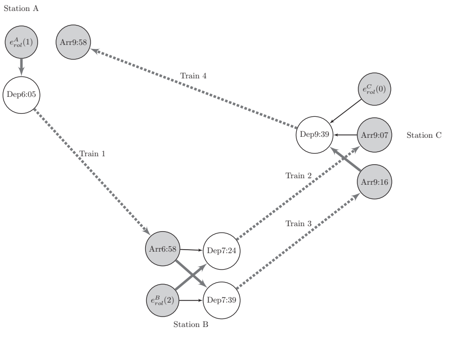

Figure 2.13: Solution 2

If a station has a shunting yard, then the rolling stock composition of an ending train can be moved to the shunting yard before it is used by a starting train again. To model this, we add the rolling stock activities which include a back-and-forth to the shunting yard to the subset of station activities. If the rolling stock composition goes to the shunting yard (with infinite capacity), then a track in the station becomes available, and a track needs to be available at the moment the rolling stock composition comes back from the shunting yard. The subset of rolling stock activities does not change to ensure that still every train has a rolling stock composition.

Furthermore, in the operations, it is preferred to have a regular turning pattern for the rolling stock units, since this requires less communication with the shunting crews. Each train belongs to a *series*, which is a set of train services that have the same departure station, the same stops and the same arrival station. In a cyclic timetable in every cycle one train of each train series runs. Trains belonging to the same train series traveling in the same direction belong to the same *subseries*. A *turning pattern* is a pair of subseries $(s, s') \in S \times S$, where S denotes the set of all subseries. A turning pattern represents the transfer of rolling stock from a train belonging to s to a train belonging to s'. A rolling stock activity a corresponding to the transfer of rolling stock between two trains therefore belongs to a turning pattern. This turning pattern is denoted by w_a. Furthermore, W is the set of all possible turning patterns, W_s is the set of all turning patterns containing subseries s, and u_w is a decision variable such that $u_w = 1$ if the activities must correspond to turning pattern w is selected and 0 otherwise. Then constraint sets (2.13)-(2.15) ensure a regular turning pattern in which rolling stock units of ending trains of a certain subseries are used only by starting trains which are from the same subseries.

$$\sum_{w \in W_s} u_w \leq 1 \qquad \forall s \in S \tag{2.13}$$

$$z_a \leq u_{w_a} \qquad \forall a = (e, f) \in A_{\text{rol}} \tag{2.14}$$

$$u_w \in \{0, 1\} \qquad \forall w \in W \tag{2.15}$$

2.4.2 Blockage of an open track section

The disruptions considered in this paper consist of blockages of tracks for a known duration. A *partial blockage* is defined as the temporary unavailability of a subset of tracks between two stations. When all tracks between two stations are blocked, we call it a *full blockage*. Note that from a modeling point of view a full blockage is a special case of a partial blockage.

In our test instances we only consider blockages of tracks within an open track section. However, blockages of tracks within a station can be handled in the same way. Another type of disruption is a defective rolling stock unit. Although this also results in a track blockage, another way to handle this kind of disruptions is by increasing the arrival time of this specific train with the time it will take to fix the rolling stock.

Here we limit ourselves to blockages of tracks. Due to the blockage of the tracks, some additional events and activities should be added. The time τ_1 at which the disruption starts and the time τ_2 at which the disruption ends is given as input, together with a list of open track sections which are partially blocked. If there is more than one track in the open track section connecting two stations, information about which tracks are blocked is given as well.

We do not assume that the network is empty at the moment τ_1 the disruption starts. Therefore we should also take into account what has happened before time τ_1, but we cannot change this.

Disrupted trains

Trains running over the disrupted area need special attention. For each train, a station is classified as a stopping station if a stop has to be made, or as a pass-through station otherwise.

Assume there is a partial blockage of an open track section on which train $t \in T$ is scheduled. Let station k (ℓ) be the last (first) stopping station of train t before (after) the blocked section. Then, if a train t has an arrival time at station k during the disruption, the events that would have been associated with train t (in case of no disruption) are now partitioned and associated with three new trains, namely trains t_α, t_β and t_γ.

Let e^{start} and e^{end} denote the first departure event and the last arrival event of train t, respectively, let e_k^{arr} and e_ℓ^{arr} denote the arrival events at k and ℓ, respectively, and let e_k^{dep} and e_ℓ^{dep}

denote the departure events from k and ℓ, respectively. Train t_α has events e^{start} to e_k^{arr}, train t_β has events e_k^{dep} to e_ℓ^{arr} and train t_γ has events e_ℓ^{dep} to e^{end}.

Furthermore, constraint sets (2.16) and (2.17) are included to ensure that if train t_β (the train over the disrupted area) runs, then both trains t_α and t_γ also run. This is equivalent to running the original train t. If t_β is cancelled, however, then t_α and t_γ may run or be cancelled, independently of each other. This is modelled as follows:

$$y_{t_\beta} \geq y_{t_\alpha} \tag{2.16}$$

$$y_{t_\beta} \geq y_{t_\gamma} \tag{2.17}$$

Additional rolling stock activities

For trains t_α and t_γ, all activities are defined as discussed in Section 2.4.1. However, in order to ensure that trains t_α, t_β and t_γ use the same rolling stock if they all run, a rolling stock activity (with a duration 0) is defined from events e_k^{arr} to e_k^{dep} and from events e_ℓ^{arr} to e_ℓ^{dep}. Furthermore, no other rolling stock activities are defined for train t_β. For event e_ℓ^{arr}, constraint (2.7) becomes an equality constraint. Hence in the case where none of these three trains is cancelled, they all use the same rolling stock, whereas if train t_β is cancelled, the rolling stock units of trains t_α and t_γ may turn on other trains at stations k and ℓ respectively.

Additional open track section activities

If train t leaves from station k before the disruption, but the disruption starts before train t reaches station ℓ, then it is assumed that train t continues along its original route. The reason for this assumption is that decisions to be made for this kind of trains depend on microscopic details of the disruption, such as whether the train is before or after the broken switches or overhead wires, or whether this is the train whose rolling stock is malfunctioning. In these cases, the events associated with train t are the same as in the case of no disruption. The open track section activities for such a train are, however, defined differently. The open track section activities of the arrival event of t at the end of the disrupted track section are only defined for events which take place after the disruption (after τ_2). This ensures that the blocked tracks are only used again after the disruption has ended.

The assumption that these trains continue as planned is a choice made by the authors. It does not affect the model itself. In specific situations also another choice could be made, for example that the trains in the disrupted area are cancelled, or return to station k.

Balancing directions

If the number of tracks is reduced, then more trains can run if all trains run in the same direction. To avoid such unbalanced timetables, which are not preferred in practice, the following constraints can be added to the formulation for each pair of subseries $(s, s') \in S \times S$ for which s and s' belong to the same train series, but differ in direction.

$$\sum_{t \in s} y_t \leq \sum_{t \in s'} y_t + 1 \tag{2.18}$$

$$\sum_{t \in s'} y_t \leq \sum_{t \in s} y_t + 1 \tag{2.19}$$

These constraints ensure that for every train series the number of trains in one direction cannot exceed the number of trains in the other direction by more than one. Note that such constraints are also (partially) enforced by the rolling stock circulation.

2.4.3 Managing all stages of the disruption process

To correctly handle the disruption, the time of all events that took place before the start of the disruption is fixed, and trains associated with these events cannot be cancelled, since they are already running. Moreover, it is preferable that the disruption does not affect the timetable for the complete day. Therefore it is assumed that at some point in time after the disruption has ended the trains should run according to their original timetable again. For this purpose a time $\tau_3 > \tau_2$ is specified such that any event that takes place after τ_3 cannot be delayed and such that a train starting after τ_3 cannot be cancelled.

The set of events E therefore only needs to contain events that are scheduled to take place after τ_1 and before τ_3, together with some events outside this range in order to correctly model the availability of capacities at time τ_1, and to ensure a smooth recovery to the original timetable at time τ_3. Without going into too much detail, especially events which took place after τ_1 minus the minimum headway or turn around time, as well as events which have to take place before τ_3 plus the minimum headway or turn around time should be considered. If these events are not considered the model assumes that at τ_1 all tracks are empty and directly available. Furthermore it then does not consider that there must be tracks available for the trains starting after τ_3.

Secondly, after the disruption there must be enough rolling stock at every station to run the timetable for the remainder of the day. Therefore, at every station an inventory event is added which is scheduled at the time the disruption is over, connected with the rolling stock activities. The number of selected rolling stock activities to this event must equal the number of rolling stock compositions there are normally (without disruption) at that station at that time.

2.4.4 Reroutings

One of the contributions of this paper is that we include the possibility for trains to be rerouted in order to avoid the disrupted area if the network under consideration allows this. The advantage of having an option to reroute a train is that passengers wishing to use this train do not have to reroute themselves (possibly with some transfers). Furthermore, passengers may experience a smaller amount of delay, and the normal trains running on the rerouted area may be less crowded.

To incorporate this functionality, for each train t which is scheduled to travel through a disrupted area an alternative list of stations between station k (the last stop before the blockage) and station ℓ (the first stop after the blockage) can be provided as rerouting option. Then, in addition to the new trains t_α, t_β and t_γ (compensating for the blocked area), a fourth train t_δ is defined which runs on the rerouted path. For each station on the rerouted path (apart from stations k and ℓ), an arrival event and a departure event is associated with train t_δ, while at station k a departure event representing the start of the train is associated with t_δ, and at station ℓ an arrival event representing the end of the train is associated with t_δ.

For train t_δ the scheduled time of the departure event at stations k is the same as for train t_β, while the arrival and departure times for the other events of t_δ are determined using information on the minimum running times necessary between the stations on the rerouted path. In order to ensure that at most one of the two trains t_β and t_δ runs, and that, if one of them runs, then both t_α and t_γ also run, constraint sets (2.20)-(2.21) replace constraint sets (2.16)-(2.17) for train t.

$$y_{t_\beta} + y_{t_\delta} \geq 1 + y_{t_\alpha} \tag{2.20}$$

$$y_{t_\beta} + y_{t_\delta} \geq 1 + y_{t_\gamma} \tag{2.21}$$

Furthermore, to handle a penalty for rerouting, a variable p_t is introduced for all trains t_β which have a rerouting option t_δ. The decision variable $p_t = 1$ if the train is cancelled and not rerouted, and $p_t = 0$ if the train runs as planned or is rerouted. To model this, constraints (2.22) are added.

$$y_{t_\beta} + y_{t_\delta} \leq 1 + p_{t_\beta} \tag{2.22}$$

In the objective function $\lambda_{t_\beta} y_{t_\beta}$ is replaced by $\theta \cdot \lambda_{t_\beta} y_{t_\beta} + (1-\theta) \cdot \lambda_{t_\beta} p_{t_\beta}$ where θ is a parameter between 0 and 1 indicating the balance between the cost of rerouting and cancelling a train. Furthermore $\lambda_{t_\delta} = 0$ for the rerouted copy of the train. These settings assure that if train t_β runs as planned ($y_{t_\beta} = 0$, $y_{t_\delta} = 1$ and $p_{t_\beta} = 0$), the costs in the objective function are

0. If train t_β is rerouted ($y_{t_\beta} = 1$, $y_{t_\delta} = 0$ and $p_{t_\beta} = 0$), the costs are $\theta \cdot \lambda_{t_\beta}$ and if train t_β is cancelled and not rerouted ($y_{t_\beta} = 1$, $y_{t_\delta} = 1$ and $p_{t_\beta} = 1$), the costs are λ_{t_β}.

2.5 Computational experiments

In order to test our approach, we carried out computational experiments on part of the Dutch railway network. The trains considered in this region are trains of Netherlands Railways, which is the major railway operator of the Netherlands. The mathematical model is solved to optimality (with a gap of 0.01%) by CPLEX 12.4 on a PC with an Intel Xeon with 3.1 GHz and 16 GB RAM.

2.5.1 Case description

For our computational tests we consider a heavily used part of the Dutch railway network which is indicated in Figure 2.14. This network consists of 39 stations. At some stations, mostly located at a double tracked section, trains cannot switch tracks. Therefore, trains cannot overtake each other in those stations. This means that those stations can be considered as part of an open track section and do not have to be included as a station. In our computational tests, we consider 26 stations/junctions where trains can switch tracks. Note that a junction has the same characteristics as a station, with the difference that trains do not have a scheduled stop there. Therefore, departures are allowed to take place there earlier than scheduled.

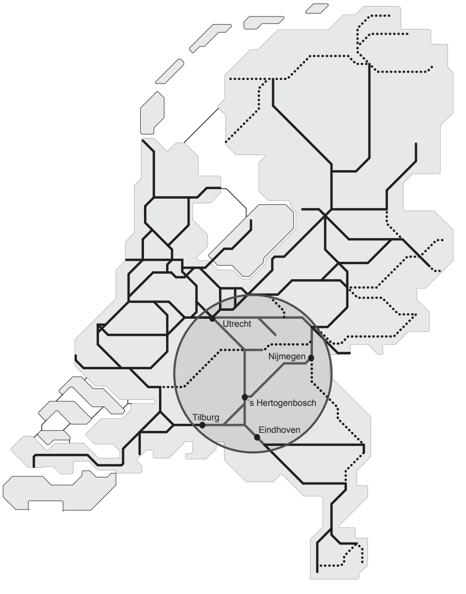

Figure 2.14: Overview of the Dutch railway network

Furthermore, our network consists of 27 open track sections between the considered stations. Of these sections, 3 are single tracked, 21 are double tracked, 1 has three parallel tracks and 2 have four parallel tracks.

In total 6 intercity and 10 regional train series run (mostly twice an hour in each direction) on this network which results in more than 60 trains per hour. We only consider the minimum number of rolling stock compositions which is required to run all these trains. This means that any spare rolling stock compositions at the shunting yards are not considered. In total 61 rolling stock compositions are necessary to run the trains of the complete day.

2.5.2 Parameter settings

In the mathematical model there are parameters for events and parameters for activities. First the parameters for the events are considered. The scheduled event times q_e are copied from the timetable of Netherlands Railways and the capacities of the stations, open track sections and rolling stock inventories i_e are set conform the described network in Section 2.5.1.

For the maximum allowed delay d_e of an train event e we make a distinction between three types of events. Trains running at the start of the disruption may be delayed more than a train which has not started yet. This prevents infeasibilities of the mathematical model, since running trains are not allowed to be cancelled. For events $e \in E_{\text{train}}$ of trains which are already running at the time the disruption starts we have $d_e = 30$ minutes, for events $e \in E_{\text{train}}$ of rerouted trains the maximum delay is equal to $d_e = 15$ minutes. For all other events $e \in E_{\text{train}}$ we have $d_e = \underline{d}$, where the value of $\underline{d}$ varies between 0, 3, 5 and 10 minutes over the different experiments.

For an activity, the minimum duration of the activity L_a can have multiple meanings. The minimum running time of a train L_a is set equal to the scheduled running time in the timetable, and the minimum dwell time is equal to the scheduled dwell time with a maximum of 2 minutes. The minimum headway time on the open track sections L_a is equal to 2 minutes if the trains run in the same direction and 0 minutes if the trains run in opposite directions. Within the stations the minimum headway time L_a is equal to 2 minutes.

The time L_a required before a rolling stock composition of an ended train can be used for a new starting train is 5 minutes. If the activity between an ending and a starting train takes longer than 10 minutes at a station with a shunting yard, the rolling stock composition goes to the shunting yard and releases the station capacity 5 minutes after the train has ended and requires station capacity from 5 minutes before the next train starts.

In the objective function we have two penalties. Penalties for delaying trains and penalties for cancelling trains. The aim is to run as many trains as possible. Therefore cancelling a train is penalized much more than delaying one. Cancelling a train is penalized by 50 times the running time of the train. Furthermore for every event there is a penalty of 1 per delayed minute. If a train is rerouted the costs are 20% of the costs for cancelling that train such that rerouting is preferred over cancelling the train, but also that the original route is preferred over the alternative route.

This research does not focus on *how* the available capacities are used. Just the utilization of the capacity is maximized, and if multiple capacity allocations lead to the same utilization, then just one possible allocation is provided, since the objective function does not contain penalties on activities.

2.5.3 Disruption scenarios

In order to test the described approach, a large set of disruption scenarios is created. For all of the 27 open track sections we constructed 30 scenarios of full blockages where all tracks of that section are blocked, and for the 24 open track sections with more than one track, an additional set of 30 scenarios is constructed where only one track is blocked. The first scenario is a 2 hour disruption of that open track section starting at 9:00. Then we increase in every new scenario the start time of the 2 hour disruption by one minute. Since the timetable of Netherlands Railways in this region is cyclic with a cycle time of 30 minutes, a disruption starting at 9:30 should be very similar to a disruption starting at 9:00. In total this leads to 810 scenarios of full blockages and 720 scenarios where only one track is blocked.

We take a buffer time of 1 hour into account before all trains should be able to run as planned again after the disruption is over. This means that with a 2 hour blockage all trains in a 3 hour period are taken into consideration in the timetable rescheduling.

2.5.4 Rerouting of trains

If there is a disruption between 's Hertogenbosch and Eindhoven, then trains can be rerouted via Tilburg (see Figure 2.14). For this case, 30 scenarios where all tracks are blocked and 30 scenarios where one of the two tracks is blocked are constructed in a similar way as described in Section 2.5.3. In these scenarios the trains of one of the Intercity lines were allowed to be rerouted. This intercity line runs twice an hour in each direction, which means that in a disruption of 2 hours 8 trains can be rerouted.

To include the rerouted trains in the timetable, more events need to get a new time x_e which differs from the scheduled time q_e. This means that it is harder to find the optimal solution. Therefore, for these cases, we first find the solution for the case where rerouting is not allowed, and then, in a second run, that solution is used as start solution for the case with rerouting. In the results, the presented computation times include the computation times of both runs.

	Minutes of delay allowed	Cancelled trains			Partially cancelled trains			Cancelled minutes			Computation time (s)		
		Min	*Avg*	*Max*	*Min*	*Avg*	*Max*	*Min*	*Avg*	*Max*	*Min*	*Avg*	*Max*
Complete blockage (808 instances)	*0*	0	2.0	12	0	4.7	18	28	405	947	3	3.2	12
	3	0	1.6	9	0	3.6	15	28	389	908	4	5.0	8
	5	0	1.4	10	0	3.3	15	28	381	905	4	5.9	11
	10	0	1.3	8	0	3.5	17	28	372	881	5	14.2	89
One track blocked no balancing (720 instances)	*0*	0	1.0	6	0	2.1	11	0	156	474	2	4.3	18
	3	0	0.6	6	0	1.5	8	0	115	348	4	6.9	29
	5	0	0.6	5	0	1.0	7	0	99	335	5	9.0	39
	10	0	0.5	4	0	0.7	5	0	68	286	6	43.3	971
One track blocked with balancing (720 instances)	*0*	0	0.9	6	0	1.9	11	0	166	474	3	4.6	14
	3	0	0.7	7	0	1.3	8	0	127	420	4	7.8	49
	5	0	0.8	5	0	1.0	6	0	108	420	5	10.1	75
	10	0	0.5	4	0	0.6	5	0	74	300	6	48.2	1992

Table 2.1: Results on cancellations and computation time

2.5.5 General results

In this section we discuss the results of all 1530 scenarios and in Section 2.5.6 we describe the results in which we included the option to reroute trains in 60 of the scenarios as described in Section 2.5.4.

Tables 2.1 and 2.2 contain the minimum, average and maximum values of the performance measures over all feasible scenarios for the complete blockages and the blockages of one track. Note that minimum (maximum) value represent the lowest (highest) value found for that measure. This means that the minimum (maximum) values for the different measures do not necessary originate from the same scenario.

There are two settings: (i) whether or not balancing constraints ((2.18)-(2.19)) were used in the case of a partial blockage, and (ii) how much delay (0, 3, 5 or 10 minutes) is allowed for trains that are not running at the start of the disruption. From now on, we refer with the term *allowed delays* only to the delays allowed for trains that are not running at the start of the disruption.

It turned out that for only 2 of the 1530 scenarios it was not possible to find a feasible solution matching all constraints in our model.

Table 2.1 gives the results on the number of cancelled trains, the number of partially cancelled trains, the number of cancelled minutes and the computation times. In order to make the results maximally comparable, we count the number of the cancelled and partially cancelled trains, and the number of cancelled minutes as follows. A train t is partially cancelled if the disruption was on its route, and if only one of the trains t_α or t_γ (the parts before and after the disruption) is operated, while the other one is cancelled. If only the part on the disruption (train t_β) is cancelled, this is not considered as a (partially) cancelled train, since this part has to be cancelled inevitably in case of a complete blockage. The number of cancelled minutes is the sum of the scheduled lengths in minutes of the cancelled parts of all cancelled, partially cancelled, and inevitably cancelled trains, which is the main part in the objective function. Since the aim is to minimize the cancelled minutes and not to minimize the number of cancelled trains, it may happen that there are solutions where more trains (with a short duration) need to be cancelled to have less cancelled minutes.

For each instance, the total number of trains is approximately 180, since we compute a schedule for 3 hours in which more than 60 trains per hour run.

In case of a complete blockage, as shown in Table 2.1, if no delays are allowed, on average an amount of 405 minutes (almost 7 hours) of train service is cancelled. Allowing up to 10 minutes of delay, reduces the amount of cancelled minutes by more than 8%.

If only one track is blocked, we see that the effect of allowing more delay is much higher. Allowing 10 minutes of delay decreases the number of cancelled minutes on average with more than 50% (from 156 to 68 minutes). The results also demonstrate that in some parts of the network it is still possible to run all trains if only one track is out of service. This may be deduced from the fact that there are solutions with 0 cancelled minutes.

The price of forcing the new schedule to be a regular one by including balancing constraints, in which the number of trains per train series in one direction differs at most one from the number of trains in the other direction, is an increase of around 10% in the number of cancelled minutes.

The largest computation time is less than 1.5 minutes in case a maximum delay up to 5 minutes is allowed. This means that our approach is able to find advanced timetables with a high probability of having a feasible rolling stock schedule within a computation time which is reasonable in practice.

We can even find better solutions by allowing more delays (up to 10 minutes). For the complete blockage case, the computation times are still below 1.5 minutes then. However, in case of a single track blockage, allowing delays for trains up to 10 minutes can increase the computation time to values which are not usable for practice. However, on average the computation

	Minutes of delay allowed	*Delayed trains*			*Delayed events*			*Total maximum delay*			*Total delay*		
		Min	*Avg*	*Max*	*Min*	*Avg*	*Max*	*Min*	*Avg*	*Max*	*Min*	*Avg*	*Max*
Complete blockage (808 instances)	*0*	0	0.1	2	0	0.2	15	0	1	35	0	2	226
	3	0	1.4	18	0	6	76	0	3	44	0	11	169
	5	0	2.1	23	0	10	129	0	6	61	0	24	331
	10	0	2.8	19	0	13	128	0	11	104	0	49	838
One track blocked no balancing (720 instances)	*0*	0	0.1	2	0	0.8	19	0	2	28	0	11	388
	3	0	3.4	20	0	19	100	0	7	38	0	39	266
	5	0	5.4	27	0	34	188	0	14	83	0	86	720
	10	0	9.1	38	0	53	266	0	39	216	0	224	1452
One track blocked with balancing (720 instances)	*0*	0	0.1	2	0	0.7	17	0	1	27	0	8	387
	3	0	3.0	20	0	17	91	0	6	35	0	36	368
	5	0	4.9	20	0	30	141	0	13	67	0	76	396
	10	0	8.3	33	0	49	233	0	34	170	0	201	1190

Table 2.2: Results on delays experienced

time is still relatively low and in both variants (with and without balancing constraints) less than 9 out of the 720 cases had a computation time larger than 5 minutes.

For the cases with a computation time larger than 5 minutes we have computed the results with a maximum CPLEX CPU time of 5 minutes. In case of no balancing constraints, on average the number of cancelled minutes was 16% higher than in the optimal solution computed without the time limit. The maximum increase in cancelled minutes was 50%. In cases with balancing constraints the average increase in cancelled minutes was 25% and the maximum increase was 65%.

The decision on which allowed delay should be preferred can be taken by the dispatchers, also based on the available computing time for obtaining the new timetable. A good trade-off between solution quality and computing time corresponds to an allowed delay of 5 minutes.

Another strategy could be to first find the solution for the problem in which a delay of 5 minutes is allowed and then use the found solution as a feasible start solution for the problem in which 10 minutes of delay is allowed. In this strategy the dispatchers can give the approach (a maximum) time to improve the solution found with the setting of a maximum delay of 5 minutes.

Table 2.2 gives the results on the number of delayed trains, the number of delayed events, the sum of the maximum delay experienced by each train on its route (*Total maximum delay*), as well as the sum of the delays of all events. As expected, allowing delays increases on average the total delay of the trains and events, but, as can be seen in Table 2.1, it also reduces the number of cancelled trains.

2.5.6 Results with the option to reroute trains

Table 2.3 presents the results of the 60 scenarios in which there is a rerouting option for intercity trains. In the case of a complete disruption there was one instance which was not feasible for all settings. Therefore, we removed this instance from the results. This leaves us with 29 scenarios in case all tracks are blocked and 30 scenarios in case only one track is blocked.

One train series is allowed to be rerouted. Since this train series runs twice an hour in each direction and since we are dealing with a 2 hour disruption, the total number of trains which can be rerouted is equal to $2 \times 2 \times 2 = 8$.

In Table 2.3 we compare the results in which rerouting is not allowed with the results of the same instances in which rerouting is allowed. We can see that, in case of complete blockages, our approach is able to reduce the number of cancelled minutes considerably (approximately 20%) if we allow reroutings. Computation times stay below 2 minutes. In addition, having the option to reroute trains reduces the number of cancelled minutes much more than simply allowing larger maximum delays.

If all tracks are blocked, we originally did not have to include the balancing constraints since, in that case, no trains at all run over the blocked area. However, in case we allow reroutings, it is worthwhile to include balancing constraints for the rerouted trains to ensure that trains are rerouted evenly in each direction. The results demonstrate that in the case of a complete blockage with reroutings these balancing constraints do not have much influence on the solution and on the computation time. Therefore we recommend to include these balancing constraints. This is due to the fact that it seems to be possible to reroute all trains in many cases. If all trains are rerouted, then the result is automatically a balanced solution with 4 rerouted trains in each direction.

Also in case only one of the tracks is blocked, allowing reroutings can reduce the number of cancelled minutes considerably (up to 38%). However, if delays of 10 minutes are allowed, we discover that our approach without reroutings is able to find solutions with the same number of cancelled minutes. In case rerouting is allowed, the reroutings are used to reduce the delays of the trains.

In case of a blockage of only one track, the balancing constraints have a larger effect than in case of a complete blockage. Especially if we allow delays of 5 or 10 minutes, adding the

	Minutes of delay allowed	Rerouted trains			Cancelled minutes			Computation time (s)		
		Min	Avg	Max	Min	Avg	Max	Min	Avg	Max
Complete blockage no rerouting (29 instances)	0				497	521	544	3	3.0	4
	3				469	512	536	4	4.9	6
	5				469	506	536	5	5.6	8
	10				451	488	525	7	10.1	16
Complete blockage with rerouting no balancing (29 instances)	0	4	7.4	8	379	419	489	6	6.8	7
	3	4	7.7	8	375	410	462	10	11.5	20
	5	4	7.7	8	351	405	462	11	14.6	25
	10	4	7.7	8	335	391	443	19	37.2	75
Complete blockage with rerouting with balancing (29 instances)	0	1	7.5	8	379	423	543	6	7.0	11
	3	1	7.5	8	375	414	517	10	11.6	20
	5	1	7.5	8	351	409	517	12	15.7	43
	10	8	8	8	335	393	468	20	39.9	81
One track blocked no rerouting no balancing (30 instances)	0				256	274	304	3	5.4	10
	3				193	221	251	5	7.8	29
	5				151	168	189	7	14.6	39
	10				60	88	96	19	71.8	284
One track blocked with rerouting no balancing (30 instances)	0	6	6.8	7	196	217	243	8	15.1	25
	3	4	4.0	4	178	193	218	17	23.8	40
	5	3	4.0	4	96	103	124	15	42.9	109
	10	5	6.8	8	60	88	96	48	170.9	407
One track blocked no rerouting with balancing (30 instances)	0				288	311	334	5	7.2	11
	3				243	283	326	8	16.8	49
	5				243	257	307	13	27.8	55
	10				94	129	153	30	144.4	396
One track blocked with rerouting with balancing (30 instances)	0	1	6.6	7	206	238	326	9	17.8	55
	3	4	4.0	4	180	193	218	20	32.1	60
	5	4	5.1	8	163	182	218	30	74.0	141
	10	5	5.1	6	94	121	137	165	284.7	830

Table 2.3: Results with reroutings

balancing constraints leads to an increase of about 50% in the number of cancelled minutes, independently of whether or not reroutings were allowed.

Also in case reroutings are allowed, our approach is able to quickly find an optimal solution, as long as we do not allow more than 5 minutes delay. If dispatchers want to allow larger delays (10 minutes) the computation times increase, especially in the case with balancing constraints.

2.6 Conclusions

In this paper we introduced an Integer Linear Program (ILP) to solve the real-time railway timetable rescheduling problem for a railway network. The railway timetable rescheduling problem considered in this paper has a macroscopic view on the infrastructure network which consists of stations and open track sections with certain numbers of tracks. Furthermore, constraints on the available rolling stock are also considered in order to have a high probability that there is a feasible rolling stock schedule for the new timetable. The possibility of rerouting trains in order to reduce the number of cancelled and delayed trains is also considered. In addition, all stages of the disruption management process (from the start of the disruption to the time the normal situation is restored) are taken into account.

The ILP is modeled as an event activity network in which each event represents an arrival or a departure of a train, and in which an activity refers to passing on a resource unit from one event to another event. The resources considered are the tracks in the open track sections, the tracks in the stations, and the rolling stock compositions.

Computational tests are performed on a heavily used part of the Dutch railway network. Solutions are provided within computation times which are very well suitable for use in practice. Most of our cases can be solved within 1.5 minutes of computation time. The results show that a smaller number of trains needs to be cancelled and the number of cancelled minutes is significantly reduced if we allow to slightly delay or to reroute some trains.

Our approach turns out to be able to handle, in short computing time, every state of the network at the time the disruption starts, and to decrease cancellations and delays of trains. This makes our approach much more flexible and efficient than the current practice of using contingency plans.

Chapter 3

Passenger Oriented Disruption Management in Railways by Adapting Timetables and Rolling Stock Schedules

This chapter considers the paper (**Veelenturf** et al. (2014b)) presented at the PATAT 2014 conference. It is under review at *Annals of Operations Research*. A preliminary version of this paper won in 2010 the Junior of the Year Award of research school TRAIL.

Co-authors: L.G. Kroon and G. Maróti

3.1 Introduction

In passenger railway operations, unforeseen events (such as infrastructure malfunctions, accidents or rolling stock breakdowns) can make parts of the railway infrastructure temporarily unavailable. Then it is not possible to operate the timetable, rolling stock schedule and crew schedule as planned. Within minutes, or even better, seconds, a new timetable and new resource (rolling stock and crew) schedules must be available. In Cacchiani et al. (2014) an overview is given of recovering models and algorithms to solve these rescheduling steps. In this overview it becomes clear that, although the schedules are interdependent, most research focuses on rescheduling one of the schedules at a time. By the complexity of the rescheduling problems, there is not enough time to solve the integrated problem. In this chapter we partly integrate the rescheduling of the rolling stock plan and the timetable. Our particular focus lies on passenger service, and we take passenger behavior explicitly into account.

Current literature on integrated rescheduling of the timetable and the rolling stock schedule is scarce. Adenso-Díaz et al. (1999) and Cadarso et al. (2013) applied research on integrated rescheduling of the timetable and rolling stock on cases of the Spanish railway operator RENFE.

Like the main focus of this chapter, Cadarso et al. (2013) take the dynamics of the passenger behavior during a disruption into account. In the current paper, however, the fundaments of the approach are from Kroon et al. (2014). In Kroon et al. (2014) the focus is on improving passenger service by considering passenger behavior while rescheduling the rolling stock. Kroon et al. (2014) use an iterative procedure for rescheduling the rolling stock and evaluating the resulting passenger behavior which is inspired by the iterative framework of Dumas and Soumis (2008). Changing the timetable can also improve the passenger service. Therefore we extend the approach of Kroon et al. (2014) by allowing the timetable to be slightly adapted as well.

It is important to focus on the passenger service since a disruption does not only affect the timetable and the resource schedules, but also the passengers. However, for railway operators without a seat reservation system it is difficult to reschedule the passengers. The passengers will make their new travel plan by themselves. If they had planned to take a train which is canceled due to the disruption, they will decide not to travel or to reroute themselves. Rerouting of passengers means that they take other trains to their destination than originally planned. This does not necessarily require the passengers to take a detour: They can also take a later train on the same line.

By the changed passenger flows, the disruption causes changes in the demand for seats. Therefore, a rolling stock rescheduling approach to handle a disruption must take the modified passenger flows into account dynamically, and not the passenger flows of a regular day. For example, since some passengers will take a detour, additional capacity on the detour routes is necessary. One way to handle this is to increase the capacity of the trains on this route. Another option is to increase the capacity by inserting more train services or by letting trains make additional stops.

The consequences of the timetable adaptations may not turn out to be advantageous for all passengers. For example, an additional stop of a train will delay the train with a few minutes. As a consequence, the original passengers of the train will get an additional delay in favor of reducing the delay of the passengers at the station at which an additional stop is made. The small delay of the train can even lead to a large delay for the passengers if they miss their transfer at a later station. The railway operator has to make trade-offs between the different consequences for the passengers.

In this research we limit the timetable decisions to adapting the stopping patterns. Other timetable decisions to influence the passenger flows, for example by inserting additional trains, are left out since an additional train requires the railway operator also to adapt the crew schedules. We assume that small delays caused by adapting the stopping patterns can be absorbed by the buffers within the crew schedule.

Delay management, which consists of deciding on whether or not trains have to wait for delayed connecting trains, is another problem in which slight timetable adaptations influence the passenger flows. Delay management is a hard problem on its own and thereby not considered in our approach. We refer the interested reader to Schachtebeck and Schöbel (2010), Kanai et al. (2011) and Dollevoet et al. (2012) for recent works on delay management approaches.

The framework of our rolling stock rescheduling approach is discussed in Section 3.2. In Section 3.3 we show a relaxation of the model discussed in Section 3.2. Instead of solving the relaxation, we make use of an iterative procedure of which in Section 3.4 up to Section 3.7 the components are explained. Results of different variants of our approach, based on a scenario in the Netherlands are discussed in Section 3.8, and Section 3.9 concludes this paper.

3.2 Rolling stock and timetable rescheduling with dynamic passenger flows

The performance of the disruption management process investigated in this paper arises from the interaction of 3 factors: (i) the timetable, (ii) the rolling stock schedule (seat capacity), and (iii) the passenger behavior.

We consider disruptions where passenger behavior has a large impact on the performance of the railway system if the timetable and rolling stock schedule are not changed. Examples are disruptions where certain tracks are blocked for a number of hours. Passengers react to these disruptions by finding alternative routes to their destinations. However, the capacity on these alternative routes can be limited, resulting in overcrowded trains and thereby longer dwell times and delays.

Two ways to handle the increased demand on the alternative routes are to enlarge the capacity of the trains and to adapt the timetable. Adapting the capacity of the trains alone is not always enough. For example, it can be impossible to increase the capacity of a train by lack of time and/or reserve rolling stock or due to limited platform lengths. Therefore, timetable adaptations such as adding extra train services, rerouting trains or adding extra stops for trains are worthwhile as well.

By adapting the timetable, the railway operator can influence the passenger flows by providing new alternative travel routes and by influencing the demand for certain trains. For example, a train can make an additional stop at a station to give passengers at that station an additional, earlier, travel option to their destination and to decrease the demand for the next train stopping at that station and travelling in the same direction.

In this research we limit the timetable changes to adaptations of the stopping patterns of trains. A stopping pattern of a train indicates the stations where the train makes a stop. A *stopping pattern* can contain, next to the scheduled stops, also new stops at stations where the train did not have a scheduled stop.

To make a new rolling stock schedule and timetable for the remainder of the day, we assume that the complete characteristics of the disruption are revealed at the moment the disruption starts. For example, at that time, the exact duration of the disruption is known.

Then, a general framework for rescheduling the rolling stock and timetable by considering the passenger behavior based on the model of Kroon et al. (2014), with the difference that now also decision variables for the timetable decisions are included, can be stated as follows:

$$\min c(x) + d(y) + e(z) \tag{3.1}$$

$$\text{subject to } z \in \mathcal{Z} \tag{3.2}$$

$$x \in \mathcal{X}_z \tag{3.3}$$

$$y = f(x, z) \in \mathcal{Y} \tag{3.4}$$

Here $\mathcal{Z}$ is the set of all possible timetables given the disruption, $\mathcal{X}_z$ is the set of all possible rolling stock schedules matching with timetable z, and $\mathcal{Y}$ is the set of feasible passenger flows. The function $f(x, z)$ returns the emerging passenger flows for a given timetable $z \in \mathcal{Z}$ and rolling stock schedule $x \in \mathcal{X}_z$. Note that the chosen timetable z and rolling stock schedule x uniquely determine the passenger flows y by the function f. This means that the only real decision variables are the rolling stock schedule x and the timetable z.

The objective function consists of three terms. The function $c(x)$ gives the system related costs of a rolling stock schedule, which can also be seen as the rolling stock rescheduling costs. The function $d(y)$ gives the service related costs of the passenger flows. The function $e(z)$ gives the system related costs of a timetable, so the timetable rescheduling costs. The highest priority is given to assigning at least one rolling stock unit to each train, to prevent the train to be cancelled by lack of rolling stock. Such cancellations will not only have a large negative influence on the passenger flows, but also make the crew schedule infeasible.

3.2.1 Iterative Procedure

The optimization model (3.1)-(3.4) is very difficult to solve directly, mainly due to the complex structure of the objective function f. We are not aware of any algorithmic framework that

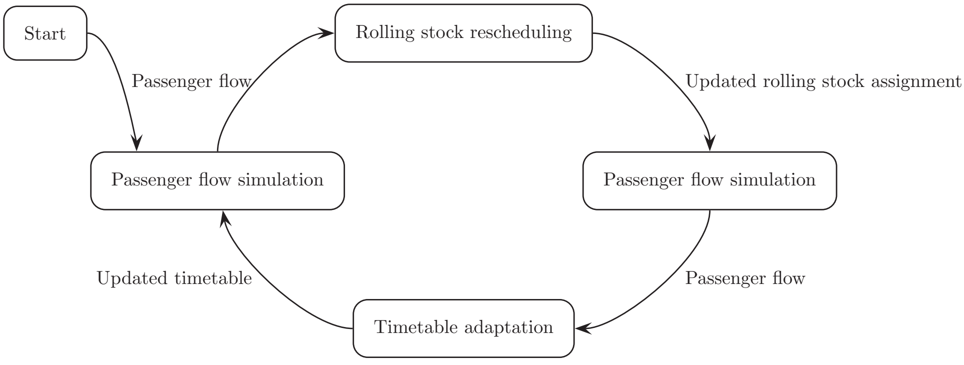

Figure 3.1: Iterative procedure for solving the rolling stock rescheduling problem with dynamic passenger flows.

would be able to handle realistic instances of (3.1)-(3.4). Therefore we propose an extension to the iterative heuristic of Dumas and Soumis (2008) and Kroon et al. (2014); the approach is sketched in Figure 3.1.

The input of our algorithm consists of the original (i.e., undisrupted) timetable, the original rolling stock schedule as well as a list of train services that must be cancelled as an immediate reaction to the disruption. The removal of these inevitably cancelled services gives the initially modified timetable.

In each iteration, we evaluate the passenger flows by using a simulation algorithm. The simulation is based on the previous iteration's timetable and rolling stock schedule. Here the rolling stock schedule is only needed because it determines the capacities of the trains. We use the simulation model introduced by Kroon et al. (2014). The details of this simulation model are summarized in Section 3.4. Note that the first iteration uses the initially modified timetable, and assumes that each train has the same capacity as in the original schedule.

The passenger simulation pinpoints the trains with insufficient capacities. The rolling stock rescheduling model computes a new schedule based on these findings, balancing it with other criteria, such as operational costs. For details we refer to Section 3.5.

After another round of passenger simulation, we evaluate which adaptations of the timetable could potentially improve the service quality. Each individual adaptation is a minor change, such as requiring a train to make an extra stop. Therefore we can assume that the just computed rolling stock schedule remains feasible. We describe several variants for finding the most promising timetable adaptation in Section 3.6. Having decided on the timetable, the next iteration will start by launching a passenger simulation.

Our method differs from the framework of Kroon et al. (2014) by adding the timetable adaptation step to the loop. Since the passenger flows can be heavily impacted both by a new rolling stock schedule and by an adapted timetable, we carry out passenger simulations after each of them.

The iterative approach is purely heuristic; it does not necessarily converge, and has no optimality guarantee. Motivated by the limited time in real-life applications, we terminate our algorithm after a certain number of iterations, and we report the best solution found. In addition, we compute lower bounds, described in Section 3.7, in order to be able to judge the quality of the solution.

3.3 Operator control

The rescheduling process may result in a better outcome if the operator can directly influence the passengers' behavior by appropriately assigning them to the train services (rather than letting the passengers choose their routes). We call this situation *operator control*. In this section we describe an optimization model for operator control which is a relaxation of the model (3.1)-(3.4). We are not going to use this model in our computational tests, since it is a computationally large model and we do not want to assume operator control. However, we still want to present this relaxation of the model to give an idea about its complexity, and thereby justifying our use of an iterative procedure.

We split all timetable services into trips $t \in \mathcal{T}$ representing a movement of a train between two consecutive planned stops. The main decision for the rolling stock schedule is to assign *compositions* to trips, where a composition consists of one or more combined train units. Let $\mathcal{G}_t$ be the set of all compositions g which can be assigned to trip t, and the capacity of composition g is denoted by Cap_g. Binary variables $x_{t,g}$ indicate whether composition g is used ($x_{t,g} = 1$) for trip t or not ($x_{t,g} = 0$).

For the timetable decisions every trip $t \in \mathcal{T}$ has a set $\mathcal{J}_t$ of possible stopping patterns for stops at the intermediate stations. Here a stopping pattern indicates a sequence of intermediate stations at which the train makes an additional stop. Binary variables z_j indicate whether stopping pattern j is used ($z_j = 1$) or not ($z_j = 0$).

A passenger $p \in \mathcal{P}$ should take a path from its origin to its destination within his/her proposed deadline, where a path itself is a sequence of rides on trains between two stations. Let K^p be the set of all paths that passenger $p \in \mathcal{P}$ could take and let $K_t^p \subset K^p$ be all paths including (part of) trip t which passenger p could take. Note that the paths in K^p and K_t^p can be based on every possible stopping pattern. Let $\bar{\mathcal{J}}_k$ be the set of all stopping patterns j matching with path k. The binary variable y_p^k is 1 if passenger p picks path k and 0 otherwise. The parameter d_p^k indicates the associated cost (delay) of passenger p taking path k.

Then, in case of operator control, the model of (3.1)-(3.4) can be relaxed by:

$$\min \qquad c(x) + \sum_{p \in \mathcal{P}} \sum_{k \in K^p} y_k^p d_k^p + e(z) \tag{3.5}$$

$$\begin{aligned}
s.t. \qquad & x \in \bar{\mathcal{X}}_z & & (3.6)\\
& \sum_{j \in \mathcal{J}_t} z_j = 1 & \forall t \in \mathcal{T} \quad & (3.7)\\
& \sum_{k \in K^p} y_k^p = 1 & \forall p \in \mathcal{P} \quad & (3.8)\\
& y_k^p - z_j \leq 0 & \forall p \in \mathcal{P}, \forall k \in K^p \text{ and } \forall j \in \bar{\mathcal{J}}_k \quad & (3.9)\\
& \sum_{p \in \mathcal{P}} \sum_{k \in K_t^p} y_k^p \leq \sum_{g \in \mathcal{G}_t} x_{t,g} Cap_g & \forall t \in \mathcal{T} \quad & (3.10)\\
& y_k^p \in \{0,1\} & \forall p \in \mathcal{P} \text{ and } \forall k \in K^p \quad & (3.11)\\
& z_j \in \{0,1\} & \forall t \in \mathcal{T} \text{ and } \forall j \in \mathcal{J}_t \quad & (3.12)
\end{aligned}$$

The objective function (3.5) is to minimize the total costs of the rolling stock rescheduling, the passenger flows (sum of delays) and the timetable rescheduling. Constraints (3.6) compactly summarize the constraints on the underlying rolling stock rescheduling problem. These rolling stock decisions are influenced by the chosen timetable z since there are some minimum process times required in the rolling stock schedule. So, if some trips take longer than planned certain processes can become infeasible. Constraints (3.7) determine that for every trip exactly one stopping pattern must be selected. Every passenger must pick exactly one path, which is modeled by Constraints (3.8). Constraints (3.9) ensure that only matching paths and stopping patterns can be chosen. The chosen paths by the passengers should also match with the available capacity in the trains which is modeled by Constraints (3.10).

Even this relaxation of the model of (3.1)-(3.4) is a complex model to solve in a real-time environment by the interdependence between the rolling stock and the timetable via the passenger flows. Therefore, we will solve the model in (3.1)-(3.4) by an iterative procedure as discussed in Section 3.2.1.

3.4 Simulation of the passenger flows

The iterative procedure starts with a simulation of the passenger flows, and each time the timetable or rolling stock schedule is updated a new simulation of the passenger flows is necessary.

To keep the simulation tractable, all passengers with the same characteristics (origin, destination and arrival time at the origin) are aggregated into *passenger groups*.

To simulate the passenger flows we use the simulation algorithm as described in Kroon et al. (2014). It is important to mention that this is a deterministic simulation algorithm to calculate the emerging passenger flows. This means that, given a timetable and rolling stock schedule, there are uniquely defined resulting passenger flows. Here we shortly summarize the assumptions of the model as described in Kroon et al. (2014). For more details we refer to that paper. We emphasize that the approach is modular, which allows us to replace the simulation model by any other simulation model to model the passenger behavior.

3.4.1 Assumptions

For the simulation of passenger behavior, Kroon et al. (2014) make assumptions on three fundamental issues: (*i*) What information is available to the passengers? (*ii*) Which traveling strategy do passengers apply to the available information? and (*iii*) How do passengers interact?

Information available for the passengers

It is assumed that passengers always know the most recent timetable. This means that if the timetable is updated due to a disruption, they know which trains are canceled, which trains make additional stops, and which trains are delayed. Passengers do not know the future timetables, so they cannot anticipate on cancellations, delays and additional stops before the disruption occurs. Furthermore, they do not know anything about whether or not they fit in the trains they would like to take.

Strategy of the passengers

Each passenger has a *traveling strategy*. This strategy decides for the passenger what will be the preferred path to the destination given the most recent timetable. In Kroon et al. (2014) all passengers have the same strategy. In our research we also use this single strategy. The used strategy is that passengers want to reach their destination as early as possible. If several paths have the same earliest arrival time, the passengers prefer the path with the least transfers between trains. If we have multiple paths with the same earliest arrival time and the same minimum number of transfers, the passengers will take the path with the earliest departure time. It is worthwhile to mention that in practice there is a more balanced trade off between transfers and travel time. It seems to be highly unrealistic that passengers are willing to transfer 2 times to save 1 minute of travel time. Note that one could easily include other strategies as well.

Each passenger wants to reach his destination before a certain *deadline*. If a passenger is not able to reach his destination before the deadline, it is assumed that the passenger gives up travelling by train. In this way it is modeled that passengers are not willing to wait endlessly.

Interaction between the passengers

If a train arrives, first the passengers who want to leave the train get the option to do so, then the passengers who wait at the platform and want to enter the train compete for the available capacity in the train. It can happen that there is not enough capacity for all passengers. Then it is assumed that the number of passengers from each passenger group who actually board a train is relative to the size of the group. This could lead to a fractional number of passengers but it is assumed that the contribution of fractional flows are neglectable.

It is possible that not all members of a passenger group are able to board the train: Some of them have to stay behind. We say that these passengers are *rejected* by the train. In case of rejections, the passenger group is split into two: those passenger who were able to board the train and those who were not. The rejected passengers must find a new preferred path from their current location, while the boarded passengers can just follow their previously computed preferred path.

3.4.2 Evaluating the passenger flows

In this research we evaluate the passenger flows by the delays which passengers face in comparison with their original expected arrival times and by the number of passengers who gave up traveling by train since they ware not able to reach their destination within their set deadline. In our experiments we try different ways to penalize delay minutes. In one setting the delay minutes are penalized uniformly and in another setting longer delays are penalized more, since one may argue that longer delays are worse than several small delays. For passengers who are not able to reach their destination within their deadline we penalize passengers leaving the system by the difference between the deadline and the expected arrival time of the intended traveling path. For each passenger, his delay or penalty for not reaching his destination within his deadline is called his *inconvenience*.

3.5 Rolling stock rescheduling

The rescheduling of rolling stock follows the procedure of Kroon et al. (2014). In this procedure the rolling stock is rescheduled based on the model described in Nielsen et al. (2012) (which is an extension of Fioole et al. (2006)). The basic decisions in the model are to assign a rolling stock composition to each trip such that as many of the passengers are accommodated. The difficulty of the rolling stock rescheduling is that a composition consists of multiple combined train units.

During operations the operator can change the compositions by decoupling or coupling units in the front or the back of the train. These operations are called *shunting operations*. Shunting personnel must be arranged to facilitate these operations. Therefore, changing the shunting operations also includes new tasks for the shunting personnel, which is not preferred.

Since a composition can consist of different types of train units, the order in which they are combined within a composition matters (i.e. one could not decouple a unit in the middle).

As discussed, the main objective of our approach is to prevent cancellations caused by lack of rolling stock. Therefore we first determine how many trains need to be cancelled due to lack of rollings stock. To do this we run the rolling stock rescheduling approach on the initially modified timetable with the single goal to find a feasible rolling stock schedule by minimizing the number of trains without rolling stock. This means that we have only a penalty for trains which do no get rolling stock assigned to them. All other penalties are set equal to 0. The result shows how many trains need to be cancelled inevitably by lack of rolling stock. In the rolling stock rescheduling steps we enforce the number of cancelled trains to be equal to this value to ensure that no more trains than necessary are cancelled. Still the rolling stock rescheduling approach has freedom in which trains it does not assign rolling stock to.

For all remaining rolling stock rescheduling steps, the model has two objectives: It consists of a trade of between minimizing the rolling stock rescheduling costs and the inconvenience for the passengers. The rolling stock rescheduling costs are mostly based on how much the rolling stock schedule is changed. For example one does not want to make too many new shunting operations, since these new shunting operations must be communicated (with a certain probability of miscommunication) and require personnel to perform them.

The inconvenience for passengers is based on the latest simulation run with the timetable and rolling stock schedule of the last iteration. Penalties are defined for assigning a certain composition to a trip. The penalties are determined by estimating the effect of the train capacities on the total passenger inconvenience measured as discussed in Section 3.4.2.

Per trip the average inconvenience per passenger who was not able to board the train is computed. To do this, per passenger group the difference in inconvenience between passengers who were not able to board and passengers who were able to board is determined. Then, the weighted (based on group size) average of these differences is considered as the average inconvenience per passenger who is not able to board the train. In the objective function the number of seat shortages is multiplied by this average inconvenience per rejected passenger. For more details we refer to Kroon et al. (2014).

Kroon et al. (2014) reported that the approach of updating the objective function could lead to cyclical behavior if the feedback from earlier iterations is ignored. We follow the described exponential smoothing procedure in Kroon et al. (2014) (which is based on Dumas and Soumis (2008)) to take feedback from earlier iterations into account as well. We use the setting which performed best in their case. This setting means that feedback from earlier iterations is weighted for 35 percent.

3.6 Timetable adaptations

The disruption management process admits timetable decisions in order to better facilitate the passenger flows. In this paper we limit the allowed timetable modifications to adding stops to timetable services.

In this paper, adapting the stopping patterns means that trains may stop at stations where they normally just pass through. Making an additional stop results in new traveling options for some passengers but also in an increased travel time for others. Therefore it is necessary to make a trade off between the positive and negative effects of the changed stopping pattern. The objective of this research is to minimize the sum of the delays of all passengers. Therefore, we only allow timetable changes that do not increase the total delay of all passengers. We assume that an additional stop will delay all further trips of a train by a fixed number of minutes and that those delays will not influence other train traffic.

The (greedy) procedure to adapt the stopping patterns goes as follows: (i) we have a list of candidate timetable adaptations, (ii) we evaluate for each candidate the consequences, (iii) we apply the timetable adaptation with the most positive consequences.

First of all, this approach requires that in step (i) a list of candidate timetable adaptations is given. The dispatchers can give this as input to the approach. In the extreme case, every timetable service is allowed to make an additional stop at every station it passes.

The effect measured in step (ii) indicates how much the total delay of the passengers will change if only that single timetable adaptation will be applied. Therefore, in step (iii) we limit ourselves to allow only one timetable adaptation per iteration of the solution approach. If no candidate timetable adaptation reduces the total delay of the passengers, no timetable change is made.

For all candidate timetable adaptations, the consequences of applying the adaptation need to be computed. In Sections 3.6.1-3.6.4 we discuss several methods and approximations to compute these consequences. In Section 3.6.1 we discuss a method to compute the exact effect of the additional stop. The exact effect can be computed since we use a deterministic simulation for the passenger flows. However, computing the exact effect can be time consuming. Therefore, we also suggest a faster approximation algorithm in Section 3.6.2. Furthermore, we introduce two heuristics in Sections 3.6.3 and 3.6.4 which are more transparent for use in practice.

In the different variants we evaluate the effects of different candidate timetable adaptations. We refer to train i as the train of which the stopping pattern will be adapted within the candidate timetable adaptation. Furthermore, the station at which train i will make the additional stop within the candidate timetable adaptation is called station b.

3.6.1 Exact effect of an additional stop (*EXACT*)

To determine the exact consequences of an additional stop, we need to run the simulation algorithm which is discussed in Section 3.4 twice. First the simulation is performed with the current timetable and then the simulation is performed with the timetable which results from the timetable adaptation. From both simulations we get the total delay minutes of the passengers. The difference between these two total delay minutes shows the consequences of the candidate timetable adaptation. This approach measures the exact effect of the additional stop and is called *EXACT*.

A variant of *EXACT*, denoted by *EXACT**, will not use the current capacity of train i in the simulation but the capacity of the largest possible composition allowed for train i. The difference between this simulation and the simulation of the current timetable will then not measure the exact effect of the extra stop but the potential (which can be larger) effect of the additional stop. In this case it is left to the rolling stock rescheduling phase to check whether it is possible to increase the capacity of train i.

3.6.2 Estimated effect of an additional stop (*EST*)

In this section we introduce an approach to *estimate* the effect of an additional stop. In this variant, both the positive and the negative effects of the additional stop are considered. For all passengers their preferred path to their destination is known. If we change the timetable by including an additional stop, some passengers might get another preferred path to their destination.

Some passengers arrive earlier at their destination due to an additional stop at their origin or destination. Other passengers might profit from the delay of the train caused by the additional stop, since due to the delay they were able to catch this train which departs earlier than the train they were intending to take. All these passengers originally did not have train i on their preferred path, but in case the additional stop is executed they have train i on their preferred path.

However, since the additional stop takes some time it also causes some delays for passengers who had train i on their preferred path in the current timetable. Due to the delay of train i it can be that their preferred path changes. Also if the preferred path does not change it can still mean that the passengers are delayed if the trip on train i was the final trip in their path.

To estimate the effect of the additional stop we compute for each passenger his preferred path in the current timetable and the preferred path in the timetable in which train i makes an additional stop. The sum of all these differences is our estimation of the consequences of the additional stop. Note that this is an estimation since this method assumes that every passenger can take his preferred path which might not be true by the limited capacity of the rolling stock compositions.

We make variants of this approach by assuming different durations of the additional stop in the determination of the shortest path. This means that we can use for the estimation of the effect another duration of the additional stop than the real duration of the additional stop. For example, if we assume a shorter duration of the additional stop, we over-estimate the potential effect of the additional stop. However, maybe this will lead to a good optimization direction. The duration of the extra stop used in the estimation approach will be called the *extra stop penalty*.

3.6.3 Rule of thumb: Do not pass passengers who did not fit in a previous train (*PRACT1*)

If passengers did not fit in a train, then they have to wait for the next train in the same direction. Especially for these passengers, since they have already been rejected, it is very frustrating if a train in their direction passes them without stopping. A rule in practice could be that it is not allowed to pass a group of passengers who were rejected by a previous train. This is an easy to use rule of thumb. We will evaluate the performance of this rule which is called $PRACT1$.

To decide whether or not train i needs to make an additional stop we have to consider two other trains. The first considered train h is the last train which arrived at station b before the passing time of train i at station b. If there were passengers with destination b rejected to board train h at the departure from the last station a before station b, train i will make an additional stop at station b to let these rejected passengers travel from station a to b.

The second considered train j (which is in most cases the same as train h), is the last train which departed from station b in the same direction as train i before train i passes station b. If there were passengers rejected to board train j at station b, train i will make an additional stop at station b to let these rejected passengers enter the train.

Since we allow one timetable adaptation per iteration we have to make a comparison on how effective the additional stop will be: Therefore, we sum up the advantages for all passengers who were rejected to board train h in station a or train j in station b. For the passengers rejected to board train h in station a the advantage is measured by the difference between the arrival time of train i at station b and the arrival time of the first train from a to b after train i. For the passengers rejected to board train j in station b the advantage is estimated by the difference between the arrival time of train i at the first station after station b where both trains h and i stop and the arrival time at the same station of the first train departing after train i from station b.

In this measurement we assume that all passengers are able to board train i, so that train i is assumed to have infinite capacity. We do not use the actual capacity since in the rolling stock rescheduling phase the capacity of train i could be increased.

3.6.4 Rule of thumb: including the negative effects *PRACT2*

The approach *PRACT1* based on a rule of thumb only considers the positive effects of an additional stop. This results in a situation that even if only one passenger may profit from the additional stop, the stop will be executed. In another rule of thumb, *PRACT2*, the delay for passengers traveling by train i caused by the additional stop is included. In this approach, the advantages for passengers are measured in the same way as approach *PRACT1*, and the inconvenience per passenger who travels by train i at the moment train i makes an additional stop at station b will be equal to a fixed parameter. This parameter is equal to the duration of the extra stop plus a possible penalty. Note that for practitioners this rule requires more knowledge. In *PRACT1* the dispatchers only need monitor whether there are trains where some passengers did not fit in the train. In *PRACT2* the dispatchers also need to know how many passengers did not fit in the train, and how many passengers are in the next train passing station b.

3.7 Lower bound

The proposed approach does not guarantee to converge to an optimal solution. To consider the quality of our solutions, we check the gap between a lower bound and the value of our solution. Depending on the nature of the disruption, the lower bound on the rolling stock rescheduling costs will not differ that much from 0, but the lower bound on the passenger delays can be quite interesting. In this section we come up with a lower bound which takes the positive effects of an additional stop into account.

This lower bound can be reached by assuming infinite capacity on all trains, together with assuming that all extra stops are executed and that an extra stop does not cause any arrival delay.

To be more precise, in this lower bound all extra stops are executed and the departure times at stations after the additional stop are delayed by the time an extra stop will take, and all arrival times are kept the same. This way we ensure that no passenger faces an arrival delay caused by the additional stop and that passengers who may profit from a delayed train caused by an additional stop still have the opportunity to enter the train. Then a simulation run with infinite capacity on the trains and with the timetable as described above gives a lower bound on the total passenger delay.

It can happen that passengers have an advantage by a delayed train since they can pick a train earlier than their planned train. This must be considered in the lower bound. Therefore we cannot just add the extra stops and leave all departure and arrival times the same.

By delaying the departure times we have a lower bound which is valid for both cases, with and without the extra stop. No one gets an arrival delay, and some passengers arrive earlier since they have an extra travel opportunity by the extra stop.

This lower bound represents the delay of the passengers which the operator cannot prevent by increasing the rolling stock capacities or adapting the timetable. This delay is caused by the train services that are inevitably cancelled due to the unavailability of infrastructure caused by the disruption.

3.8 Computational Results

We tested the proposed approach on instances based on cases of Netherlands Railways (NS) which is the major railway operator in the Netherlands. In these instances, a disruption, due to some blocked switches, caused that fewer trains than normally can be operated on certain tracks.

3.8.1 Detailed case description

The instances take the busiest part the Dutch railway network into account which is represented in Figure 3.2. The original passenger flows are constructed conform a regular weekday of Netherlands Railways, which resulted in 15064 passenger groups with a total of about 450,000 passengers.

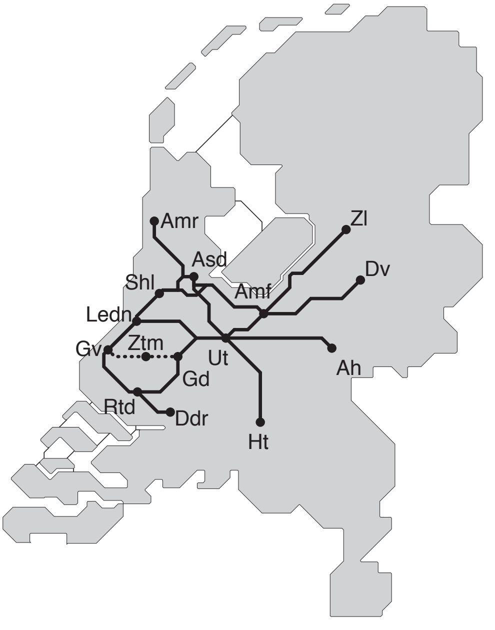

Figure 3.2: Part of the Dutch railway network

In almost all parts of the network we have four Intercity trains per hour in each direction. A *Intercity train* is a train which only stops at larger stations. All intercity trains in this network are considered, furthermore the *regional trains*, that stop at every station, between The Hague (Gv) and Utrecht (Ut) are also considered.

For the rolling stock rescheduling, four types of rolling stock are available; two types for regional trains and two types for intercity trains. The regional train types can be coupled together, which leads to 5 possible compositions, and the Intercity train types can also be coupled together in 10 different compositions.

In Figure 3.2 the dotted line represents the disrupted area. On those tracks on a normal day each hour 4 Intercity trains and 4 regional trains run in each direction. We constructed two instances with a disruption in the rush hours between 7:00 A.M. and 10:00 A.M. In the first instance (*ZTM1*), 2 regional trains per hour per direction are canceled. In the second instance (*ZTM2*) also 2 Intercity trains per hour per direction are canceled. This means that in instance *ZTM1* in each direction 6 trains per hour still run between Gouda (Gd) and The Hague (Gv), and only 4 trains per hour in each direction in instance *ZTM2*.

Table 3.1: Rolling stock rescheduling costs

Type of costs	value
New shunting operation	500
Changed shunting operation	500
Canceled shunting operation	100
Off balances at the end of the day, per unit	200
Seat shortage per seat per kilometer	0.1
Carriage Kilometers	0.0001

3.8.2 Parameter settings

The objective function consists of system related costs for the timetable adaptation and the rolling stock rescheduling, and costs for the passenger delays. For the timetable adaptations we do not consider any penalties other than that we assume that an additional stop will delay a train by 3 minutes.

The rolling stock rescheduling costs are given in Table 3.1. Most important is that the rolling stock schedule should not change too much from the original plan, since changed plans require communication between the dispatchers and the personnel, and a failure in this communication is easily made. Therefore we introduce costs for having other shunting operations than planned. Changing the shunting operations also includes new tasks for the shunting personnel, which is not preferred. We consider the carriage kilometers as least important.

The passenger service costs consist of the passenger delay minutes as discussed in Section 3.4.2, where the penalties for passengers who left the system because of not reaching their end station within their deadline are also measured in delay minutes.

The approach will make at maximum 15 iterations.

To solve the composition model of the rolling stock rescheduling we used CPLEX 12.5. The test instances are run on a laptop with a Intel(R) Core(TM) i7-3517U 1.9/2.4 Ghz and 4.0 GB RAM.

3.8.3 Results

This section provides the results of the two test instances. For the timetable rescheduling part we had different approaches to decide which Intercity trains should make an additional stop. We compare the effect of the different approaches on the final solution. We also compare our approach (which includes the option to adapt the timetable) with the method of Kroon et al. (2014) (which does not have an option to adapt the timetable) referred to as (*NO_STOP*). In the approach (*EST*) we estimated in the timetable rescheduling step the effect of an additional stop.

Table 3.2: Results

Solution Method	Lower bound	Objective	Passenger delay minutes	Rolling stock rescheduling costs	Extra stops	Iteration of best solution	Computation time (sec)
Instance *ZTM1*							
(EXACT)	33526	55927	55874	53	4	4	563
(EXACT*)	33526	55927	55874	53	4	9	573
(EST 0min)	33526	57372	56818	554	3	3	349
(EST 1min)	33526	58857	58304	554	4	4	352
(EST 2min)	33526	57372	57318	53	5	9	352
(EST 3min)	33526	91534	90980	554	0	1	291
(PRACT1)	33526	64304	64251	53	3	4	282
(PRACT2)	33526	64304	64251	53	3	4	307
(NO_STOP)	49626	91534	90980	554	-	1	235
Instance *ZTM2*							
(EXACT)	110848	139588	136527	3061	3	3	456
(EXACT*)	110848	139588	136527	3061	3	8	427
(EST 0min)	110848	139630	136368	3262	3	4	352
(EST 1min)	110848	139630	136368	3262	3	4	315
(EST 2min)	110848	162228	159167	3061	2	6	351
(EST 3min)	110848	177123	173861	3262	0	1	291
(PRACT1)	110848	152062	149000	3061	4	4	320
(PRACT2)	110848	166356	163295	3061	3	7	326
(NO_STOP)	120373	177123	173861	3262	-	1	249

Within this estimation we discussed that we could assume different lengths of the additional stops. This assumed length of the additional stop is also called the extra stop penalty. In our experiments we used 0, 1, 2 and 3 minutes for the extra stop penalty. Note that an extra stop penalty of 0 minutes means that it is assumed that nobody faces negative effects of the additional stop. Furthermore, note that the realized timetable adaptation always includes a 3 minute delay caused by the additional stop.

In Table 3.2 we provide the best result found in the iterative procedure for each of the variants of the approach. Note that the iterative procedure does not necessarily converge to an optimal solution and thus the best solution can be found at any iteration. Therefore, we included the number of the iteration where the best solution was found. For each approach the *lower bound* as discussed in Section 3.7 is given. Furthermore, the table contains the value of the *objective* function which consist of the sum of the *rolling stock rescheduling costs* (by considering the parameters in Table 3.1) and the passenger inconvenience (measured in *passenger delay minutes* as discussed in Section 3.4.2). In the table fractional values for the total passenger delay minutes are rounded to whole minutes. We also report how many *extra stops* are included in the timetable of the best result. The *computation time* is measured in seconds and reports the computation time over all 15 iterations, and not just the computation time up to the moment the best solution is found. The latter would not be fair, since beforehand it is not known at which iteration the best solution will be found.

Performance

In the approach *NO_STOP* based on Kroon et al. (2014), timetable adaptations were not allowed. Our results show that allowing the stopping patterns to be adapted can reduce the passenger delays dramatically by about 25 to 35 percent.

From Table 3.2 we can deduce that the approach *EXACT* led in both cases to the lowest passenger delay minutes and the lowest rolling stock rescheduling costs. *EXACT**, the variant of the approach *EXACT*, reaches the same solutions, but it takes longer to get there. The estimation approach works well as long as we overestimate the positive effects of the additional stop by having a lower extra stop penalty (0 or 1 min) than the realized delay (3 min).

The performance of the approach *EST 0min* is surprising. It underestimates the negative effects and overestimates the positive effects of the additional stop but it is still able to reach solutions which do not differ much from the solutions reached by the approach *EXACT*. In deciding on which train should make an additional stop, the approach *EST 0min* assumes that an additional stop does not cause any delay and thereby no one faces negative effects of the additional stop. In every iteration an additional stop is introduced (by assuming that every additional stop has at least some positive effect).

On the other hand, the bad performance of the approach *EST 3min* is also surprising. Especially since in this approach the duration of the additional stop in the estimation approach matches the realized duration of an additional stop. However, this approach finds it never worthwhile to make an additional stop. Since this approach does not consider the capacities of the trains, it does not take rejected passengers into account. The *EST* approaches, thereby underestimate the positive effect the additional stop could have for rejected passengers. It seems that in *EST 0min* and *EST 1min* this underestimation is balanced by the overestimation of the other positive effects, but in the *EST 3min* approach the underestimation is not corrected by another overestimation.

The rules of thumb approaches *PRACT1* and *PRACT2* are outperformed by our exact approach *EXACT* and by our estimation approaches *EST 0min* and *EST 1min*. This shows that our more complex approaches are able to come to better solutions.

Iterative behavior

In Figures 3.3 - 3.8 we show for six of the variants how the solution of the case *ZTM1* changes over the iterations. The black dot indicates the first solution and by arrows we indicate how the solution evolves. On the horizontal axis we have the rolling stock rescheduling costs and on the vertical axis we have the passenger delays. With the rolling stock rescheduling both the rolling stock rescheduling costs and the passenger delays can change. However a timetable adaptation only influences the passenger delays, and therefore, a vertical drop or increase in the figure can generally be associated with a timetable adaptation.

The *EXACT* approach has a quite clear converging path to its best solution by decreasing passenger delays and rolling stock rescheduling costs. The solution of the approaches *EST 0min* and *EST 1min* first goes to solutions with low passenger delays and low rolling stock rescheduling costs, but from a certain moment, the passenger delays are increasing again. The approaches *EST 2min* and *PRACT1* converge like the *EXACT* approach to their best solution, but especially the solution of *PRACT1* does not come close to the solution of *EXACT*. The approach *EST 3min* has in every iteration the same solution.

In Figures 3.9 and 3.10 the iterative behavior of our best performing variants (*EXACT* and *EST 0min*) on case *ZTM2* are given. All variants did not show converging behavior for this case. The figures demonstrate that the *EXACT* approach explores a smaller region of solutions. The approach *EST 0min* first goes to solutions with large passenger delays, then gets to solutions with low passenger delays and in the end it goes again into the direction of solutions with high passenger delays.

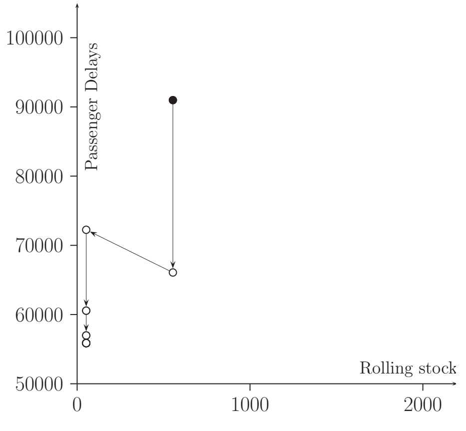

Figure 3.3: *EXACT* case *ZTM1*

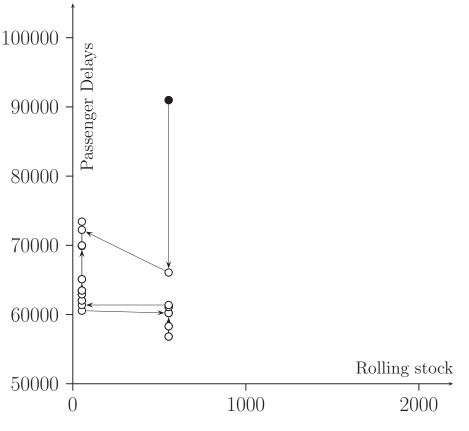

Figure 3.4: *EST 0min* case *ZTM1*

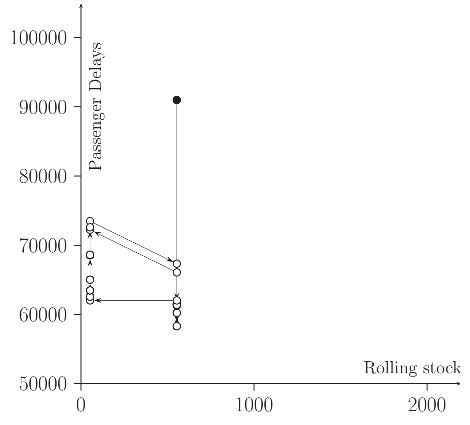

Figure 3.5: *EST 1min* case *ZTM1*

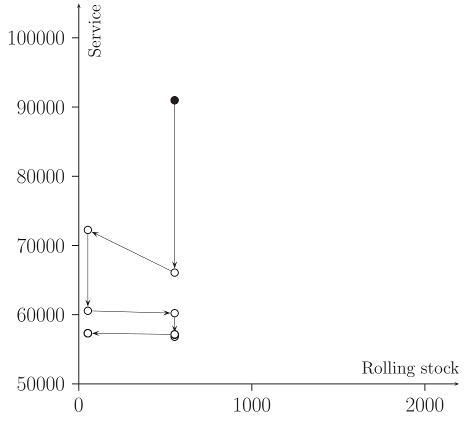

Figure 3.6: *EST 2min* case *ZTM1*

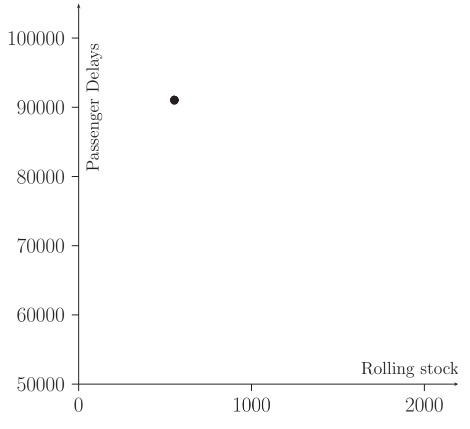

Figure 3.7: *EST 3min* case *ZTM1*

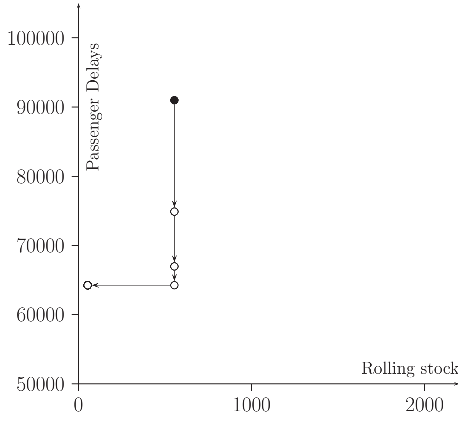

Figure 3.8: *PRACT1* case *ZTM1*

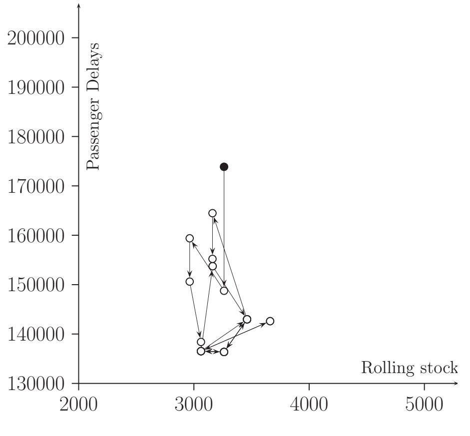

Figure 3.9: *EXACT* case *ZTM2*

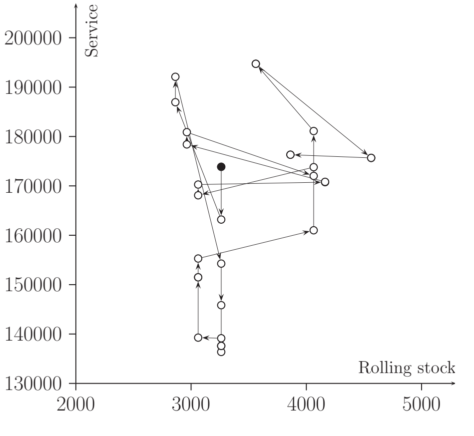

Figure 3.10: *EST 0min* case *ZTM2*

Computation time

Our approach added a module which adapts the timetable to the iterative procedure of the approach *NO_STOP* of Kroon et al. (2014). This means that we assume that by the additional computations our approach cannot be faster than the *NO_STOP* approach.

If we use the *EXACT* approach, an instance is solved in about double the time of the *NO_STOP* approach. The other variants of the approach solve the instances faster (within 6 minutes).

The first rolling stock rescheduling step, to determine the number of trains without rolling stock, is carried out in about 80 seconds. Then, next rolling stock rescheduling steps take 4 to 5 seconds per iteration. The timetable rescheduling phase takes 8 to 15 seconds per iteration within the *EXACT* approach, since multiple simulations must be carried out. The computation time of the timetable rescheduling phase drops to 1 to 6 seconds per iteration for the estimation approaches *EST* and to less than 1 second for the approaches *PRACT1* and *PRACT2*.

Table 3.3: Results with costs of the passenger flows times 10

Solution Method	Lower bound	Objective	Passenger delay minutes	Rolling stock rescheduling costs	Extra stops	Iteration of best solution	Computation time (sec)
Instance *ZTM1*							
(EXACT)	335260	558793	55874	53	4	7	568
(EXACT*)	335260	558793	55874	53	4	14	541
(EST 0min)	335260	568737	56818	554	3	3	346
(EST 1min)	335260	583589	58304	554	4	4	351
(EST 2min)	335260	568737	56818	554	3	6	350
(EST 3min)	335260	910355	90980	554	0	1	363
(PRACT1)	335260	642566	64251	53	3	7	297
(PRACT2)	335260	642566	64251	53	3	7	296
(NO_STOP)	496260	910355	90980	554	-	1	210
Instance *ZTM2*							
(EXACT)	1108480	1363686	135962	4064	3	11	431
(EXACT*)	1108480	1363686	135962	4064	3	5	430
(EST 0min)	1108480	1375949	137189	4064	4	4	346
(EST 1min)	1108480	1366944	136368	3262	3	4	298
(EST 2min)	1108480	1593181	158992	3262	2	4	331
(EST 3min)	1108480	1741873	173861	3262	0	1	359
(PRACT1)	1108480	1473060	146900	4064	2	2	313
(PRACT2)	1108480	1628519	162446	4064	3	12	293
(NO_STOP)	1203730	1741873	173861	3262	-	1	237

3.8.4 Additional tests

To see what happens with the solutions if we put more weight on the passenger delays, we run all approaches also with an objective in which the cost of the passenger flows is multiplied by 10. The results are presented in Table 3.3. The results and the performance are almost similar to the results in Table 3.2. In one third of the cases the rolling stock rescheduling costs are slightly higher to reach lower passenger delays.

In a third test we experiment on how the approaches behave if we give additional penalties to longer delays. In these tests we penalize delays between 15 and 30 minutes with an additional 5 minutes delay and delays longer than 30 minutes with an additional 10 minutes delay. Again one can see from the results in Table 3.4 that these additional penalties do not influence the solutions. In more than half of the cases, the best solution found is the same as in the situation without these additional penalties (as presented in Table 3.2). For the other cases the differences were not large.

These two additional tests show that our approach is not sensitive to changes in the evaluation of the passenger inconvenience.

Table 3.4: Results with 5 minutes additional penalty for delays larger than 15 minutes and 10 minutes additional penalty for delays larger than 30 minutes

Solution Method	Lower bound	Objective	Passenger delay minutes	Rolling stock rescheduling costs	Extra stops	Iteration of best solution	Computation time (sec)
Instance *ZTM1*							
(EXACT)	36171	59342	55874	53	4	8	572
(EXACT*)	36171	59342	55874	53	4	8	563
(EST 0min)	36171	60777	56818	554	3	3	341
(EST 1min)	36171	62297	58304	554	4	4	348
(EST 2min)	36171	60777	56818	554	3	3	361
(EST 3min)	36171	94969	90980	554	0	1	360
(PRACT1)	36171	67684	64251	53	3	8	290
(PRACT2)	36171	67683	64251	53	3	8	273
(NO_STOP)	52741	94969	90980	554	-	1	214
Instance *ZTM2*							
(EXACT)	114988	144513	136527	3061	3	5	417
(EXACT*)	114988	144513	136527	3061	3	5	492
(EST 0min)	114988	144540	136368	3262	3	4	312
(EST 1min)	114988	144540	136368	3262	3	4	331
(EST 2min)	114988	167383	159167	3061	2	5	344
(EST 3min)	114988	182073	173861	3262	0	1	339
(PRACT1)	114988	155026	146930	3061	4	4	324
(PRACT2)	114988	167639	159563	3061	2	5	309
(NO_STOP)	124813	182073	173861	3262	-	1	201

3.9 Conclusions and further research

In this paper we proposed a disruption management approach which integrates the rescheduling of rolling stock and the adaptation of stopping patterns with the aim of improving passenger service.

Computational tests are performed on realistic large-scale instances of the Dutch railway network. The two tested instances show that allowing the timetable to be adapted can reduce the total delay of passengers by more than 20 percent without increasing the rolling stock rescheduling costs. We suggested several variants of the approach, with the difference lying in the way of how the timetable changes are evaluated. These variants lead to different results and different computation times, but the results per variant are not quite sensitive to the exact cost parameter settings.

Our solution approach does not necessarily converge to an optimal solution. The lower bounds indicate that the gap between the solution and the lower bound is decreased by allowing stopping pattern adaptations. However the gap is still significant, which is probably caused by the weak lower bound. This is a topic for future research.

In future research we will incorporate other timetable decisions as well, for example reroutings of trains. Furthermore we want to proceed with integrating delay management decisions into the model, which will be quite challenging since the delay management approach is already a difficult problem to solve on its own.

Chapter 4

Railway Crew Rescheduling with Retiming

This chapter started with research in my MSc thesis and is my first paper finished during my PhD. It has been published in *Transportation Research Part C: Emerging Technologies* (**Veelenturf** et al. (2012)).

Co-authors: D. Potthoff, D. Huisman and L.G. Kroon

4.1 Introduction

Passenger railway operations frequently face unforeseen events like malfunctioning infrastructure, accidents, or rolling stock breakdowns. As a consequence of these events, part of the railway infrastructure is temporarily unavailable. Therefore, it may not be possible to operate the timetable as planned then. In this paper we focus on disruptions, which are situations in which an unforeseen event makes the resource schedules infeasible. Jespersen-Groth et al. (2009) describe the common disruption management process in Europe as the accomplishment of three interconnected steps: (1) timetable adjustment, (2) rolling stock rescheduling, and (3) crew rescheduling. Because of their complexity and the limited time available for decision making in a disrupted situation, these steps are usually carried out sequentially. First, an adjusted timetable is constructed by canceling, delaying or rerouting a number of trains. In the next step it is checked whether modified rolling stock schedules compatible with the adjusted timetable can be found. Finally, in the third step, the crew members (train drivers and conductors) are rescheduled according to the adjusted timetable and rolling stock schedule. However, notice that in case of train drivers the rolling stock schedule hardly influences the crew rescheduling: every train needs exactly one driver. If during the rolling stock or crew rescheduling steps no rolling stock or crew for a task of the adjusted timetable can be found, then another iteration through the steps is necessary. In that case, a different timetable, where some trains run on different times or are canceled, is needed.

An infeasibility of the crew rescheduling step suggests to use a further adjusted timetable where some additional trains are canceled. If this is compatible with the rolling stock schedule, then this is a solution. However, in this paper we show that sometimes no additional trains need to be canceled if the departures of some trains are delayed by just a couple of minutes, which is called *retiming*. It is quite clear that up to 1,000 passengers waiting for a train on a busy station during the peak hours will prefer a somewhat delayed train over a canceled one.

In this paper, we study an extension of the crew rescheduling problem, where the crew rescheduling step is combined with small timetable modifications. More precisely, the departures of some trains may be delayed by a few minutes. This gives more flexibility to the rescheduling of the crew duties in the disruption management process, and may avoid undesirable iterations through the three steps. Moreover, this new approach is able to provide high quality solutions from a service point of view, as was indicated above. The crew rescheduling approach in this paper will be applied for train drivers only since, in contrast to the conductors, a train cannot run without a driver.

Note that in this paper we do not aim at a complete integration of the timetable adjustment step and the crew rescheduling step: The timetable adjustment step is the first and leading step in the disruption management process. After the rolling stock has been rescheduled, the adjusted timetable is modified again in the crew rescheduling step *only* if this turns out to be appropriate there. Clearly, the latter timetable modifications are initiated by the train operator. However, since these are only minor timetable modifications, they cannot be distinguished from other small delays, and thus they do not lead to discussions with the infrastructure manager. For the same reason, the structure of the rolling stock circulation remains the same under the retiming options. In addition, we do not study integration of other steps in the disruption management process, such as integrating the rescheduling of rolling stock and crew, or train drivers and conductors. Such integrations are very hard from a computational point of view. Moreover, the main relation between rolling stock and crew is caused by the number of conductors that is needed to operate a train. In practice, these rules are often violated in the disruption management process, because a train can always run with less (possibly even zero) conductors. On the other hand, the possibility of retiming in crew rescheduling is often applied in practice. Therefore, it is relevant to study whether algorithms in the crew rescheduling process can consider this feature.

The first contribution of this paper is a new formulation for railway crew rescheduling with retiming, where the retiming options are modeled as discrete alternatives. Moreover, we show how to adapt the solution approach of Potthoff et al. (2010) in order to keep the increase in

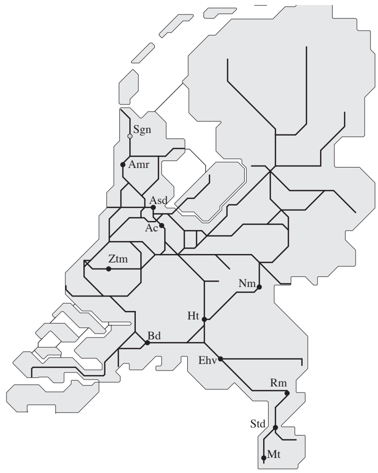

Figure 4.1: Part of the Dutch railway network used by NS

computation time for the extended model moderate. We evaluate our approach based on real-life data from Netherlands Railways (NS), the largest passenger railway operator in the Netherlands. Finally, we show that crew rescheduling with retiming allows to find better solutions than crew rescheduling without retiming.

The remainder of this paper is organized as follows. A problem description is provided in Section 4.2. The existing literature on crew rescheduling is reviewed in Section 4.3. In Section 4.4 we present the mathematical formulation of railway crew rescheduling with retiming. Our solution approach is discussed in Section 4.5. Computational results are presented in Section 4.6. In Section 4.7 we draw some conclusions and give some recommendations for further research.

4.2 Problem description

We first introduce some railway terminology which is necessary to clearly describe the problem. Most of the services offered by passenger railway operators are regular service trips (commonly known as trains) on specified lines according to a published timetable. A line is determined by a start station and an end station and a number of intermediate stops. In the Netherlands, all lines are operated with a frequency of once or twice per hour. An example of such a line is the 800-line between Maastricht (Mt) and Alkmaar (Amr) with 13 intermediate stops. In the rush hours the 800-line is extended to Schagen (Sgn) with 2 additional intermediate stops, see Figure 4.1.

As stated earlier, we limit ourselves to rescheduling the train drivers. In order to operate the timetable, trains are split into trips between *relief points*. A relief point is a station where a driver can switch from one rolling stock unit to another. The work that must be carried out by the drivers is divided into *tasks*. Several tasks may correspond to the same trip. For example, each trip is related to the task of driving the train, as well as to one or more tasks for conductors.

The relief points on the 800-line are Maastricht (Mt), Sittard (Std), Roermond (Rm), Eindhoven (Ehv), 's-Hertogenbosch (Ht), Utrecht (Ut), Amsterdam (Asd) and Alkmaar (Amr). Note that the begin/end station during peak hours, Schagen (Sgn), is not a relief point, so a driver arriving in Schagen has to stay on the rolling stock and drive the next train of the 800-line from Schagen back to Alkmaar. This results in a task from relief point Alkmaar to relief point Alkmaar.

A *duty* is a sequence of tasks which is carried out by one crew member on a single day. Each duty ends at the same crew base as where it started. The set of crew bases is a subset of the set of relief points. Sometimes a duty contains a so called *deadheading task*, which is used to relocate the crew. A deadheading task means that the driver is not driving the train, but that he is a passenger on that train. Another possibility is that the duty contains a *repositioning task*. A repositioning task is comparable to a deadheading task, with the difference that it uses another mode of transportation, for example a bus, a taxi or a train of another operator.

The operational crew rescheduling problem (OCRSP) is relevant in a disrupted situation, where, starting at some point in time τ (the start time of the disrupted situation), parts of the original timetable and rolling stock and crew schedules have become infeasible. OCRSP assumes that the timetable and the rolling stock have been rescheduled already from time instant τ onwards, and takes the rescheduled timetable and rolling stock schedule as input. OCRSP tries to find a replacement duty for each original duty, such that as many tasks as possible of the adjusted timetable are covered. For each original duty, a replacement duty consists of the already completed (possibly empty) part of the duty until time instant τ, together with a feasible completion of the duty. Here a feasible completion is a sequence of tasks following the already completed part of the duty, resulting in a replacement duty which satisfies a number of rules. In this research the following rules are used:

- A replacement duty needs to start and end at the same crew base as the original duty.
- A replacement duty may end up to 60 minutes later than the end time of the original duty.
- If, in a replacement duty, two subsequent tasks must be carried out on different rolling stock units, then a certain minimum transfer time between the two tasks must be taken into account.

- A replacement duty which is longer than 5 1/2 hours must contain a meal break of at least 30 minutes at a relief point with a canteen. Moreover, the time before and after the meal break must be less than 5 1/2 hours.

- A replacement duty can cover a task only if the involved driver is qualified for the route and is licensed for the rolling stock type.

Not all duties in the original crew schedule contain assigned tasks. There exist a number of *reserve* duties, where the driver is on *stand-by* for a specified amount of time at a major station. The purpose of these reserve duties is that they can be used during crew rescheduling in a disrupted situation.

If in a solution to the OCRSP a task cannot be covered by any crew member, it means that no compatible crew schedule for the adjusted timetable can be found. In that case, the adjusted timetable cannot be carried out, and hence the railway operator has to come up with another adjusted timetable, for which it is possible to find a compatible crew schedule (and rolling stock circulation).

The idea of retiming is to evaluate not just one fixed timetable but a relatively small number of similar timetables at once. By slightly delaying the departure of some trains, more connections for drivers are allowed and hence more feasible completions may exist. Thus it may be possible to find a better crew schedule. Anyway, the objective of retiming is to keep the amount of additional delays as small as possible. Retiming is only allowed if there is no other way to get a certain task covered.

In Figure 4.2a we show the original duty Ah 114 from crew base Arnhem (Ah) in case the two southbound routes from 's-Hertogenbosch (Ht) to Breda (Bd) and Eindhoven (Ehv) are blocked from 15:30 to 18:30. The duty started with driving task $3043/e$ (the fifth task of train 3043) from Arnhem (Ah) to Nijmegen (Nm). At 15:30, when the disruption started, the driver has completed his next two tasks and is carrying out task $3653/b$. The meal break (MB) was planned in Roosendaal (Rsd), and thereafter the duty was supposed to end with driving train 3666 from Roosendaal (Rsd) to Arnhem (Ah), $3666/a$–$3666/d$. However, due to the route blockage, task $3653/c$ is canceled. Therefore, original duty Ah 114 is infeasible now. A replacement duty is shown in Figure 4.2b. Note that because the rescheduling takes place at 15:30, the first four tasks of the duty cannot be changed. After those four tasks, the driver arrives in 's-Hertogenbosch (Ht) at 15:48. If the next task has to be carried out on different rolling stock, a minimal transfer time of 10 minutes must be respected. So the replacement duty is allowed to carry out task $16054/a$ to Utrecht (Ut) at 16:02, which is operated on different rolling stock than task $3653/b$. From Utrecht (Ut) the driver could go back to 's-Hertogenbosch

(Ht) by driving task $861/e$. Finally, the duty can carry out tasks $3666/c$ and $3666/d$ as in the original duty.

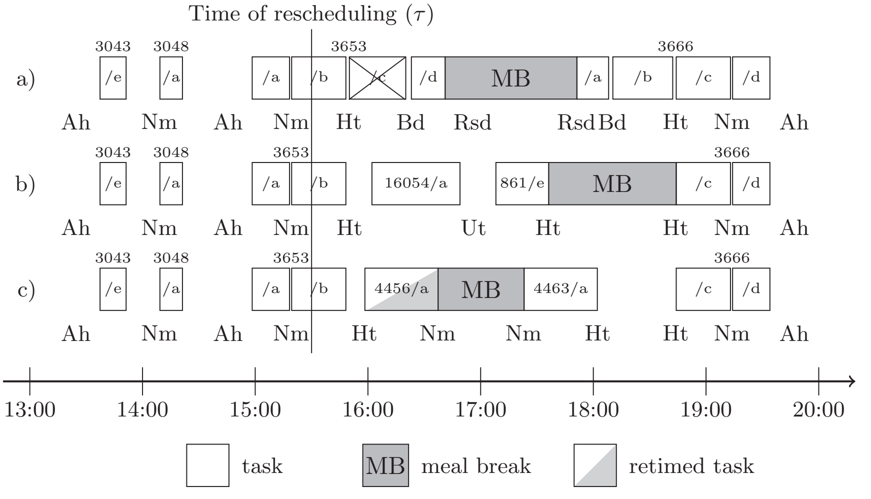

Figure 4.2: Replacement duties for duty "Ah 114"

The motivation for retiming is to facilitate replacement duties that are not feasible in a fixed timetable. For example, the planned departure time of task $4456/a$ is 15:56 and the task is operated on different rolling stock than task $3653/b$, which means that due to the minimum transfer time a transfer between task $3653/b$ and task $4456/a$ is only allowed if the latter task is delayed by at least 2 minutes. Figure 4.2c shows a replacement duty where task $4456/a$ is delayed by 2 minutes. This replacement duty is not feasible without retiming. Note that, if duty Ah 114 is the first duty arriving in 's-Hertogenbosch (Ht) after 15:30, then allowing the delay of task $4456/a$ by at least 2 minutes is the *only* way to get this task covered by a duty.

Modeling flexibility of departure times in a railway timetable is far from trivial due to a large number of interdependencies. Throughout this paper we therefore assume that:

(i) A delayed departure of a task by χ minutes leads to a delayed arrival of the task by χ minutes.

(ii) A delayed task does not affect other tasks using different rolling stock.

Assumption (i) is not always true in practice. On the one hand, the planned running time for a train may include some time supplement that could be utilized to (partly) absorb delays. On the other hand, a train that is running later than planned could experience an additional delay due to conflicts with other trains. Conversely, a delayed train may also affect other trains. For example, a faster train may get stuck behind a slower delayed train. Figure 4.3 shows part of the 2007 timetable for the route between 's-Hertogenbosch (Ht) and Nijmegen (Nm). Two lines use this route, the 3600 intercity line from Roosendaal to Arnhem and the 4400 regional line from 's-Hertogenbosch to Nijmegen. If the departure of the regional train 4456 is delayed, for example by 9 minutes, it still departs before the intercity train 3656. As indicated in the figure the faster intercity train 3656 catches up with the delayed regional train 4456. This causes a conflict in the timetable. If overtaking on the last part of the route is not possible, the intercity train will get stuck behind the regional train and experience a delay. This example shows that assumption (ii) does not always hold as well. However, at this point in time it seems reasonable since the objective of this paper is to analyze the potential retiming in crew rescheduling might offer. Note that assumption (ii) holds if train 4456 is delayed by just 2 minutes.

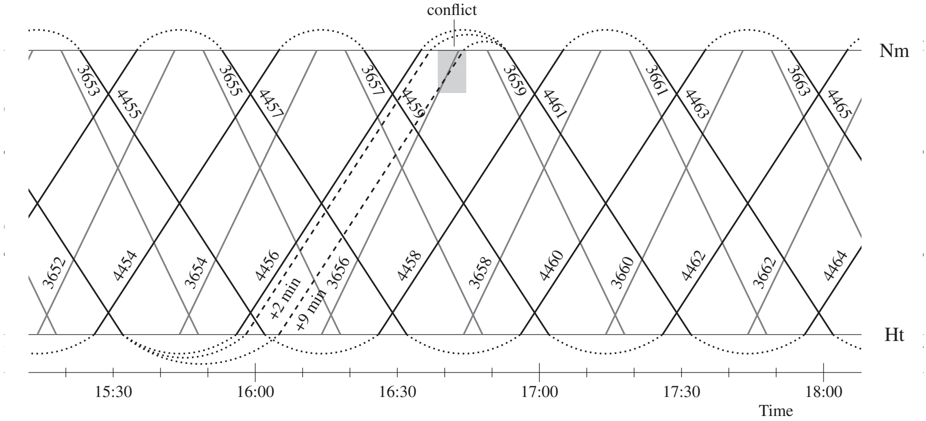

Figure 4.3: An example of a delayed task between 's Hertogenbosch (Ht) and Nijmegen (Nm)

4.3 Literature review

The literature on railway crew rescheduling is still scarce. Walker et al. (2005) was the first paper describing railway crew rescheduling. They present a model for simultaneous railway timetable adjustment and crew rescheduling. The timetabling part of the model where the departure of tasks can be chosen within certain time windows is linked with a crew rescheduling part where generic driver duties are chosen. Here a generic driver duty is a sequence of tasks that is feasible with respect to the start and end locations of consecutive tasks. Duty length and task (piece-of-work) sequencing constraints ensure that the departure times are chosen such that only the rule related to the break may be violated in the selected duties. Breaks are added to the duties during the branching process. A conflict free timetable could be achieved by adding an enormous number of train crossing and overtaking constraints. The authors propose to relax these constraints in the initial model and to resolve violations by branching on the waiting decisions between involved train pairs.

Recently, Rezanova and Ryan (2010) and Rezanova (2009) presented a solution approach for railway crew rescheduling under the assumption that the timetable is fixed. The problem is formulated as a set partitioning problem with side constraints. The LP-relaxation of the problem provides strong lower bounds, which is due to the perfectness of certain submatrices of the constraint matrix. The proposed solution approach is a depth-first search in a branch-and-price tree. The problem is first initialized with a small disruption neighborhood, which contains only duties that cover delayed, canceled or re-routed tasks and that is limited by a recovery period. As long as constraints are violated while solving the LP-relaxation, the disruption neighborhood is extended by either adding more duties to the problem or by extending the recovery period. In order to deal with new information becoming available, the crew rescheduling algorithm is used on a rolling time horizon approach similar to the one proposed by Nielsen et al. (2012) for rolling stock rescheduling.

Potthoff et al. (2010) also deal with railway crew rescheduling without retiming. The problem is modeled in a similar way as in Rezanova and Ryan (2010) and Rezanova (2009) with the difference that tasks may be covered by more than one duty. First, an initial core problem containing the infeasible duties and some candidate duties is solved. If tasks cannot be covered, then new core problems representing the neighborhood of an uncovered task are explored iteratively. The neighborhood is constructed based on duties that can potentially cover the uncovered task. Therefore, in the neighborhood they include duties that currently cover tasks that depart around the same time from the same station as the uncovered task in a first step. In a second step they add duties which are similar to the ones selected in the first step. The similarity of two duties is defined as the number of stations that are visited by both duties at roughly the same time. See Section 4.5.2 for further explanation. The core problems are explored with a column generation based heuristic. Within the column generation procedure, Lagrangian relaxation is applied to the restricted master problems and approximate dual solutions are obtained with subgradient optimization. Moreover, vectors of Lagrangian multipliers are used as input for a greedy algorithm that constructs feasible solutions. The current paper describes an extension of the method of Potthoff et al. (2010).

In the airline domain, crew rescheduling received attention much earlier than in the railway domain. An early paper on airline crew rescheduling is Johnson et al. (1994). Note that in the airline domain crew rescheduling is also known as crew recovery. For a recent review of literature on airline crew rescheduling we refer to Clausen et al. (2010). Stojković and Soumis (2001) and Abdelghany et al. (2004) are the first papers that extend crew rescheduling with the possibility to retime flights.

In Stojković and Soumis (2001) some flights may be delayed within specified time windows while new duties for pilots are generated simultaneously. The problem is formulated as a multi-commodity network flow problem with time windows and *flight precedence constraints*. The purpose of the flight precedence constraints is to ensure that minimum transfer times in the underlying aircraft rotations are not violated and to keep important passenger connections. The problem is separable per pilot and is solved with a branch-and-price algorithm. The decisions about the departure times of the flights are taken in the master problem. Therefore, it is not possible to take the meal break rule as presented in Section 4.2 into account in a straightforward manner.

The model of Stojković and Soumis (2001) is extended to the multi-crew case in Stojković and Soumis (2005). In the multi-crew case every flight has to be covered by exactly ν crew members. This is achieved by deriving ν tasks per flight which need to be covered exactly once. Again the departure time of some flights may be chosen within a time window. *Same departure time constraints* constraints are added to the model to make sure that the same departure time is chosen for all tasks selected for a flight. Two options are presented in order to deal with flights that cannot be covered ν times. In one option covering less than ν tasks is accepted, while in the second option either all ν tasks or none of the tasks derived for a flight are covered. As in Stojković and Soumis (2001) the problem is solved with a branch-and-price algorithm using specialized branching decisions.

Abdelghany et al. (2004) present a rolling approach for multi-crew rescheduling with retiming of flights. The approach tries to resolve as many conflicts as possible in crew duties during irregular operations. In a preprocessing step, flights from duties with conflicts and flights from selected candidate crews are divided into sets of resource independent flights, each leading to a recovery stage. Flights are resource independent if they cannot appear in a resource schedule together. In the rolling approach the recovery stages are tackled in increasing order of time. For each recovery stage an assignment problem with additional continuous variables for the departure times is solved with a Mixed Integer Programming solver. In the model, every flight has three crew positions. Additional constraints enforce that neither duty limits nor transfer times are violated. The model allows to assign less than three crew members to a flight, which means that the flight is still under-staffed in the final solution. In general, it seems possible to apply this approach also in a railway setting as considered in this paper. However, when decisions are taken in the recovery stages, the effect of these decisions for the assignment of flights in the later stages is not considered. This could lead to suboptimal solutions.

Abdelghany et al. (2008) present an integrated approach to recover the flight schedule, aircraft and crew at the same time. The overall approach follows Abdelghany et al. (2004), but the Mixed Integer Program for the recovery stages is extended to deal with different resources, namely aircraft, pilots and flight attendants. Either the required number of resource units per type has to be assigned to a flight, or no resource units at all. The latter means that the flight is canceled. Moreover, qualification constraints are added. For example, the pilot must be qualified for the assigned aircraft type.

Crew scheduling with flight retiming in the planning phase is discussed by Klabjan et al. (2002). Mercier and Soumis (2007) introduce an integrated model for flight retiming, aircraft routing, crew rescheduling.

4.4 Mathematical formulation

In this section, we formulate the operational crew rescheduling problem with retiming as an integer linear program. Therefore, we first introduce some notation. We use *copies of tasks* to represent the retiming possibilities of the tasks, as proposed by Mercier and Soumis (2007). The copies differ from each other in their departure and arrival times. Using copies of tasks limits the retiming possibilities, since the departure time cannot be chosen continuously and the retiming possibilities of a task must be determined beforehand.

We denote the set of tasks by N, indexed by i. Let s_i^{dep} (s_i^{arr}) denote the departure (arrival) station of task $i \in N$. The planned departure and arrival time are given by t_i^{dep} and t_i^{arr}, respectively. The minimum required dwell time after task i is w_i. Moreover, for every task $i \in N$ a penalty f_i is defined for not covering task i. Furthermore, we derive a number of copies $e \in E_i$ for every task $i \in N$. E_i contains at least the copy representing the planned departure time of task i. Denote by $N^c \subseteq N$ the tasks i for which $|E_i| \geq 2$. E is the union of all sets E_i. With $i(e)$ we refer to the task copy e is derived from. With every copy $e \in E$ we associate the delay d_e compared to the planned departure time $t_{i(e)}^{\text{dep}}$, as well as a cost parameter g_e representing the penalty for the delay. The sets $\hat{E}_e$ and $\check{E}_e$ contain all copies of the same task ($e' \in E_i(e)$) for which the delay $d_{e'}$ is respectively larger or smaller than the delay d_e.

A rolling stock composition may propagate a delay from one task to another. In the following we describe how this is taken into account. If two tasks i and j are operated directly after each other on the same rolling stock composition, then task j is denoted by $r(i)$. If task i is the last task on a rolling stock composition, then $r(i)$ is defined to be 0. If $r(i) \neq 0$, then a minimum turnaround time u_i between tasks i and $r(i)$ is to be respected. Thus the selection of the copy for task $r(i)$ that is used in a duty depends on the selection of the copy for task i and vice versa. Note that the turnaround time is 0 if the rolling stock composition is continuing in the same direction after task i. Let $h_i = \max(w_i, u_i)$ be the minimum time that is needed after the arrival of task i before the rolling stock composition is available for task $r(i)$. Then for each copy $e \in E_i$ we define the set L_e as the set of copies of task $r(i)$ that can be selected for task $r(i)$ if copy e is selected for task i. More precisely, an additional constraint on L_e ensures that it only contains copies of $r(i)$ which are not in a set $L_{e'}$ of a copy e' of the same task $i(e)$ with less delay. So every copy of $r(i)$ is in exactly one set L_e of a copy $e \in E_i$. This means that:

$$L_e = \{f \in E_{r(i)} \backslash \bigcup_{e' \in \check{E}_e} L_{e'} \,|\, (t^{\text{dep}}_{r(i)} + d_f) - (t^{\text{arr}}_i + d_e) \geq h_i, \forall e' \in \hat{E}_e \;\; (t^{\text{dep}}_{r(i)} + d_f) - (t^{\text{arr}}_i + d_{e'}) < h_i\} \quad (4.1)$$

Thus the set L_e contains all copies of task $r(i)$ which cannot be selected for task $r(i)$ if a copy of task i with more delay than copy e is selected for task i. Note that it is possible that $L_e = \emptyset$. Moreover, let B_e be the set of copies of the same task, but with a smaller delay. Formally,

$$B_e = \{e' \in E_{i(e)} \mid d_{e'} \leq d_e\} \quad (4.2)$$

We introduce a binary decision variable z_i for every task $i \in N$. If task i is canceled, z_i is set to 1, otherwise z_i is set to 0. Furthermore, v_e is a binary decision variable with $v_e = 1$ if copy e is selected for task $i(e)$ and 0 otherwise. Now we can introduce the following constraints to model the delay propagation:

$$z_i + \sum_{e' \in B_e} v_{e'} - \sum_{e' \in L_e} v_{e'} \geq 0 \quad \forall i \in N^c : r(i) \neq 0, \forall e \in E_i \quad (4.3)$$

This ensures that a copy in L_e can only be used for $r(i)$ if task i is canceled or if one of the copies $e' \in B_e$ is selected for task i. If a copy with more delay than copy e is selected for task i, a copy in L_e may not be used.

The following example in Table 4.1 illustrates the definition of L_e (see (4.1)). Consider train 3552 from Eindhoven (Ehv) to Hoofddorp (Hfdo) via 's-Hertogenbosch (Ht) and Utrecht (Ut). Thus there are three consecutive tasks assigned to the same rolling stock composition, hence $u_l = u_m = 0$. Assume we derive two copies for the first two tasks with 0 and 3 minutes delay respectively. Detailed information about the copies is shown in Table 4.1. Let us assume that $h_l = h_m = 2$ minutes. Then according to (4.1): $L_d = \{e\}$, $L_{d'} = \{e'\}$, $L_e = \emptyset$ and $L_{e'} = \{f\}$. The last set results from the fact that the planned dwell time of train 3552 in Utrecht is 6 minutes, so even if this train arrives with a delay of 3 minutes in Utrecht, the next task can still depart at the planned time. This is an example where a delay can be absorbed due to

Task	Copy	Delay (min)	Origin	Destination	Departure	Arrival
l	d	0	Ehv	Ht	14:47	15:06
l	d'	3	Ehv	Ht	14:50	15:09
m	e	0	Ht	Ut	15:08	15:37
m	e'	3	Ht	Ut	15:11	15:40
n	f	0	Ut	Hfdo	15:43	16:27

Table 4.1: Example of copies for train 3552 from Eindhoven (Ehv) to Hoofddorp (Hfdo)

margins in the timetable. Then Constraints (4.3) will become:

$$z_l + v_d - v_e \geq 0 \tag{4.4}$$

$$z_l + v_d + v_{d'} - v_{e'} \geq 0 \tag{4.5}$$

$$z_m + v_e + v_{e'} - v_f \geq 0 \tag{4.6}$$

Furthermore, $\Delta = \Delta_A \cup \Delta_R$ is the set of unfinished original duties, where Δ_A is the set of active duties and Δ_R is the set of stand-by duties. Let K^δ be the set of all feasible completions for duty $\delta \in \Delta$. With every feasible completion $k \in K^\delta$ we associate cost c_k^δ and binary parameters a_{ik}^δ and b_{ek}^δ. Here a_{ik}^δ is equal to 1 if feasible completion k for duty δ is qualified to drive task i and 0 otherwise. Next, b_{ek}^δ is equal to 1 if feasible completion k for duty δ uses copy e and 0 otherwise. Note that b_{ek}^δ is 1 if feasible completion k uses copy e for deadheading.

Let x_k^δ be binary variables indicating if feasible completion k is chosen ($x_k^\delta = 1$), or not ($x_k^\delta = 0$). Furthermore, recall that for all $i \in N$ the binary decision variable z_i indicates whether task i is canceled or not, and that for all $e \in E$ the binary decision variable v_e indicates whether copy e is selected for task $i(e)$. Now we can formulate the operational crew rescheduling problem with retiming (OCRSPT) as follows.

$$\min \quad \sum_{\delta \in \Delta} \sum_{k \in K^\delta} c_k^\delta x_k^\delta + \sum_{i \in N} f_i z_i + \sum_{e \in E} g_e v_e \tag{4.7}$$

$$\text{s.t.} \quad \sum_{\delta \in \Delta} \sum_{k \in K^\delta} a_{ik}^\delta x_k^\delta + z_i \geq 1 \qquad \forall i \in N \tag{4.8}$$

$$\sum_{k \in K^\delta} x_k^\delta = 1 \qquad \forall \delta \in \Delta \tag{4.9}$$

$$|\Delta| v_e - \sum_{\delta \in \Delta} \sum_{k \in K} b_{ek}^\delta x_k^\delta \geq 0 \qquad \forall e \in E \tag{4.10}$$

$$\sum_{e \in E_i} v_e + z_i = 1 \qquad \forall i \in N \tag{4.11}$$

$$z_i + \sum_{e' \in B_e} v_{e'} - \sum_{e' \in L_e} v_{e'} \geq 0 \qquad \forall i \in N^c : r(i) \neq 0, \forall e \in E_i \tag{4.12}$$

$$x_k^\delta \in \{0,1\} \quad \forall \delta \in \Delta, \forall k \in K^\delta \tag{4.13}$$

$$v_e \in \{0,1\} \quad \forall e \in E \tag{4.14}$$

$$z_i \in \{0,1\} \quad \forall i \in N \tag{4.15}$$

We refer to Model (4.7)–(4.15) as OCRSPRT$_1$. In the objective function (4.7) the deviation from the planned crew schedule, the penalties for canceled tasks, and the penalties for delays are minimized. Constraints (4.8) ensure that every task is either assigned to one or more qualified drivers, or is canceled. By Constraints (4.9) exactly one feasible completion must be selected for every original duty. Constraints (4.10) make sure that the binary variable v_e is set to 1 if copy e is used in any selected feasible completion. That only one copy per task may be used is modeled by Constraints (4.11). Moreover, these constraints guarantee that deadheading is not possible on tasks which have been canceled.

Constraints (4.12) are the same as Constraints (4.3), and model the dependency between the selected copies of consecutive tasks on the same rolling stock composition: If copy f is used for task $r(i)$, then task i is either canceled, or an appropriate copy from the set E_i is selected for this task. Here we assume that stand-by rolling stock may be used if necessary. That is, if a task has been canceled, then the next task on the rolling stock composition is served by stand-by rolling stock and may therefore depart at every possible departure time. Obviously, Constraints (4.12) are only required for tasks with multiple copies. Some of the Constraints (4.12) are redundant if $L_e = \emptyset$, but also if $L_e \neq \emptyset$ they can be redundant by Constraints (4.11), (4.14), and (4.15). This is true even in the linear relaxation of OCRSPRT$_1$. Note that in the example discussed above only Equation (4.4) is needed, since Equations (4.5) and (4.6) are redundant.

An alternative model OCRSPRT$_2$ is obtained by replacing Constraints (4.10) in OCRSPRT$_1$ by

$$v_e - \sum_{k \in K^\delta} b^\delta_{ek} x^\delta_k \geq 0 \ \ \forall \delta \in \Delta, \forall e \in E \tag{4.16}$$

Constraints (4.16) are clearly stronger than Constraints (4.10). Indeed, if Constraints (4.16) are satisfied, then $\sum_{\delta \in \Delta} \left(v_e - \sum_{k \in K^\delta} b^\delta_{ek} x^\delta_k \right) \geq 0$ for all $e \in E$. Then, it also holds that $|\Delta| v_e - \sum_{\delta \in \Delta} \sum_{k \in K^\delta} b^\delta_{ek} x^\delta_k \geq 0$ for all $e \in E$. It is not difficult to see that Constraints (4.16) are really stronger than Constraints (4.10).

The foregoing implies that replacing Constraints (4.10) by Constraints (4.16) results in a tighter LP relaxation and hence in a better LP lower bound. However, $|E|$ constraints of type (4.10) are replaced then by $|E||\Delta|$ constraints of type (4.16). Thus the number of constraints of type (4.16) is much larger than that of type (4.10).

After several experiments with the solution approach described in Section 4.5, we discovered that the approach of model OCRSPRT$_2$ resulted in less uncovered tasks and less retimed tasks than the approach of model OCRSPRT$_1$. In principle the models have the same integer solutions, but since we use a heuristic approach, we do not always find an optimal solution. We also noticed that the computation times are higher if we use model OCRSPRT$_2$ instead of model OCRSPRT$_1$. However, we accept the increase in computation time to obtain better results. So, in the remainder of this paper we only consider model OCRSPRT$_2$.

4.5 Solution approach

On an average workday a crew schedule of NS contains about 1,000 duties for drivers covering in total more than 10,000 tasks. Our aim is to provide solutions of good quality within a couple of minutes of computation time. Since considering all original duties and all tasks leads to extremely large crew rescheduling instances, we extract core problems containing only subsets of the duties and the tasks. We use a Lagrangian heuristic embedded in a column generation (CG) scheme that is very similar to the one proposed by Potthoff et al. (2010). Further details of this method are provided in Section 4.3. In the following, the method of Potthoff et al. (2010) is referred to as *iterative neighborhood exploration* (INE). In this paper we investigate two approaches which use the same heuristic to explore the core problems, but that differ in the way the core problems are defined.

Our first approach is outlined in Figure 4.4. We first define an initial core problem where retiming is not allowed. A solution for this core problem is computed using the column generation based heuristic. If the computed solution covers all tasks, then we stop. Otherwise we iterate over the uncovered tasks and define one new core problem per uncovered task. These

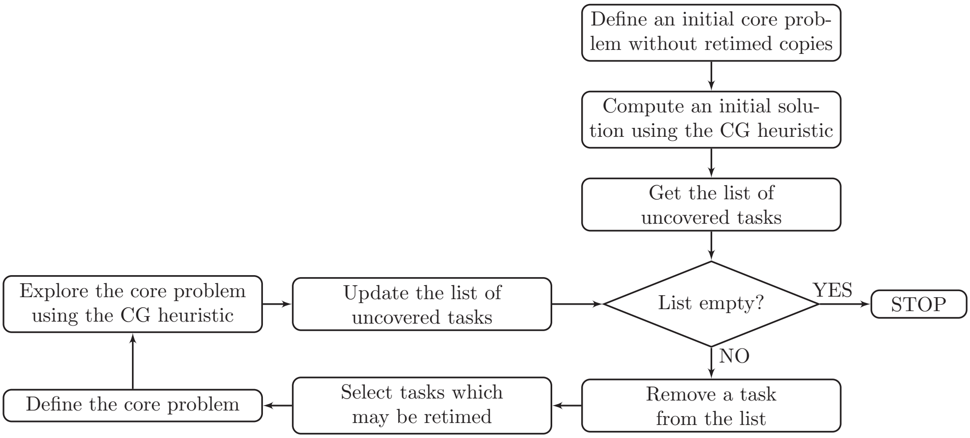

Figure 4.4: Iterative neighborhood exploration with retiming (INER)

core problems are dealt with consecutively. We start with the uncovered task with the earliest departure time. We use a neighborhood definition to select the tasks for which we allow retiming and the tasks for constructing the core problems. Each core problem is solved using the column generation heuristic, and the list of uncovered tasks is updated before solving the next core problem. If the uncovered task on which the core problem was based, is not covered after solving the core problem, no new core problem will be generated anymore around this task. This means that the task is neglected in the list of uncovered tasks. The next core problem to solve is the core problem around the uncovered task with the earliest departure time which is not neglected. We refer to this approach as *iterative neighborhood exploration with retiming* (INER). The difference with the approach of Potthoff et al. (2010) is that in INER retiming of some tasks is allowed in the neighborhood exploration phase.

Our second approach is outlined in Figure 4.5. Here we do not use an iterative neighborhood exploration. If the solution of the initial core problem contains some uncovered tasks, then a second core problem is constructed and solved. This second core problem is an extension of the initial core problem, which is obtained by adding retiming possibilities for a number of tasks. In the remainder of this paper we refer to this approach as *extended core problem with retiming* (ECPR).

In both approaches INER and ECPR we relax the initial core problem by using only Constraints (4.8), (4.9), (4.13) and (4.15). Note that in this model it can happen that feasible completions are chosen that contain deadheading on tasks which have been canceled. However, in the next core problem which is considered in both approaches, these deadheadings are not allowed anymore and a different solution is computed. We decided to use the relaxed model

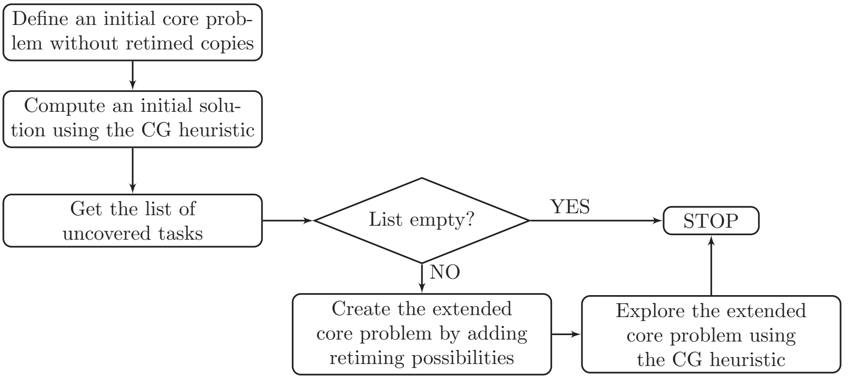

Figure 4.5: Extended core problem with retiming (ECPR)

for solving the initial core problem because the computation time of the relaxed model is much shorter than that of OCRSPRT_2.

4.5.1 Defining the initial core problems

The initial core problems in INER and ECPR are constructed in the same way as in Potthoff et al. (2010). The intention is to select the duties that are affected by the timetable adjustments and to add a small number of duties containing some tasks close in space and time to the modified tasks.

4.5.2 Neighborhoods for uncovered tasks in the INER approach

Given a task that is uncovered after the solution of the initial core problem, we define a neighborhood which is extended by retiming possibilities in a subsequent step. First we select a number of candidate duties. These duties can possibly cover the uncovered task. In order to offer some reassignment possibilities we also select a number of similar duties for each candidate duty.

The candidate duties are selected as follows. Given the departure time and station of the uncovered task j, we look at the latest task j^- that departs from the same station before task j. Then we consider the replacement duty σ that covers j^- in the current solution and check heuristically if σ could cover j, thereby considering the qualifications of the driver. If yes, then we select σ as a candidate and continue with the previous task that departs from station s_j^{dep} before j^- until we have selected r candidates. We repeat the procedure considering tasks that depart from station s_j^{dep} after task j.

Furthermore, we select the replacement duty which covers task $\hat{\jmath}$, the first task that leaves s_j^{arr} and goes back to station s_j^{dep} such that a driver can transfer from task j to task $\hat{\jmath}$. Including this replacement duty ensures that it is possible to carry out task j and then to deadhead back to station s_j^{dep}.

In the next step we select for every candidate duty the S most similar duties that have not been selected yet. We define similarity between duties in terms of the numbers of stations that are visited around the same time. We refer to Potthoff et al. (2010) for the exact definition.

4.5.3 Core problems with retiming possibilities

The primary goal of retiming is to enable solutions where less tasks have to be canceled. In order to limit the computational effort, we allow retiming only for a subset of the tasks. If we have an uncovered task which starts, for example, at 's Hertogenbosch, then this indicates that there is a shortage of crew in 's Hertogenbosch around the start time of the task. By allowing to delay some tasks starting at 's Hertogenbosch around that time, we can hopefully prevent the crew shortage. Therefore, we propose the following procedure to determine this subset. Let N^u be the set of uncovered tasks after the initial core problem has been solved. Then, for an uncovered task $i \in N^u$ we construct a set N_i^c with tasks that may be retimed as $N_i^1 \cup N_i^2$, where $N_i^1 = \{j \in N \mid s_j^{\text{dep}} = s_i^{\text{dep}}$ and $t_j^{\text{dep}} \in [t_i^{\text{dep}} - p, t_i^{\text{dep}} + p]\}$ for a certain positive parameter p. Furthermore, the set N_i^2 is recursively defined as the set of all tasks which are linked by rolling stock connections to tasks in N_i^1 or N_i^2: $N_i^2 = \Big\{r(i) \mid i \in N_i^1 \cup N_i^2, t_i^{arr} - t_{r(i)}^{dep} < h_i + \max_{e \in E_i}(d_e)\Big\}$.

For the method INER, we set $N^c = N_i^c$ for the uncovered task i currently under consideration. For the extended core problem in the ECPR approach, the tasks that may be retimed are $N^c = \cup_{i \in N^u} N_i^c$.

Let the set $\ddot{N}$ contain all tasks covered by an original duty in the neighborhood of the uncovered task under consideration when using the method INER. For the ECPR approach $\ddot{N}$ is the set of tasks of the initial core problem. The core problems are then defined by a subset of the original duties $\bar{\Delta}$ and a subset of the tasks $\hat{N}$. Here $\bar{\Delta} = \{\delta \in \Delta \mid \delta$ is covering a task $i \in \ddot{N} \cup N^c\}$ and $\hat{N}$ is the set of all tasks covered by at least one original duty $\delta \in \bar{\Delta}$. Note that, due to overcovering and deadheading, it can happen that for a task $j \in \hat{N}$ not all duties δ covering task j are in $\bar{\Delta}$. By definition of $\bar{\Delta}$, retiming is not allowed for these tasks. Denote by $\bar{N} = \{i \in \hat{N} \mid \delta \in \bar{\Delta}\ \forall \delta \in \Delta$ covering task $i\}$.

Given $\bar{\Delta}$ and $\bar{N}$ we define $\bar{E} = \cup_{i \in \bar{N}} E_i$. Moreover, we denote by $\bar{K}^\delta$ the set of feasible completions for duty δ which only cover tasks $i \in \hat{N}$. The mathematical model for a core problem is obtained by replacing N with $\bar{N}$, Δ with $\bar{\Delta}$, E with $\bar{E}$ and K with $\bar{K}$ in the model OCRSPRT_2.

4.5.4 Exploring a core problem

For computing near optimal solutions and lower bounds for the core problems we adapt the heuristic presented in Potthoff et al. (2010), which is based on a combination of column generation and Lagrangian relaxation. For an introduction to column generation we refer to Desrosiers and Lübbecke (2005). Let us first describe the building blocks, before we present our column generation based heuristic.

Combining column generation and Lagrangian relaxation

A lower bound for a given core problem can be obtained by Lagrangian relaxation. In this section we present the details of model OCRSPRT$_2$. We relax Constraints (4.8), (4.12), and (4.16) of the core problems in a Lagrangian fashion using multiplier vectors η, λ, and μ, respectively.

For simplicity we introduce $\gamma_e = \sum_{\{d \in \bar{E} \mid e \in L_d\}} \eta_d - \sum_{\{d \in \bar{E} \mid e \in \bar{B}_d\}} \eta_d$. Then, the Lagrangian subproblem equals:

$$\begin{aligned} \Theta(\eta, \lambda, \mu) \quad = \quad & \min \sum_{i \in \bar{N}} \lambda_i + \sum_{\delta \in \Delta} \sum_{k \in \bar{K}^\delta} (c_k^\delta + \sum_{e \in \bar{E}} \mu_e^\delta b_{ek}^\delta - \sum_{i \in \bar{N}} \lambda_i a_{ik}^\delta) x_k^\delta \\ & + \sum_{i \in \bar{N}} (f_i - \lambda_i - \sum_{e \in \bar{E}_i} \eta_e) z_i + \sum_{i \in \bar{N}} \sum_{e \in \bar{E}_i} (g_e + \gamma_e - \sum_{\delta \in \bar{\Delta}} \mu_e^\delta) v_e \\ & \text{s.t. } (4.9), (4.11), (4.13), (4.14) \text{ and } (4.15) \end{aligned} \tag{4.17}$$

For given vectors η, λ and μ, the optimal value of $\Theta(\eta, \lambda, \mu)$ can be calculated with a simple procedure. First, we determine the values for all x_k^δ variables. To ensure that Constraints (4.9) are not violated, we set x_k^δ equal to 1 for exactly one $k \in \arg\min \left\{ \bar{c}_k^\delta(\eta, \lambda, \mu) \middle| \, k \in \bar{K}^\delta \right\}$ for every duty $\delta \in \bar{\Delta}$. Here $\bar{c}_k^\delta(\eta, \lambda, \mu) = (c_k^\delta + \sum_{e \in \bar{E}} \mu_e^\delta b_{ek}^\delta - \sum_{i \in \bar{N}} \lambda_i a_{ik}^\delta)$ is the Lagrangian reduced cost of feasible completion k. The values of the z_i and v_e variables can be determined independently from the x_k^δ variables. The algorithm in Figure 4.6 determines for every task $i \in \bar{N}$ the values of the variables z_i and v_e ($\forall e \in \bar{E}_i$) such that Constraints (4.11) are not violated.

The Lagrangian dual problem is to find the best Lagrangian lower bound Θ^*:

$$\Theta^* = \max \Theta(\eta, \lambda, \mu), \quad \eta \geq 0, \lambda \geq 0 \text{ and } \mu \geq 0 \tag{4.18}$$

Since the number of feasible completions can be huge for some original duties, we combine Lagrangian relaxation with column generation. Instead of considering all feasible completions we consider only a subset of them in a restricted master problem (RMP). Denote by $\bar{K}_n^\delta$ the

For all $e \in \bar{E}_i$ determine $\bar{g}_e = (g_e + \gamma_e - \sum_{\delta \in \bar{\Delta}} \mu_e^\delta)$;
Select $e^* \in \arg\min \left\{ \bar{g}_e \,\middle|\, e \in \bar{E}_i \right\}$;
if $\bar{g}_{e^*} \leq f_i - \lambda_i - \sum_{e \in \bar{E}_i} \eta_e$ **then**
    Set $z_i = 0$, $v_{e^*} = 1$ and for all $e \in \bar{E}_i \setminus \{e^*\}$, set $v_e = 0$
else
    Set $z_i = 1$ and for all $e \in \bar{E}_i$, set $v_e = 0$
end

Figure 4.6: Algorithm to determine z_i and v_e for the solution of a Lagrangian subproblem

feasible completions present in the n^{th} RMP. A lower bound Θ_n^* for the n^{th} RMP is obtained by subgradient optimization, see Fisher (1981) and Beasley (1993).

Let η^n, λ^n and μ^n be the vectors of the Lagrangian multipliers corresponding to Θ_n^*. In the pricing problems of our column generation algorithm we check, per original duty, if feasible completions exist that are not in the RMP, but have a lower Lagrangian reduced cost than the feasible completions in the RMP. We refer to them as *promising* feasible completions. The pricing problem is formulated as a shortest path problem with resource constraints (see below). If promising feasible completions exist we add them to the RMP. Let $p_n^\delta = \min\{\bar{c}_k^\delta(\eta, \lambda, \mu) \mid k \in \bar{K}^\delta\}$ be the solution value of the pricing problem for duty δ, and let $r_n^\delta = \min\{\bar{c}_k^\delta(\eta, \lambda, \mu) \mid k \in \bar{K}_n^\delta\}$ be the smallest Lagrangian reduced cost of a feasible completion for duty δ in the n^{th} RMP. After solving the pricing problems for all duties $\delta \in \bar{\Delta}$, we can compute a lower bound for the core problem as $LB_n = \Theta_n^* + \sum_{\delta \in \bar{\Delta}}(p_n^\delta - r_n^\delta)$.

Feasible solutions

Next to a good lower bound, we are especially interested in near optimal feasible solutions. Based on Lagrangian multiplier vectors η, λ, and μ we try to generate feasible solutions with a Lagrangian heuristic called *GREEDY*, which is shown in Figure 4.7.

First, we order the original duties by increasing reduced cost of the x_k^δ variables that were set to 1 in the Lagrangian subproblem solution. Then we select for every duty the best feasible completion (Line 4–13). If it is the first time that a certain task appears in a selected feasible completion, the copy which is used for that task will be the only copy that is allowed to be used in all duties. So after a certain copy for a task has been selected, all feasible completions which use another copy of the same task are ignored. Moreover, we ignore feasible completions which cover copies of tasks that would violate Constraints (4.12). Since for every active duty δ the set $\bar{K}_n^\delta$ contains the artificial completion without any additional tasks (that is, the driver is assumed to go to his home depot as soon as possible), it is ensured that for every active duty at least one feasible completion is left to select. If, after the feasible completions of the duties have been selected, still some tasks are uncovered, we check if the idle stand-by duties can cover those tasks. A stand-by duty is idle if it does not cover any tasks.

The procedure *GREEDY* does not always find a feasible solution, however in most cases it will. Only in the extraordinary case that a crew member is assigned to be a passenger on a train which is not covered by a driver, the solution is infeasible. This condition is checked in Line 22.

1 Order the original duties $\delta \in \bar{\Delta}$ by increasing reduced cost of the selected x_k^δ variables;
2 Set $z_i = 1$ for all $i \in \bar{N}$ and set $v_e = 0$ for all $e \in \bar{E}$;
3 Set $\hat{\eta} = \eta$, $\hat{\lambda} = \lambda$, and $\hat{\mu} = \mu$;
4 **foreach** $\delta \in \bar{\Delta}$ **do**
5   Choose $k^*(\delta) \in \arg\min\{\bar{c}_k^\delta(\hat{\eta}, \hat{\lambda}, \hat{\mu}) \mid k \in \bar{K}_n^\delta\}$ and set the corresponding $x_{k^*(\delta)}^\delta = 1$;
6   Set $\hat{\lambda}_i = 0$ and $z_i = 0$ for all $i \in \bar{N}$ with $a_{ik^*(\delta)}^\delta = 1$;
7   **foreach** $e \in \bar{E}$ with $b_{ek^*(\delta)}^\delta = 1$ **do**
8   Define E^*: the set of copies which may not be used if copy e is used;
9   Define K^*: the set of completions which use at least one copy $d \in E^*$;
10   Ignore $\forall \delta \in \bar{\Delta}$ the completions $k \in K^*$ out of $\bar{K}_n^\delta$;
11   Set $v_e = 1$ and $\hat{\eta}_e = 0$;
12   **end**
13 **end**
14 **foreach** $i \in \bar{N}$ **do**
15   Set $\hat{\lambda}_i = f_i$, if $z_i = 1$;
16 **end**
17 Construct the set of idle stand-by duties $\bar{\Delta}_I = \{\delta \in \bar{\Delta}_R \mid a_{ik^*(\delta)}^\delta = 0 \text{ for all } i \in \bar{N}\}$;
18 **foreach** $\delta \in \bar{\Delta}_I$ **do**
19   Set $x_{k^*(\delta)}^\delta = 0$;
20   Repeat lines 5 until 12;
21 **end**
22 Check if $\sum_{e \in \bar{E}_i} \sum_{\delta \in \bar{\Delta}} \sum_{k \in K^\delta} b_{ek}^\delta x_k^\delta = 0$ for all $i \in \left\{i \in \bar{N} \mid z_i = 1\right\}$. If this condition holds, a feasible solution has been found.

Figure 4.7: Procedure *GREEDY* to construct feasible solutions

Solving the pricing problems

The problem of finding the path corresponding to the feasible completion with the smallest Lagrangian reduced cost is modeled as a shortest path problem with resource constraints. To that end, for every duty in a core problem we construct a directed acyclic graph that models all possible feasible completions. The nodes represent arrivals or departures of copies derived from the tasks. An arc goes from an arrival node to a departure node if it is possible to use the corresponding copies after each other in a feasible completion. Besides the cost, every arc has two additional parameters: a time consumption and a boolean value indicating if the arc can represent a meal break. We use a resource to measure the time spent before or after the meal break. This resource is reset to 0 if we traverse an arc that corresponds to a meal break. Moreover, along a path this resource must be between 0 and 5 1/2 hours at any node. For solving the pricing problems, we adapted the generic dynamic programming algorithm presented by Irnich and Desaulniers (2005). Note that shortest path problems with resource constraints are in general NP-hard since they are generalizations of the weight constrained shortest path problem.

The column generation based heuristic

1 $stopFix$ = false, $LB_F = -\infty, UB^* = \infty, UB_F = 0$;
2 **while** *stopFix = false* **do**
3     $stopColGen$ = false;
4   **while** *stopColGen = false* **do**
5   Compute the lower bound Θ_n^* for the RMP with subgradient optimization;
6   Call *GREEDY* with at most *maxMV* multiplier vectors and update UB^*;
7   Solve pricing problems and add promising feasible completions;
8   Compute LN_n if all pricing problems have been solved;
9   **if** *any stopping criterion for column generation is met* **then**
10             $stopColGen$ = true, $LB_F = UB_F + LB_n$;
11   **end**
12   **end**
13   **if** *any stopping criterion for fixing is met* **then**
14         $stopFix$ = true;
15   **else**
16   Fix the feasible completions for at most *maxFix* original duties and update UB_F;
17   **end**
18 **end**

Figure 4.8: The algorithm to solve a core problem

Our column generation based heuristic using the building blocks as described in Section 4.5.4, is outlined in Figure 4.8. It can be seen as a depth-first search approach in a branch-and-bound tree with column generation in every node. This is a common way of designing column generation based heuristics for crew scheduling problems, see Desaulniers et al. (2001). In Line 5 a dual solution for the RMP is obtained by Lagrangian relaxation as explained above. Another specialty in our approach is that we generate solutions throughout the algorithm, see Line 6. We denote by UB^* the cost of the best found feasible solution. When solving the pricing problems for the original duties, we do pricing and stop if we have found promising columns for more than a fraction *maxPP* of the duties. In Line 9 we use three criteria to decide

Location	ID	Time	Type
Abcoude	*Ac:1*	11:00-14:00	two sided blockage, some trains are rerouted
’s-Hertogenbosch	*Ht:1*	15:30-18:30	two sided blockage
Zoetermeer	*Ztm:1*	08:00-11:00	reduced number of trains

Table 4.2: Information about the disruption scenarios

if we stop column generation in the current node. First, we stop if no columns have been added to the RMP. Second, we stop if Θ^*_n is close to LB_n. As a third criterion we use a maximum number of column generation iterations *maxItCG* to perform in the current node. In the root node, where no feasible completions have been fixed, *maxItGC* $= \infty$, in the other nodes we can use a relatively small number to speed up the algorithm.

After terminating the column generation procedure for a node, we check in Line 13 if the best feasible solution of value UB^* is close to the lower bound LB_F which is the sum of the fixed part UB_F and the lower bound of the free variables Θ^*_n. If this is the case, we can terminate the algorithm since we know that it is unlikely to find a better feasible solution if we only fix more variables. Otherwise, we fix the feasible completions for more original duties. This is done based on the number of times a feasible completion was set to 1 in the solution of a Lagrangian subproblem during the last subgradient optimization.

4.6 Computational results

In this section we evaluate our two new approaches with retiming INER and ECPR on three disruption scenarios, *Ac:1*, *Ht:1*, and *Ztm:1*. These scenarios are based on past real-life disruptions near Abcoude (Ac), ’s-Hertogenbosch (Ht), and Zoetermeer (Ztm), see Figure 4.1. Table 4.2 presents information about the scenarios. Furthermore, we used a crew schedule from NS that was planned for a regular working day in September 2007. In order to evaluate the benefits of retiming, we compare our new methods with the method proposed in Potthoff et al. (2010). As was mentioned earlier, we refer to the latter as *iterative neighborhood exploration* (INE). Moreover, we investigate the effect of considering stand-by duties. For that reason, we determine two cases. In the first case we do not use any stand-by duty and in the second case we use a set of 46 stand-by duties.

All approaches have been implemented in C++. The tests have been performed under Windows XP on a quad core 2.99 GHz CPU machine with 3.25 GB RAM memory. However, only a single core was used in the tests.

4.6.1 Parameter settings

First of all, we used the following parameter settings which are required to determine the core problems. In the definition of N_1 we set $p = 30$ minutes. For every task in $\bar{N}^c$ we derive four copies with delays d_e equal to 0, 1, 3 and 5 minutes.

In the column generation based heuristic, we use the following settings. For partial pricing we set *maxPP* $= 0.3$. For calling *GREEDY* we set *maxMV* $= 100$. In the root node of our depth-first search *maxItCG* $= \infty$. In all other nodes we use *maxItCG* $= 10$. Furthermore, *maxFix* was set to 0.05.

4.6.2 Cost parameters for the objective function

After some initial experiments, we chose the following settings to account for the different aspects of the objective function. First, the cost of changing a duty is set to 400. The cost of sending a stranded driver home by taxi is 3,000. Covering a task in a feasible completion costs 0 if the corresponding original duty was already covering that task, and 50 otherwise. Moreover, the cost of a transfer is 0 if the transfer was already in any original duty, and 1 otherwise. The usage of a new repositioning task costs 1,000. The penalty for retiming a task is 200 per minute of delay.

The penalty f_i for canceling task i depends on the characteristic of the task. A task is of type A-B if $s_i^{\text{dep}} \neq s_i^{\text{arr}}$ and of type A-A if $s_i^{\text{dep}} = s_i^{\text{arr}}$. We set $f_i = 20,000$ if task i is of type A-B and $f_i = 3,000$ otherwise. This is motivated by the overall disruption management process. If only tasks of type A-A are canceled, then the crew schedule is compatible with the underlying rolling stock schedule under the assumption that the rolling stock assigned to the canceled A-A tasks can remain idle at the platform or can be shunted to a nearby shunt yard and pulled out again for its next trip.

4.6.3 Numerical results

Table 4.3 shows the results of the three approaches in case we allow 46 stand-by duties to be used. Table 4.4 shows the results without using stand-by duties. In these tables we use the following abbreviations: "It" is the iteration number of the general solution approach as given in Figures 4.4 and 4.5. The costs in the columns "LB" and "UB" are the lower bound on the optimal solution and the cost of the best found solution for the core problem. "Gap" represents the relative difference between the best solution and the lower bound of the core problem. The column "Sol" represents the cost of the total solution: the cost of the core problem ("UB") plus the rescheduling cost of the other duties that were selected in earlier iterations, and that are

	Method	It	$\|\bar{\Delta}\|$	$\|\bar{N}\|$	$\|\bar{E}\|$	LB	UB	Gap (%)	Sol	TT (s)	A-B	A-A	DT	TD (min)
Ac:1	INE	1	176	629	0	58718	59211	0.8	59211	98	1	0	0	0
Ac:1	**INE**	**2**	**98**	**259**	**0**	**25324**	**25324**	**0.0**	**59211**	**110**	**1**	**0**	**0**	**0**
Ac:1	**INER**	**2**	**115**	**317**	**106**	**31202**	**31202**	**0.0**	**59212**	**137**	**1**	**0**	**0**	**0**
Ac:1	**ECPR***	**2**	**187**	**670**	**30***	**58718**	**59116**	**0.7**	**59116**	**493**	**1**	**0**	**0**	**0**
Ht:1	INE	1	126	660	0	61661	61744	0.1	61744	95	1	1	0	0
Ht:1	INE	2	77	391	0	30637	30637	0.0	61694	106	1	1	0	0
Ht:1	**INE**	**3**	**72**	**372**	**0**	**30450**	**30450**	**0.0**	**61694**	**118**	**1**	**1**	**0**	**0**
Ht:1	INER	2	87	455	37	17637	17809	1.0	45649	169	0	1	3	9
Ht:1	**INER**	**3**	**79**	**413**	**56**	**14007**	**14007**	**0.0**	**45649**	**190**	**0**	**1**	**3**	**9**
Ht:1	**ECPR**	**2**	**147**	**835**	**119**	**43241**	**43751**	**1.2**	**43751**	**602**	**0**	**1**	**2**	**6**
Ztm:1	INE	1	117	432	0	51940	51991	0.1	51991	25	2	0	0	0
Ztm:1	INE	2	99	247	0	43264	43264	0.0	51991	36	2	0	0	0
Ztm:1	**INE**	**3**	**100**	**301**	**0**	**23563**	**23563**	**0.0**	**32339**	**50**	**1**	**0**	**0**	**0**
Ztm:1	**INER**	**2**	**133**	**398**	**175**	**5355**	**5667**	**5.8**	**13392**	**167**	**0**	**0**	**1**	**3**
Ztm:1	**ECPR†**	**2**	**186**	**768**	**185†**	**11982**	**12389**	**3.4**	**12389**	**572**	**0**	**0**	**1**	**3**

Table 4.3: Results with stand-by drivers.

needed to complete the solution. The total computation time in seconds including the current iteration is given in the column "TT". The columns "A-B" and "A-A" represent the number of uncovered tasks of the respective types. The last two columns give information about the used retimed copies. The column "DT" displays the number of delayed tasks and the column "TD" represents the total number of delayed minutes.

For the approaches INER and ECPR, we solve the model OCRSPRT_2. We compare the results with the INE method. Since the rescheduling model without retiming is used in the initial core problem of all three approaches, the results of the first iteration are the same. Therefore we report this result only once (for the method INE) in Tables 4.3 and 4.4. The final result obtained by each approach is shown in bold. We were not able to use the ECPR approach with $p = 30$ in the definition of N_1 since it ran out of memory. For *Ac:1* (*) we had to set $p = 5$ and for *Ztm:1* (†) we had to set $p = 20$.

In Table 4.3, we notice that by using stand-by duties the ECPR method has the best solution in all cases. However, the computation times of this approach are more than three times longer than those of the other two approaches. In terms of the number of uncovered tasks the INER approach performs the same as ECPR, except for case *Ht:1* where in the solution of INER an additional task is delayed. By delaying at most 3 tasks, both retiming approaches have less uncovered tasks than the INE approach in the cases *Ht:1* and *Ztm:1*. In case *Ac:1*, retiming did not result in better crew schedules. However, the solution of the method INE for *Ac:1* is a crew schedule which is not completely compatible with the adjusted timetable, since it has one driver deadheading on a canceled task. Note that this driver could not carry out this task himself due

	Method	It	$\vert\bar{\Delta}\vert$	$\vert\bar{N}\vert$	$\vert\bar{E}\vert$	LB	UB	Gap (%)	Sol	TT (s)	A-B	A-A	DT	TD (min)
Ac:1	INE	1	130	629	0	61136	62187	1.7	62187	86	1	0	0	0
Ac:1	**INE**	**2**	**59**	**287**	**0**	**27235**	**27235**	**0.0**	**62187**	**97**	**1**	**0**	**0**	**0**
Ac:1	**INER**	**2**	**79**	**351**	**106**	**34066**	**34066**	**0.0**	**62136**	**146**	**1**	**0**	**0**	**0**
Ac:1	**ECPR***	**2**	**141**	**670**	**30***	**60967**	**62390**	**2.3**	**62390**	**539**	**1**	**0**	**0**	**0**
Ht:1	INE	1	90	660	0	65567	65803	0.4	65803	94	1	2	0	0
Ht:1	INE	2	44	407	0	34489	34489	0.0	65803	105	1	2	0	0
Ht:1	INE	3	39	407	0	31080	31080	0.0	65803	115	1	2	0	0
Ht:1	**INE**	**4**	**40**	**405**	**0**	**29941**	**29941**	**0.0**	**63657**	**124**	**1**	**1**	**0**	**0**
Ht:1	INER	2	52	454	37	23200	23208	< 0.1	52861	144	0	3	2	6
Ht:1	INER	3	45	439	56	16747	16747	0.0	50364	163	0	2	2	6
Ht:1	**INER**	**4**	**54**	**429**	**82**	**17812**	**17812**	**0.0**	**46666**	**205**	**0**	**1**	**2**	**6**
Ht:1	**ECPR**	**2**	**114**	**871**	**157**	**44502**	**44660**	**0.4**	**44660**	**641**	**0**	**1**	**2**	**6**
Ztm:1	INE	1	71	432	0	51991	51992	0.0	51992	16	2	0	0	0
Ztm:1	INE	2	54	249	0	42058	42058	0.0	51992	23	2	0	0	0
Ztm:1	**INE**	**3**	**55**	**306**	**0**	**42159**	**42159**	**0.0**	**51992**	**36**	**2**	**0**	**0**	**0**
Ztm:1	**INER**	**2**	**86**	**390**	**175**	**5354**	**5818**	**8.7**	**14046**	**138**	**0**	**0**	**1**	**3**
Ztm:1	**ECPR†**	**2**	**140**	**768**	**185†**	**12058**	**12441**	**3.2**	**12441**	**477**	**0**	**0**	**1**	**3**

Table 4.4: Results without stand-by drivers.

to lacking licenses of this driver. The uncovered task in *Ac:1* is rerouted due to the disruption and takes half an hour longer. The crew member which was originally assigned to this task does not have a license for the new route and is therefore not allowed to drive this train. This task has to be carried out exactly at the moment of rescheduling. Thus it is not possible to cover this task without retiming it. Because of the minimum transfer time of 10 minutes, the task must be delayed by at least 10 minutes. However, the retiming approaches INER and ECPR use a maximum retiming possibility of only 5 minutes. Therefore they were also not able to cover this task. Furthermore, in additional tests in which INER and ECPR also allowed retimed copies with 10 minutes delay, it was still not possible to cover all tasks.

If we do not allow any stand-by duties to be used, see Table 4.4, then ECPR resulted twice in the best solution and INER found once the best solution. In terms of uncovered tasks and delayed minutes, the methods performed equally well. Except for case *Ac:1*, retiming of at most 2 tasks results in less uncovered tasks. Again the computation time of ECPR is by far the largest, and INER has a computation time which is at most 2 minutes longer than that of INE.

We notice that the solutions in which stand-by duties are used have lower costs, but if we only consider the number of uncovered tasks and the number of delayed tasks, it was not necessary to use stand-by duties. Moreover, for *Ht:1*, the use of stand-by duties increased the number of delayed tasks.

We performed a sensitivity analysis on the cost per minute delay to check if our conclusions still hold under different values. Therefore, we did some experiments with a cost of 100 and 300 instead of 200 per minute delay. A cost of 300 per minute delay does not lead to any significant changes in the solutions. We also discovered that a cost of 100 per minute delay resulted for INER in one case in an additional delayed train. In all other cases the solutions were quite the same as with cost of 200 per minute delay. So the performance of the different methods compared to each other does not change when we increase or decrease the delay cost.

A comparison of the computed solutions with manual solutions is not possible for several reasons. First, the solutions obtained manually by the dispatchers do not satisfy all constraints that we took into account in the paper, since it is sometimes too hard for them to find a feasible solution at all. Second, the instances described in this paper consider only single disruptions. However, in practice there are usually several disturbances per day. Since in the dispatching system used by NS it is possible only afterwards to obtain information on how the duties have been performed, it is not possible to obtain information on how the duties have been adjusted in response to a single disruption.

4.7 Conclusions and future research

We presented two approaches to solve railway crew rescheduling with retiming. We compared our new approaches INER and ECPR with the existing method INE of Potthoff et al. (2010) that does not allow retiming. In 4 out of 6 cases (*Ht:1* and *Ztm:1*, both with and without standby drivers), the new approaches found solutions with less canceled tasks. Moreover, the total amount of delay that was introduced into the timetable by retiming is very small, which makes it likely that these solutions can be implemented in practice. The computation times of the INER approach are within a range that should make this method applicable within a decision support system for disruption management.

In this paper we have limited ourselves by considering only the train drivers. However, in a disrupted situation, conductors need to be rescheduled at the same time. This can be done as in Stojković and Soumis (2005) and Abdelghany et al. (2008) by using multiple tasks per trip. These tasks represent the different functions of the crew members. Here the delays and the cancelations of the tasks that are connected with the same trip must be synchronized in an adequate way.

Conflicts between trains due to retiming decisions should be taken into account as well. In our future research we will extend the presented model and solution approaches into that direction. Furthermore, as was indicated already in the paper, crew rescheduling is linked with rolling stock rescheduling. Therefore, integrating the crew rescheduling process with the rolling stock rescheduling process is a relevant subject for further research. This should be accomplished in such a way that the computation times remain short, but at this moment this seems to be still too complex.

Disruption management takes place in a highly uncertain environment. Therefore it can only be *estimated* how long it will take e.g. until a broken switch has been repaired. This means that, at the point in time when the first rescheduling decisions must be made, it is not certain for how long the timetable will be adjusted during the rest of the day. Therefore the rescheduling process of the timetable, the rolling stock and the crew duties may have to be carried out several times, possibly with a rolling horizon, if the duration of the disruption turns out to be different than the initial estimate. New models and algorithms that take the uncertainty in the duration of the disruption into account are subject for further research.

Chapter 5

A Quasi-Robust Optimization Approach for Crew Rescheduling

This paper started by work of Daniel Potthoff on railway crew rescheduling under uncertainty in his PhD thesis (Potthoff (2010)). The model is extended by adding more flexibility in the level of robustness of the resulting crew schedule. This chapter has been accepted for publication in *Transportation Science* (**Veelenturf** et al. (forthcoming)). The research in this paper has been granted in 2012 a second place in the Student Paper Award Competition of the Railway Application Section of INFORMS.

Co-authors: D. Potthoff, D. Huisman, L.G. Kroon, G. Maróti, A.P.M. Wagelmans

5.1 Introduction

Transportation systems of bus, rail or air traffic often have to deal with disruptions. For example, weather conditions, accidents, and malfunctioning infrastructure or vehicles may block the traffic at a certain location for a certain period of time. As a consequence, the timetable as well as the schedules for vehicles and crews cannot be executed as planned: They have to be rescheduled.

Large-scale disruptions generally require substantial rescheduling steps (as opposed to minor disruptions due to small delays). This chapter focuses on such rescheduling problems due to large-scale disruptions where the underlying infrastructure becomes temporarily unavailable. We describe the problem setting and our solution approach in the context of railway crew rescheduling, more specifically: for rescheduling train drivers. We want to emphasize, though, that the ideas are directly applicable in a broader class of service or production scheduling settings where timetabled and location dependent tasks are to be carried out by a given number of servers (e.g. real-life vehicle, crew and machine scheduling problems).

The operation of a railway system is based on an extensive planning process, resulting in a timetable and schedules for the rolling stock and crews. The interested reader can find further details about the underlying planning problems in Abbink et al. (2005) and Kroon et al. (2009).

Effective disruption management is key to a good operational performance of a train operating company. We refer to Jespersen-Groth et al. (2009) for a detailed description of the disruption management process. Decision support for rescheduling the crews is particularly important since the crew duties are subject to complex rules and regulations. In what follows, the problem of rescheduling the crew duties in a disrupted situation is called the Operational Crew Rescheduling Problem (OCRSP).

Potthoff et al. (2010) proposed an approach for solving OCRSP. However, this approach assumes that an accurate estimate of the duration of the disruption is available at the time the rescheduling is carried out. The same holds for the approach of Rezanova and Ryan (2010) and for the models developed for crew rescheduling in the airline industry. We refer to Clausen et al. (2010) for an overview of crew rescheduling models in the airline industry.

The main goal of crew scheduling is to cover a given set of *tasks* by a certain number of crews. The tasks are characterized by their start time, end time, start location and end locations. A *duty* is an ordered sequence of tasks that can be assigned to a single crew.

A particularly challenging issue in real-time rescheduling is the uncertainty about the duration of the disruption. For example, recovery works on a broken switch in a railway network may take two hours in the optimistic scenario, but they may stretch up to four hours in the pessimistic scenario. The traffic is interrupted during the recovery works, leading to cancelled tasks for rolling stock and crews. It is unclear at the start of the disruption how many tasks will have to be cancelled.

A common approach to tackle this uncertainty is to modify the schedule at the start of the disruption based on an estimated duration of the disruption. Usually, the initial estimate is the optimistic scenario: the shortest possible duration. Later, when it turns out that the disruption lasts longer than initially estimated, the schedule is modified again and again. This approach is also called wait-and-see.

Existing algorithmic frameworks for dealing with uncertainty include the classical approaches of *robust optimization* and *stochastic programming*:

- *Robust optimization* tries to find the best solution that remains feasible under all specified scenarios without applying any modifications or recovery actions. For more information about robust optimization we refer to Bertsimas and Sim (2003) and Ben-Tal and Nemirovski (2002).

- Two-stage *stochastic programming* minimizes the sum of the first stage costs and the expected costs of the recovery in the second stage. An important assumption is that the probability for the occurrence of each of the considered scenarios is known a priori. For more information about stochastic programming we refer to Birge and Louveaux (1997) and Kall and Wallace (1994).

Both robust optimization and stochastic programming lead to significantly more complex optimization problems than the underlying deterministic problems. In most cases, realistic instances cannot be solved in (near) real-time. In addition, robust optimization is very conservative, while stochastic programming needs information about a probability distribution for the occurrence of the different scenarios, which is usually not available in practice.

Liebchen et al. (2009) introduced the concept of *recoverable robustness* which aims at finding a first stage solution that, in the second stage, can easily be turned into feasible solution no matter which scenario takes place. Cacchiani et al. (2012) study recoverable robust railway rolling stock planning. Cacchiani and Toth (2012) provide a survey of different robustness notions in the context of railway timetabling.

In this paper we propose a *quasi-robust rescheduling approach*; it is built upon the concept of recoverable robustness. The main idea is to compute a good schedule for the optimistic scenario in such a way that it can easily be turned into a feasible schedule in any other scenario. This is achieved by requiring in the first stage that a given number of the rescheduled duties must have an alternative for tasks for which it is not certain whether they must be carried out. Thus, if such tasks turn out to be cancelled in the realized scenario, then these duties can easily be made feasible again. Duties with this property are called *recoverable*. A more detailed definition of this concept is provided in Section 5.2.3

Recoverability of a duty is a local property: It depends on the duty itself and not on the entire solution. It is therefore rather easy to incorporate it in column generation based algorithms without substantially raising their running time. Furthermore, the approach admits to balance the robustness and the operational costs by requiring a *given* number of the rescheduled duties to be recoverable.

Our method consists of two stages. In the first stage we assume that the optimistic scenario takes place, and we compute the modified schedule subject to a constraint that a given number of rescheduled duties must be recoverable. Then, in the second stage, when it turns out that another scenario than the optimistic scenario is realized, we compute the rescheduled duties from scratch. That is, we do not limit the recovery action in the second stage to simply falling back to the recovery alternatives of the recoverable duties. Computing the rescheduled duties from scratch in the second stage will be necessary when not all duties are recoverable after the first stage. Moreover, rescheduling from scratch in the second stage also helps to reduce the second stage costs.

The primary criterion for assessing the quality of a schedule is the number of additionally cancelled tasks (i.e., the ones that are cancelled on top of those that are cancelled due to the disruption), both in the first stage and in the second stage. The two-stage evaluation framework allows us to analyze how the robustness requirements in the first stage influence the actual rescheduling performance in the pessimistic scenario or in any other scenario. We also compare our approach with a typical rolling horizon approach where initially only the optimistic duration of the disruption is taken into account, and where the duties are rescheduled whenever the information about the duration of the disruption is updated.

We demonstrate the effectiveness of our approach by the results of the computational tests that were carried out on a number of railway crew rescheduling instances of Netherlands Railways (NS), the main operator of passenger trains in the Netherlands.

The contributions of this paper are summarized as follows.

- We consider disruption management of crew scheduling under uncertainty.
- We develop a framework to deal with the uncertainty about the duration of a disruption.
- We evaluate our approach on realistic railway crew rescheduling instances of Netherlands Railways.

We want to emphasize that our focus lies both on developing new methods and on practical applications. We consider real-time disruption management of substantially complex scheduling systems. In fact, our computational tests are based on railway crew rescheduling instances that are quite challenging, even without taking into account the uncertainty about the duration of the disruption. We focus on rescheduling the duties of the train drivers, but the approach is also applicable for rescheduling the duties of conductors

This paper is organized as follows. In Section 5.2, we give a description of our quasi-robust rescheduling approach and of the uncertainty that has to be dealt with. We also give a formal definition of the concept of quasi-robustness. Section 5.3 presents our computational results.

This paper is concluded in Section 5.4 with suggestions for further research. Appendix 5.5 gives a detailed example of a crew rescheduling instance of NS.

5.2 Quasi-robust optimization approach

In Section 5.2.1 we first discuss the deterministic crew rescheduling problem. Then in Section 5.2.2 the framework of rescheduling under uncertainty is provided, and in Section 5.2.3 the definitions are formalized. The solution approach is discussed in Section 5.2.4, and Section 5.2.5 concludes with how the approach is tested on crew rescheduling problems of NS.

5.2.1 Crew rescheduling problems

Crew scheduling problems can be seen as a set of timetabled *tasks* which must be carried out by a number of crews. In the crew scheduling problems that we consider, each task has a fixed start and end time and a given start and end location. A driver's task can correspond to driving a train or to travelling (as passenger) on a train from one station to another.

A sequence of tasks to be carried out by a single driver is called a *duty*. If a task is carried out by a certain driver, we say that the task is *covered* by that driver. Each driver belongs to a crew base: his or her assigned duty must start and end at that crew base.

If a disruption occurs, a number of tasks must be cancelled due to the unavailability of the infrastructure or vehicles. As a consequence, some of the original duties become infeasible and must be rescheduled. In such an operational crew rescheduling problem (OCRSP), the duties must be modified such that as many as possible of the remaining tasks are covered by a driver, and such that the modifications of the duties are minimal.

In this section we first assume that the disruption starts at time τ_1 and that the duration of the disruption is known. Thus the set of remaining tasks that still have to be carried out is known at time τ_1. In Section 5.2.2 we relax this assumption.

The *completion* for a driver is a new feasible sequence of tasks to replace the original duty which starts at τ_1. The completion only consists of non-cancelled tasks. Preferably, the completion covers the same tasks as the original duty and ends at the same time. The concept of completions is illustrated in the example described in Appendix 5.5.

We use the following notations.

- T: The set of tasks that have not started yet at the time of rescheduling (i.e., at time τ_1), and that, given the duration of the disruption, are still to be carried out.

- Δ: The set of drivers whose original duty has not finished by time τ_1.

- K^δ: The set of all feasible completions for driver $\delta \in \Delta$. For every feasible completion $k \in K^\delta$ we have:
  - c_k^δ: The cost of completion k for driver δ. The cost of a completion is zero if the original duty of the driver is not modified.
  - $a_{ik}^\delta \in \{0,1\}$ to indicate whether task i is covered by completion k for driver δ.
- f_i: The cost for not covering a task i.

Given these definitions, we can formulate the OCRSP with a given duration of the disruption as a Mixed Integer Program (MIP). This MIP is an adapted version of a Set Covering model. The model uses binary variables x_k^δ to represent whether or not completion k is selected for driver δ. Furthermore, binary variable $z_i = 0$ if task i is covered and $z_i = 1$ if task i is not covered by any of the selected completions. The model is formulated as follows.

$$
\begin{aligned}
(OCRSP): \min \quad & \sum_{\delta \in \Delta} \sum_{k \in K^\delta} c_k^\delta x_k^\delta + \sum_{i \in T} f_i z_i && && (5.1)\\
\text{s.t.} \quad & \sum_{\delta \in \Delta} \sum_{k \in K^\delta} a_{ik}^\delta x_k^\delta + z_i \geq 1 && \forall i \in T && (5.2)\\
& \sum_{k \in K^\delta} x_k^\delta = 1 && \forall \delta \in \Delta && (5.3)\\
& x_k^\delta, z_i \in \{0,1\} && \forall \delta \in \Delta, \forall k \in K^\delta, \forall i \in T && (5.4)
\end{aligned}
$$

Here the objective function (5.1) describes the aim of minimizing the sum of the costs of the completions and the costs of not covering certain tasks. Constraints (5.2) make sure that every task is either covered by a completion or it is marked as not covered. Constraints (5.3) ensure that every driver is assigned exactly one completion. Constraints (5.4) describe the binary character of the decision variables. The model for this rescheduling problem differs from a standard scheduling (set covering) model since the completions are different for each driver depending on the tasks the driver has already performed. For example, the maximum end time and end location are fixed for each driver.

5.2.2 Rescheduling under uncertainty

In contrast with the assumption in Section 5.2.1, the duration of the disruption is usually not known at the start time τ_1 of the disruption. In this paper we deal with this uncertainty about the duration of the disruption by considering the rescheduling problem as a two-stage optimization problem.

The first stage rescheduling is carried out at time τ_1. At time τ_1, an optimistic estimate $\underline{\tau}$ and a pessimistic estimate $\bar{\tau}$ of the end time of the disruption are assumed to be known. Later, at time τ_2, with $\tau_1 < \tau_2 \leq \underline{\tau}$, the actual end time of the disruption τ (with $\underline{\tau} \leq \tau \leq \bar{\tau}$) becomes known. Time τ_2 is the time at which the second stage rescheduling is carried out.

Without loss of generality we can restrict ourselves to a finite set S of scenarios. Indeed, each scenario is characterized by the subset of tasks that are to be covered, and the number of such subsets is finite.

The optimistic scenario $\underline{s}$ and the pessimistic scenario $\bar{s}$ correspond to the optimistic and the pessimistic end time of the disruption, respectively. In the optimistic scenario the set of tasks $T_{\underline{s}}$ must be carried out, and in the pessimistic scenario the smaller set of tasks $T_{\bar{s}}$ must be carried out. The tasks in the set $C := T_{\underline{s}} \setminus T_{\bar{s}}$ are called the *critical* tasks.

All other scenarios are obtained by cancelling a number of critical tasks from the set C. The set of tasks that must be carried out in scenario s is denoted by T_s. We assume that the scenarios have been ordered in such a way that $T_{s_2} \subset T_{s_1}$ if the disruption ends earlier in scenario s_1 than in scenario s_2.

Then the rescheduling problem under uncertainty is stated as follows. Given the set of possible scenarios S, find (at time τ_1) a new schedule valid for the optimistic scenario $\underline{s}$ that minimizes the sum of the costs of this new schedule and the worst case costs for the additional rescheduling in the second stage (at time τ_2). Note that, since we do not assume knowledge of a probability distribution of the scenarios, we cannot minimize the expected costs. The objective function is explained in more detail in Section 5.2.3.

5.2.3 Definitions

In this section we define the concept of q-quasi-robustness. First we give an informal description. The idea behind q-quasi-robust optimization is to generate completions for q drivers in a way that is, in some sense, robust against all possible scenarios, i.e., the completions can be carried out no matter which scenario is actually realized. By doing so, we aim at minimizing the rescheduling costs (in particular the number of cancelled tasks) in the second stage (at time τ_2) if a scenario other than the optimistic scenario $\underline{s}$ is realized. We call our approach quasi-robust rather than robust because we do not require *all* completions to be robust.

Now we proceed with the formal definition of q-quasi-robustness in three steps in Definitions 1, 2 and 3.

Definition 1. *Let k and γ_s be completions for driver δ. Suppose that k is feasible in the first stage OCRSP and that γ_s is feasible in the second stage OCRSP when scenario s is realized. Then γ_s is said to be a* ***recovery alternative*** *for completion k in scenario s if $a^{\delta}_{i\gamma_s}$ = 1 holds for each task $i \in T_s$ with $a^{\delta}_{ik} = 1$.*

In words, γ_s is a recovery alternative for k in scenario s if each task $i \in T_s$ that is covered by completion k is also covered by completion γ_s. Informally speaking, γ_s circumvents the critical tasks of k that are cancelled in scenario s, while all non-critical tasks in k are still covered by γ_s. We observed in our tests that completions with recovery alternatives often had to contain a certain amount of idle time around (i.e., before and/or after) a critical task.

Now a recoverable completion is defined as follows.

Definition 2. *Let k be a completion for driver δ and suppose that k is feasible in the first stage OCRSP for the optimistic scenario $\underline{s}$. Then k is called* ***recoverable*** *if*

- *k does not contain two critical tasks directly after each other, and*
- *there exists a recovery alternative γ_s for completion k in each scenario $s \in S$.*

Note that, by Definition 2, a completion without any critical task is recoverable. Note further that a recoverable completion may contain more than one critical task, but only if there is at least one non-critical task between each pair of critical tasks. We use this restriction to make the alternative path for each critical task independent of the other critical tasks and thereby also of the scenario. If two critical tasks are performed directly after each other, this property does not hold. For example, then the first critical task needs an alternative path including the second critical task (for scenarios in which the second critical task is not cancelled) and an alternative path excluding both critical tasks (for scenarios in which both critical tasks are cancelled).

Definition 3. *Let q be a non-negative integer. A schedule obtained in the first-stage rescheduling phase is called* ***q-quasi robust*** *if at least q drivers have a recoverable completion.*

Based on the above Definitions 1, 2 and 3, we next describe the q-quasi-robust rescheduling problem (q-QRSP) that is to be solved in the first-stage rescheduling phase. We denote the set of recoverable completions for driver δ by $R^{\delta} \subset K^{\delta}$. Furthermore, for each driver δ and each feasible completion k of δ, the binary decision variable x^{δ}_{k} describes whether or not completion $k \in K^{\delta}$ is selected in the solution. For each task $i \in T_{\underline{s}}$, the binary variable $z_i = 0$ if i is covered and $z_i = 1$ if i is not covered by the selected completions. Now we state q-QRSP in the first stage as follows.

$$\min \quad \sum_{\delta \in \Delta} \sum_{k \in K^\delta} c_k^\delta x_k^\delta + \sum_{i \in T_{\underline{s}}} f_i z_i \tag{5.5}$$

$$\text{s.t.} \quad \sum_{\delta \in \Delta} \sum_{k \in K^\delta} a_{ik}^\delta x_k^\delta + z_i \geq 1 \qquad \forall i \in T_{\underline{s}} \tag{5.6}$$

$$\sum_{k \in K^\delta} x_k^\delta = 1 \qquad \forall \delta \in \Delta \tag{5.7}$$

$$\sum_{\delta \in \Delta} \sum_{k \in R^\delta} x_k^\delta \geq q \tag{5.8}$$

$$x_k^\delta, z_i \in \{0,1\} \qquad \forall \delta \in \Delta, \forall k \in R^\delta, \forall i \in T_{\underline{s}} \tag{5.9}$$

The objective function (5.5) describes that the aim is to minimize the sum of the costs of the selected completions and the costs of leaving certain tasks not covered. Constraints (5.6) specify that each task $i \in T_{\underline{s}}$ must be covered by a completion or it must be marked as not covered. Constraints (5.7) describe that each driver must get a completion. Constraints (5.8) determine that at least q drivers must get a recoverable completion. Finally, constraints (5.9) require the decision variables to be binary valued.

The model (5.5)–(5.9) for q-QRSP in the first stage is very similar to the model (5.1)–(5.4) for OCRSP. The difference is that in (5.5)–(5.9) we require at least q drivers to have a recoverable completion. Note that if $q = |\Delta|$, then all drivers must have a recoverable completion. That implies that in each scenario a feasible solution can be obtained by using the corresponding recovery alternative for each driver. On the other hand if $q = 0$, then no driver needs a recoverable completion, which means that no robustness at all is taken into account.

If another scenario is realized than the optimistic one, then the second stage amounts to solving an OCRSP instance. Recall that the exact duration of the disruption is known when the second stage problem is solved.

5.2.4 Solution approach

The solution approach for q-QRSP in the first stage consists of a combination of Lagrangian relaxation and column generation, and is based on Caprara et al. (1999), Huisman et al. (2005) and Potthoff et al. (2010). If the problem contains many tasks, then drivers can have a huge number of feasible completions. Therefore we use a column generation approach where only promising completions are considered.

Lagrangian relaxation

In the master problem for q-QRSP in the first stage, Constraints (5.6) are relaxed which results in the following Lagrangian subproblem.

$$\Theta(\lambda) = \min \sum_{\delta \in \Delta} \sum_{k \in K^\delta} c_k^\delta x_k^\delta + \sum_{i \in T_{\underline{s}}} f_i z_i + \sum_{i \in T_{\underline{s}}} \lambda_i (1 - \sum_{\delta \in \Delta} \sum_{k \in K^\delta} a_{ik}^\delta x_k^\delta - z_i) \tag{5.10}$$

$$\text{s.t.} \quad \sum_{k \in K^\delta} x_k^\delta = 1 \qquad \forall \delta \in \Delta \tag{5.11}$$

$$\sum_{\delta \in \Delta} \sum_{k \in R^\delta} x_k^\delta \geq q \tag{5.12}$$

$$x_k^\delta, z_i \in \{0, 1\} \qquad \forall \delta \in \Delta, \forall k \in R^\delta, \forall i \in T_{\underline{s}}, \tag{5.13}$$

The latter can be rewritten as follows.

$$\Theta(\lambda) = \min \quad \sum_{i \in T_{\underline{s}}} \lambda_i + \sum_{\delta \in \Delta} \sum_{k \in K^\delta} (c_k^\delta - \sum_{i \in T_{\underline{s}}} \lambda_i a_{ik}^\delta) x_k^\delta + \sum_{i \in T_{\underline{s}}} (f_i - \lambda_i) z_i$$

$$\text{s.t.} \quad (5.11) - (5.13)$$

For given Lagrange multipliers λ_i, the Lagrangian subproblem can be solved in the following way. Let $z_i = 1$ if $f_i - \lambda_i < 0$, and let $z_i = 0$ otherwise. To determine the x_k^δ values, we have to choose at least q drivers which must have a recoverable completion. This can be accomplished as follows.

First, we define $\bar{c}_k^\delta = c_k^\delta - \sum_{i \in T_{\bar{s}}} \lambda_i a_{ik}^\delta$, $k^\delta = \arg\min\{\bar{c}_k^\delta \mid k \in K^\delta \backslash R^\delta\}$ and $r^\delta = \arg\min\{\bar{c}_k^\delta \mid k \in R^\delta\}$. Thus k^δ represents the completion with the lowest reduced costs of all non recoverable completions for driver δ, and r^δ represents the completion with the lowest reduced costs of all recoverable completions for driver δ. For every driver either of these completions must be selected. Based on this information we can determine the optimal values for the x_k^δ variables by the following steps.

1. Set all x_k^δ variables equal to 0.
2. Compute for every driver δ the difference between the reduced costs of the two completions $c^{\delta,*} = \bar{c}_{r^\delta}^\delta - \bar{c}_{k^\delta}^\delta$.
3. For the q drivers δ with the lowest values of $c^{\delta,*}$, set $x_{r^\delta}^\delta = 1$.
4. For all remaining drivers, set $x_{r^\delta}^\delta = 1$ if $c^{\delta,*} \leq 0$. Otherwise set $x_{k^\delta}^\delta = 1$.

Restricted master problem

Since we use column generation, we consider a restricted master problem (RMP) of (5.10)-(5.13) containing only a subset of the x_k^δ variables. In our implementation, this subset always contains at least one recoverable completion for every driver (but not necessarily a non-recoverable completion). For example, a recoverable completion for a driver that is feasible in each scenario is to end its duty without carrying out any further tasks.

In the n^{th} column generation iteration, the x_k^δ variables in the RMP are given by $\cup_{\delta\in\Delta}\{x_k^\delta : k \in K_n^\delta\}$, where $K_n^\delta \subseteq K^\delta$ is a subset of completions for driver δ. Further, let $R_n^\delta = K_n^\delta \cap R^\delta$ be the recoverable completions for driver δ in iteration n. As mentioned, $R_n^\delta \neq \emptyset$.

Let Θ_n^* be the optimal value of the Lagrangian subproblem of the n^{th} column generation iteration. We use subgradient optimization to compute an approximate value A_n^* for Θ_n^* satisfying $A_n^* \leq \Theta_n^*$. Let λ_n^* be the corresponding multiplier vector. We solve a pricing problem for every driver $\delta \in \Delta$ to check if A_n^* is a good approximation of Θ^*. If it is not, we need to add more completions to the RMP in order to improve the solution. For the details of this procedure we refer to Potthoff et al. (2010).

The pricing problems are modeled as shortest path problems with resource constraints (SPPRC) in dedicated graphs. These graphs are introduced later. Let $u_n^\delta = \min\{\bar{c}_k^\delta(\lambda_n^*) : k \in R_n^\delta\}$ and $v_n^\delta = \min\{\bar{c}_k^\delta(\lambda_n^*) : k \in K_n^\delta\}$ be the smallest Lagrangian reduced costs of the already generated recoverable and general completions, respectively. Furthermore, let $s_n^\delta = \min\{\bar{c}_k^\delta(\lambda_n^*) : k \in R^\delta\}$ and $t_n^\delta = \min\{\bar{c}_k^\delta(\lambda_n^*) : k \in K^\delta\}$ be the optimal values of the recoverable and the general pricing problem for driver δ, respectively. Then the completions corresponding to u_n^δ and v_n^δ should be added to the RMP if $s_n^\delta - u_n^\delta < 0$ and $t_n^\delta - v_n^\delta < 0$.

The pricing problem

The pricing problem for a driver δ can be modeled as a shortest path problem with resource constraints (SPPRC) in the pricing problem graph. We follow here the graph model described by Potthoff et al. (2010). The resources are required to handle additional application-specific properties of the duties. For example, labor rules at NS have limitations on the lengths of the parts of a duty before and after the meal break.

In a pricing problem graph, a node represents the start or the end of a task. Arcs are used to represent the tasks and to indicate which tasks can follow each other. The latter depends on the scheduled time as well as on the start and end locations of the tasks.

Example 4. *Figure 5.1 shows an example of a pricing problem graph involving tasks g, h, i, j, l, m, and n. The bold arcs correspond to the tasks. The thin arcs indicate transfers from one task to another.*

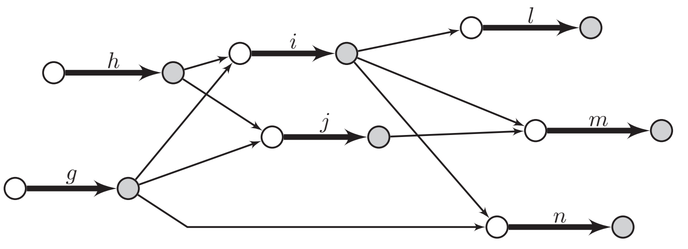

Figure 5.1: A pricing problem graph involving tasks g, h, i, j, l, m, and n.

By construction, any feasible completion corresponds to a path in a pricing problem graph. However, the reverse is not true in general, due to complex labor rules such as the aforementioned restriction on meal breaks. These rules are handled by the resource constraints.

Task i is said to be covered directly after task h in a completion if task h is followed directly by task i in the corresponding path. In this case, task i is a *successor* of task h, and task h is a *predecessor* of task i.

Finding recoverable completions Completions for a driver can be generated based on regular pricing problem graphs as shown in Figure 5.1. However, for generating only *recoverable* completions, we have to modify the pricing problem graphs in order to guarantee the existence of a recovery alternative for completions containing critical tasks. In other words, when constructing a completion, we have to guarantee that for each critical task in the completion also an alternative path is available that can be used in case the critical task is cancelled. Here we use the following lemma.

Lemma 5. *Let k be a feasible recoverable completion for driver δ, and let i be any critical task in k. Then there exists a path, consisting of non-critical tasks only, from the end node of the non-critical predecessor of task i to the start node of the non-critical successor of task i.*

Proof. If completion k is recoverable, then it does not contain two critical tasks directly after each other. Thus each critical task i in k has a non-critical predecessor and a non-critical successor. Now the claim follows from the fact that completion k should have a recovery alternative in the pessimistic scenario $\bar{s}$, in which all critical tasks have been cancelled. In this pessimistic scenario the non-critical predecessor of task i and the non-critical successor of task i still have to be carried out, but task i is cancelled. Thus there exists a path as indicated. □

Lemma 5 easily extends to the case when a critical task is the first task of completion k. Then the end node of its predecessor is replaced by a dummy node representing the start of the completion. Similarly, if a critical task is the last task of completion k, then the start node of its successor is replaced by a dummy node representing the end of the completion.

Note that the relation in Lemma 5 is not an equivalence: The existence of a path does not imply the existence of a feasible completion, due to the resource constraints.

Based on Lemma 5 we propose an algorithm to generate the recoverable completions. The algorithm consists of thee steps: (*i*) finding alternative paths, (*ii*) modifying the pricing problem graph, and (*iii*) considering additional resources and solving the SPPRC.

For the third step, we introduce additional arc properties next to the costs. These arc properties are used to define so-called Resource Extension Functions. For each scenario, a separate Resource Extension Function is needed to check whether a generated completion is feasible in the corresponding scenario. For the details of the Resource Extension Functions, we refer to Potthoff (2010).

Finding alternative paths We have to find out for each critical task i whether there exists an alternative path in the pricing problem graph consisting of non-critical tasks only that can be used to replace critical task i if this task is cancelled.

To that end, we remove all arcs corresponding to the critical tasks from the graph. In the reduced graph we can use a shortest path algorithm to determine, for every *non-critical* predecessor task h of critical task i, all *non-critical* successor tasks of critical task i that can be reached from task h via a path consisting of non-critical arcs only. In case of a SPPRC we are not interested in reachability alone, but also in information about *how* a successor node can be reached. To be more precise, we would like to find the path from the non-critical predecessor task h to the non-critical successor task j which uses the fewest resources.

Modifying the pricing problem graph Next we go back to the original pricing problem graph, and we modify it as follows. Let task i be a critical task. Then the arc corresponding to this task is removed from the graph. Consider any *non-critical* predecessor task h of critical task i and consider any *non-critical* successor task m of critical task i that can be reached from h via a path consisting of non-critical arcs only (as described in the previous paragraph). Then we insert a copy i' (with two new nodes) of the just removed arc, and we connect i' to h and m in such a way that h, i' and m appear in a directed path.

These steps are carried out for each critical task. The procedure is illustrated in Example 6 and Figure 5.2. Note that we consider here only non-critical predecessors and non-critical

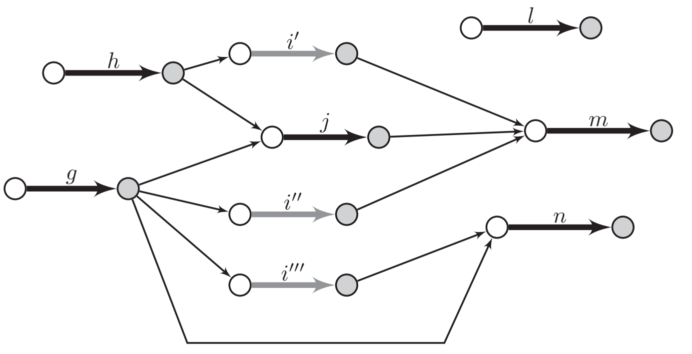

Figure 5.2: Part of a pricing problem graph *after* the preprocessing step for critical task i that is cancelled in scenario $\overline{s}$ has been carried out.

successors in order to avoid the occurrence of two critical tasks directly after each other in a completion.

Example 6. *Figure 5.2 shows the modified pricing problem graph that is obtained when the preprocessing steps are applied to the pricing problem graph shown in Figure 5.1. Here task i is a critical task that is carried out in scenario $\underline{s}$ and cancelled in scenario $\overline{s}$. Task i has two predecessors $\{h, g\}$, and three successors $\{l, m, n\}$. From task h, only task m can be reached via task j in the auxiliary problem when critical task i has been removed. For this relation we introduce a new task i' and the necessary arcs. From task g, tasks m and n can be reached in the auxiliary problem, which is represented by the copies i'' and i''' of critical task i. Note that task l cannot be reached from any predecessor of task i. Therefore, task l cannot be covered by any recoverable completion.*

Lemma 7. *Suppose that feasible completion k is obtained as a resource constrained shortest path in the graph constructed according to the above described procedure. Then k is recoverable.*

Proof. If the feasible completion k does not contain any critical task, then it is recoverable by definition.

If completion k contains exactly one critical task i, then task i is not preceded nor succeeded in k directly by another critical task. Thus the alternative path for critical task i that was determined in the described procedure fits between the non-critical predecessor of task i in k to the non-critical successor of task i in k.

If completion k contains more than one critical task, then, due to the construction of the graph, two critical tasks in k do not follow each other directly in k. As a consequence, their alternative paths do not interact with each other. They are separated from each other in time by at least one non-critical task.

The foregoing cases imply that, if task i is a critical task, then task i can be replaced by its alternative path in any scenario *not* containing task i. Clearly, task i can be carried out as planned in any scenario containing task i.

Finally, the Resource Extension Functions per scenario are used to handle the resource consumption in each scenario. As a consequence, completion k is a feasible completion in the first stage under the optimistic scenario. If any other scenario than the optimistic one is realized, then the corresponding recovery alternative for completion k is a feasible completion in the second stage. □

It is clear from the above example that the number of nodes and arcs in the pricing problem graphs increase significantly if many successors of the arrival node of a critical task can be reached from many predecessors of the departure node of the critical task. This has consequences for applying the concept of recoverability on instances of practical relevance.

5.2.5 Implementation for crew rescheduling problems of NS

The method proposed in this paper is an extension of the approach of Potthoff et al. (2010) for solving OCRSP. Since real-life applications need results within a few minutes, Potthoff et al. (2010) do not aim at rescheduling *all* duties, but only a subset of them. This subset is updated as long as there are still tasks which are not covered. The initial subset consists of the duties that are directly affected by the disruption (these duties *must* be rescheduled), but also of a set of heuristically chosen additional duties that may help to find better solutions.

Our approach selects the subset of duties in the same manner as Potthoff et al. (2010) chooses the initial subset. We adapt the master and pricing problem of Potthoff et al. (2010) to handle the recoverability. Especially in the pricing problems we have to keep track of more resource constraints since we have to ensure that also in the recovery alternatives meal break rules are not violated.

In the first stage, our approach ensures for crews that need a recoverable completion that there is a recovery alternative if another scenario than the optimistic scenario is realized. From all feasible recovery alternatives, this approach picks the one which consumes the fewest resources. Such an alternative does not have to be the cheapest alternative. Therefore we reschedule again in the second stage to search for the cheapest alternative.

For rescheduling in the second stage we use the method of Potthoff et al. (2010) to solve OCRSP, since we assume that the duration of the disruption is known at that time. Note that we can only skip the rescheduling in the second stage if in the first stage *all* crews have a recoverable completion. However, since we do not optimize on what the recovery alternative looks like, rescheduling in the second stage can still reduce the second stage rescheduling costs.

5.3 Computational results

In this section we report our computational results for the q-quasi-robust optimization approach for crew rescheduling under uncertainty. We test the method on realistic crew rescheduling instances of NS.

In order to explore the balance between robustness and nominal costs, every case is solved multiple times. We start with the instance where no completion is required to be recoverable, which corresponds with the wait-and-see approach. Then we gradually increase the number of completions that must be recoverable, until we finally reach the point where all completions are required to be recoverable.

Since we are using a heuristic to solve the problem it may happen that we do not find an optimal solution for an instance. Due to this phenomenon, it may happen that a solution of the first stage where q_1 crews need a recoverable completion has lower costs than a solution where q_2 crews need a recoverable completion, even though $q_1 > q_2$. In such a case we use in our results the solution of q_1 for all values of q with $q_2 \leq q \leq q_1$.

It is worthwhile to compare the two extreme cases. If no crew needs a recoverable completion, we just solve the underlying OCRSP for the optimistic scenario without any robustness requirements. This may lead to high rescheduling costs in the second stage. On the other hand, if the entire schedule is required to be quasi-robust and if we find a solution that covers all tasks, then in principle no further rescheduling steps are necessary no matter which scenario is realized.

The q-quasi-robust optimization approaches have been implemented in C++ and compiled with the Visual C++ 10.0 compiler.

5.3.1 Cases

For our computational study, we used five large-scale disruptions that actually took place in the past in the Netherlands. On a regular working day about 10,000 tasks are carried out by about 1,000 duties, about 90 of which are reserve duties. In all five cases a route becomes suddenly unavailable for 2 to 3 hours due to a disruption.

Location	ID	Time	# crews affected	# crews selected
Abcoude	Ac_A	16:30-18:30	67	142
Abcoude	Ac_B	16:30-18:30	59	116
Beilen	Bl_A	07:00-10:00	15	42
Beilen	Bl_B	16:00-19:00	15	39
’s-Hertogenbosch	Ht	08:00-11:00	55	98

Table 5.1: Summary of the different cases.

As a preparation for the crew rescheduling step, we modify the timetable according to the contingency plans currently used by NS. In particular, the timetable services on the disrupted line are cancelled. The rolling stock is rescheduled following some basic rules, which are also described in the contingency plans. The original (undisrupted) duties of the crews are taken from the operational schedules of NS on a workday in September 2007.

A brief description of the five cases is given in Table 5.1. In all of the cases railway traffic was blocked in both directions. This table considers the optimistic scenario where the disruption ends after the minimum possible duration. We note that the case Bl_A is described in detail in Appendix 5.5.

The cases around Abcoude involve a disruption of the centrally located and heavily utilized route between Utrecht and Amsterdam. The network does allow rerouting possibilities for passengers and crews, although these are time-consuming. Around 60 drivers are directly affected by the disruption in these cases. The two cases around Beilen show a disruption on a less heavily used route, but the blockage cuts off the northern part of the network from the rest. The cases around Beilen have a direct effect on 15 drivers. The case with a disruption around ’s Hertogenbosch has a big impact, since this also involves a heavily utilized route. In total 55 drivers are directly affected by this disruption.

For each of the 5 cases we consider two scenarios: we define the optimistic scenario $\underline{s}$ and the pessimistic scenario $\overline{s}$ as the scenarios corresponding to the shortest and the longest duration of the disruption, respectively. For the q-quasi-robust optimization approach, especially the number of critical tasks is important. Table 5.2 shows the main characteristics of the five cases. For every case we present the optimistic duration of the disruption, and the time the disruption lasts longer in the pessimistic scenario $\overline{s}$. We also show the number of critical tasks.

5.3.2 Objective function

The quality of a solution is measured by a combination of the operational costs and the rescheduling costs. The most important goal is that all remaining tasks are covered by the modified duties.

Case	Optimistic Duration	Considered Extension	Critical tasks in $\overline{s}$
Ac_A	2:00	0:30	8
Ac_B	2:00	0:30	8
Bl_A	3:00	1:00	6
Bl_B	3:00	1:00	5
Ht	3:00	0:30	10

Table 5.2: Information about the disruptions and the considered uncertainty.

Therefore, we account a cost of 20,000 for the additional cancellation of a task due to a missing driver.

The cost of each completion is zero if the duty is unchanged. Otherwise the cost is defined as the sum of the individual penalties depending on the way the duty is changed. We use the following values for the penalties. We account a cost of 400 for each duty that is changed anyhow. Every task that is not assigned to its original duty has a cost of 50. A cost of 1 is accounted for every transfer between two tasks that was not used in the original plan by some crew member. Finally, if a crew member has to be repositioned by using a taxi ride, the accounted cost equals 1,000. These values for the cost parameters performed best in a preliminary study.

5.3.3 Numerical results

The first stage problem of q-QRSP amounts to computing the completions for the optimistic scenario subject to the additional requirements about the number of recoverable completions. In the first stage problem of q-QRSP we consider initial core problems as described in Section 5.2.5. In order to account for the uncertainty about the duration of the disruption, we construct the initial core problems based on the optimistic duration of the disruption plus the possible extension in time.

In the second stage problem we assume that the pessimistic scenario is realized, and we solve the OCRSP using the results of the first stage problem as input. All second stage instances are solved by the algorithm presented in Potthoff et al. (2010) with the same subsets of duties as in the first stage. Note that we do *not* restrict the recovery action to the mere use of the recovery alternatives of the completions. That approach would lead to a feasible solution only if *all* completions were required to be recoverable in the first stage.

Table 5.3 shows the objective values obtained for the first and second stage problems. For every fixed value of q, denoting the number of duties that are required to have a recoverable completion, we report the results of the first stage, the second stage, and the sum of them. The outcome of the first stage indicates what happens under the optimistic scenario, while the sum of the first and second stage represents what happens under the pessimistic scenario. Note that tasks which are not covered in the first stage will not be considered and consequently not be covered in the second stage. For every solved instance we give the lower bound (*LB*), the cost of the best found solution (*UB*), and the number of cancelled tasks (*#CANC*).

First, we notice the result that more robustness requirements (i.e., a higher value of q) leads to higher first stage costs. The number of additionally cancelled tasks shows the same pattern, which can be expected since these tasks constitute the main part of the first stage costs. So we see that in the first stage more tasks are cancelled to have, most of the time, a better solution in the second stage. Cancelling tasks in the first stage also means that there will be more slack in the completions, which can be used in the second stage.

The total costs of the two stages indicate the tradeoff between costs and robustness. Especially for the cases Bl_A en Bl_B, the requirement of more and more robustness initially decreases the total costs. From a certain value of q on, however, the total costs start increasing again. That is, the robustness requirements help to decrease the second stage rescheduling costs, but too much robustness turns out to be expensive in the first stage without any further added value in the second stage.

In the other cases we have somewhat irregular behavior: a more robust schedule in the first stage may lead to higher costs in the second stage. This can happen because not all duties have a recoverable completion and then rescheduling the duties which have not yet a recovery alternative could lead to bad luck and additionally cancelled tasks in the second stage. Anyway, in all cases the second stage costs are negligible if all crews have a recoverable completion in the first stage.

For all cases, except for Ac_A, we have solutions that have the same number of cancelled tasks in the first stage as the solution where no uncertainty is taken into account ($q = 0$), but with less cancelled tasks in the second stage. Thus in these cases, at the price of some slightly higher total costs for the first stage but without additional cancellations if the optimistic scenario is realized, we can reduce the total number of cancellations if the pessimistic scenario is realized. For these cases, the instance with minimum total costs, a minimum number of cancellations if the optimistic scenario is realized, and a minimum total number of cancellations if the pessimistic scenario is realized is indicated with **bold figures** in Table 5.3.

Only for the case Ac_A, the dominating solution is realized by not taking uncertainty into account, so with $q = 0$.

To illustrate the foregoing, Figure 5.3 gives a graphical representation of the results for the case Bl_A. The black line indicates the costs of the first stage for varying levels of the number of recoverable duties q, and the dashed black line gives the corresponding lower bounds for the

Case Ac_A	Stage 1			Stage 2			Stage 1 + 2	
q	LB	UB	#CANC	LB	UB	#CANC	UB	#CANC
0-135	471,430	530,162	22	55,296	56,302	2	586,464	24
136-137	479,646	530,513	22	103,037	112,425	5	642,938	27
138	494,392	550,767	23	115,631	116,349	5	667,116	28
139-140	506,108	551,377	23	43,206	44,604	2	595,981	25
141	534,303	592,182	25	9,979	10,332	0	602,514	25
142	575,931	612,734	26	297	1,807	0	614,541	26
Case Ac_B	Stage 1			Stage 2			Stage 1 + 2	
q	LB	UB	#CANC	LB	UB	#CANC	UB	#CANC
0-109	46,182	52,486	0	55,434	58,031	2	110,517	2
110-111	43,690	54,148	0	43,852	48,484	2	102,632	2
112	38,478	55,244	0	49,085	66,529	3	121,773	3
113	**42,863**	**57,667**	**0**	**26,491**	**26,582**	**1**	**84,249**	**1**
114	42,326	63,252	0	26,969	28,840	1	92,092	1
115-116	48,309	86,567	1	3,918	3,918	0	90,485	1
Case Bl_A	Stage 1			Stage 2			Stage 1 + 2	
q	LB	UB	#CANC	LB	UB	#CANC	UB	#CANC
0-36	32,093	32,350	1	36,046	45,168	2	77,518	3
37	32,786	32,896	1	34,516	43,661	2	76,557	3
38	**31,222**	**34,653**	**1**	**22,747**	**24,110**	**1**	**58,763**	**2**
39	36,645	39,021	1	5,623	23,314	1	62,335	2
40	56,550	77,666	3	2,157	2,157	0	79,823	3
41	78,384	80,925	3	1,833	1,909	0	82,834	3
42	108,055	118,568	5	1,240	1,305	0	119,873	5
Case Bl_B	Stage 1			Stage 2			Stage 1 + 2	
q	LB	UB	#CANC	LB	UB	#CANC	UB	#CANC
0-34	46,626	58,406	2	65,453	66,511	3	124,917	5
35	**47,955**	**59,765**	**2**	**29,868**	**42,360**	**2**	**83,923**	**4**
36	56,549	76,803	3	24,114	24,114	1	83,923	4
37	73,236	78,164	3	21,876	21,908	1	103,612	4
38	93,673	99,165	4	1,405	1,405	0	102,972	4
39	117,441	119,727	5	2,609	2,609	0	121,832	5
Case Ht	Stage 1			Stage 2			Stage 1 + 2	
q	LB	UB	#CANC	LB	UB	#CANC	UB	#CANC
0-86	151,455	157,135	5	50,620	55,138	2	212,273	7
87-88	150,683	158,198	5	55,903	61,674	2	219,872	7
89	149,497	158,599	5	60,226	61,818	2	220,417	7
90	148,476	158,903	5	71,199	85,320	3	244,223	8
91	147,851	161,222	5	84,797	99,855	4	261,077	9
92	145,239	162,071	5	51,011	60,712	2	222,783	7
93	145,634	164,484	5	55,768	56,091	2	220,575	7
94	141,731	164,578	5	74,104	97,043	4	261,621	9
95	**144,067**	**164,581**	**5**	**14,698**	**15,735**	**0**	**180,316**	**5**
96	145,375	188,234	6	94,815	98,510	4	286,744	10
97	164,382	191,141	6	7,126	7,974	0	199,115	6
98	183,565	209,077	7	4,436	6,721	0	215,798	7

Table 5.3: Results of the different cases for different values of q

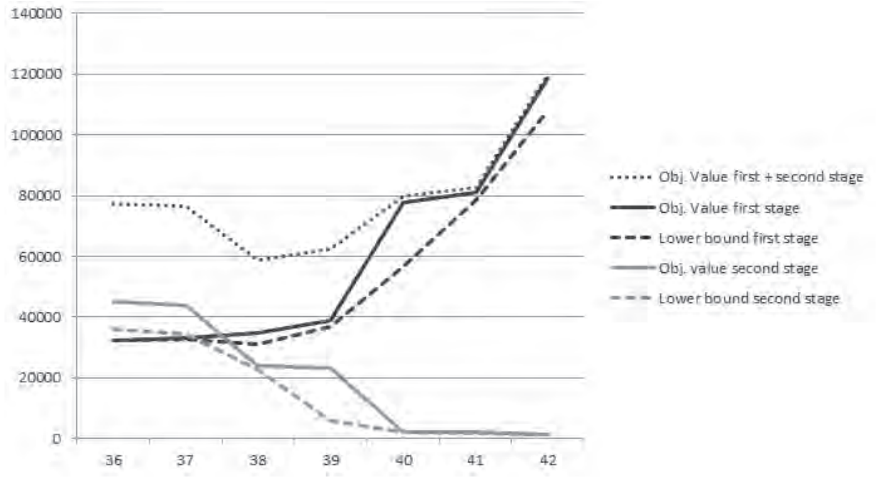

Figure 5.3: Graphical representation of the results for the case Bl_A.

costs of the first stage. The gray line indicates the costs of the second stage for varying levels of q, and the dashed gray line gives the corresponding lower bounds. The dotted line represents the total costs for the first and second stage for varying levels of q. The figure clearly indicates that the solution corresponding to $q = 38$ has only slightly higher costs for the first stage than the solution corresponding to a value of q between 0 and 36. However, for this solution the total costs for the first and second stage are significantly lower. Note that the differences between the upper and the lower bounds are small.

Summarizing, the computational results confirm the intuition that a higher degree of first stage q-quasi-robustness in general leads to higher first stage costs as well as to lower second stage costs, and that an optimal level of q-quasi-robustness can be obtained by varying the number of recoverable duties q. Most of our cases reach the lowest total costs at an intermediate robustness level: no robustness and full robustness are both inferior. Our algorithm can explore the consequences of several robustness levels, and thereby help the decision makers to find the best balance between total costs and robustness.

5.3.4 Computation times

Computation times are very important for our application, since we are dealing with an application of real-time rescheduling. The cases around Beilen can be solved very quickly. For any given value of q for the number of crews that need to have a recoverable completion, the first stage is solved within 10 seconds and the second stage is solved within 3 seconds. The other cases need longer computation times. For a given value of q, the second stage is solved within 1.5 minutes, but the first stage can take up to 3.5 minutes. These running times are promising and fast enough for a real-time application.

5.4 Concluding remarks

In this paper we study real-time resource rescheduling problems in case of large-scale disruptions. We propose a novel rescheduling approach that explicitly deals with the uncertain duration of the disruption. We introduce the concept of q-quasi-robustness, and argue why classical models (such as robust optimization and stochastic programming) are unsatisfactory for the problems we consider.

Our method is widely applicable to problems containing tasks with fixed start and end times and locations which have to be scheduled on a certain set of resources. Examples are real-life vehicle, crew and machine scheduling problems. Furthermore, the robustness requirements can easily and tractably be integrated into existing column generation models, a commonly used optimization framework for resource scheduling and rescheduling.

We demonstrate the power of our approach on real-life crew rescheduling problems of NS. Our method is able to find solutions of reasonably good quality (proven by lower bounds) in a matter of minutes. A detailed analysis shows that q-quasi-robustness reflects the intuitive notion of robustness quite well.

Besides its methodological contributions, the method has good prospects to be valuable in practice. First, computations on challenging real-life cases reliably lead to good solutions. Second, the computation times of a few minutes are close to what is needed in real-life decision making. And third, our approach is able to balance robustness requirements against operational and recovery costs. This allows decision makers to explore several variants and with different robustness levels.

5.5 Appendix: Example for a crew rescheduling problem of NS

In this section we show an example of a typical crew rescheduling instance of NS.

5.5.1 Disruption

The disruption affects the Northern part of the Netherlands. The disrupted timetable is represented in Figure 5.4 as a time-space diagram. Due to a broken catenary, no railway traffic is possible between Hoogeveen (Hgv) and Beilen (Bl) from 7:10 on. It is estimated that the repair works will last between 3 and 4 hours. The timetable is updated according to a pattern described by the contingency plan that is applicable in this situation.

In this case, the trains of the train lines 500, 700, and 9100, that are operated between Zwolle (Zl) and Groningen (Gn) (and vice versa) in an hourly periodic timetable, are turned in four intermediate stations. (The 500 line consists of train numbers between 501 and 599; a similar assumption holds for the other lines). In particular, the intercity trains of the 500 and 700 lines are turned in Hoogeveen and Assen (Asn). The regional trains of the 9100 train line are turned in Meppel (Mp) and Beilen. The corresponding trips between Hoogeveen and Assen (and vice versa) and between Meppel and Beilen (and vice versa) are cancelled.

At the intermediate stations where the trains are turned, the crew is supposed to stay with the turning trains. This means effectively that tasks from Groningen to Zwolle are changed into tasks from Groningen to Groningen, and that tasks from Zwolle to Groningen are changed into tasks from Zwolle to Zwolle. These combined tasks are so-called *rerouted* tasks. The rerouted tasks are indicated later with “/r” after their train number. Note that the concept of rerouted tasks can easily be added to the framework developed in this paper.

Figure 5.4 shows how the timetable between Zwolle and Groningen is updated. Since the repair works take at least 3 hours, the turning pattern is applied for sure for three southbound and three northbound trains of each of the three involved train lines. For the trains in the fourth hour after the start of the disruption, it is uncertain whether the trains will take their normal routes (dashed lines in Figure 5.4) or whether they will be turned as well (dotted arcs in Figure 5.4).

Traditional crew rescheduling approaches deal with this situation as follows. At time τ_1 the optimistic duration of the disruption is estimated, and the modified timetable corresponding to this estimate is used as input for OCRSP.

This means that the blockage is estimated to be resolved by 10:10. Therefore, the modified timetable that is given as input to OCRSP assumes that the trains 727, 736, 529, 538, 9129

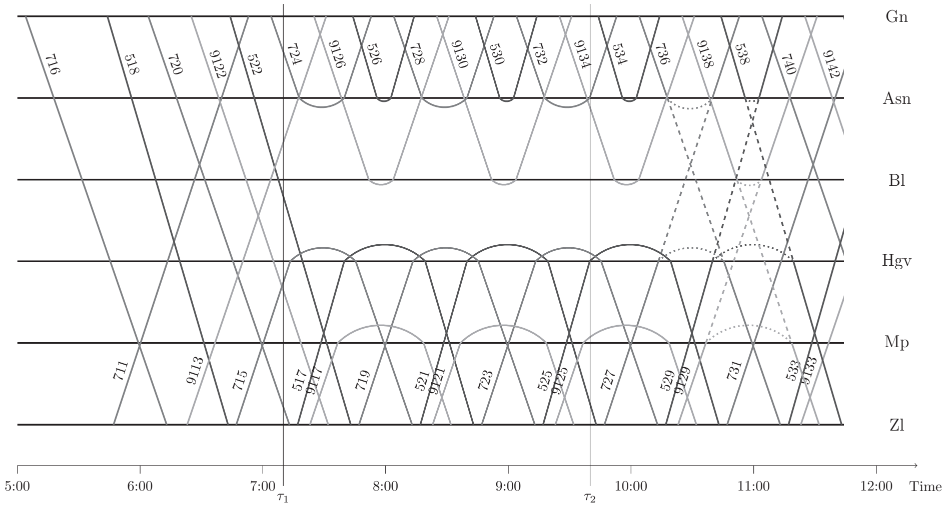

Figure 5.4: Time space diagram showing how the timetable between Groningen (Gn) and Zwolle (Zl) is updated, if the route between Beilen (Bl) and Hoogeveen (Hgv) is blocked temporarily.

and 9138 can run between Beilen and Hoogeveen as planned. Thus the corresponding tasks indicated with dashed lines in Figure 5.4 are considered in the instance of OCRSP.

However, it may happen that new information becomes available at $\tau_2 = 9{:}40$ saying that the route will be blocked until 11:10. This means that the timetable has to be updated again and that the trains 727, 736, 529, 538, 9129 and 9138 must also be turned at the intermediate stations. Thus at time τ_2 the rolling stock and crew schedules must be rescheduled as well, given this new information and the rescheduled timetable. The rerouted tasks 727/r, 736/r, 529/r, 538/r, 9129/r, and 9138/r are used as input for a second instance of OCRSP. Here, for example, task 727/r consists of a task from Zwolle to Hoogeveen in the time slot of the original task 727, followed by a return task from Hoogeveen to Zwolle in the time slot of the original task 736, see Figure 5.4. Since the two consecutive parts of task 727/r must be carried out by the same crew, the two parts together are considered as one single rerouted task.

5.5.2 Feasible completion

In this section we illustrate the concept of recovery alternatives and recoverability for the aforementioned route blockage between Hoogeveen and Beilen.

Figure 5.5a shows a planned duty from crew base Groningen (Gn). Due to the route blockage between Hoogeveen and Beilen, task 724 from Groningen to Zwolle (Zl) is rerouted and

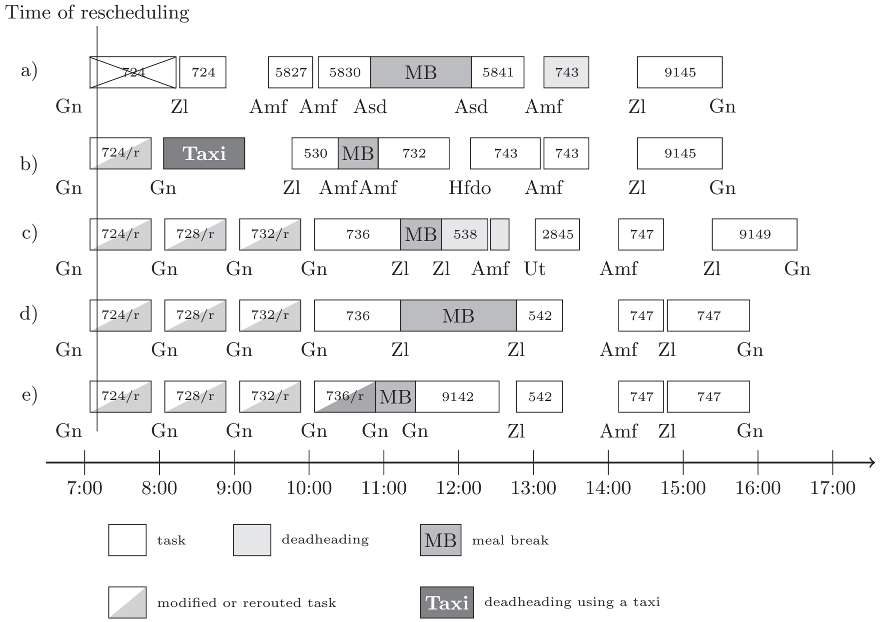

Figure 5.5: Examples of feasible completions for an affected original duty from crew base Groningen (Gn).

returns to Groningen. Therefore, the driver cannot follow his planned duty. A feasible completion of the duty under the optimistic scenario $\underline{s}$ is shown in Figure 5.5b. The optimistic scenario $\underline{s}$ assumes that the route blockage lasts until 10:10. Since this completion does not cover any critical task, it is a recoverable completion. The completion in Figure 5.5c is not recoverable. It covers critical task 736 from Groningen to Zwolle. If the pessimistic scenario $\overline{s}$ is realized, which means that the route is blocked until 11:10, then this task is rerouted (736/r) and ends in Groningen. Then the driver is not able to get to Zwolle in time to deadhead on task 538 from Zl to Amersfoort (Amf). Figure 5.5d shows a recoverable feasible completion covering the critical task 736 from Groningen to Zwolle. Its recovery alternative that is valid in the pessimistic scenario $\overline{s}$ is shown in Figure 5.5e. In this recovery alternative, task 9142 is in fact a deadheading task, since, according to the definition of a recoverable duty, this task is also covered by another feasible completion.

Chapter 6

Summary and conclusion

This thesis studies disruption management in railway systems and aims to reduce the inconvenience passengers face during disruptions. In case of a disruption, the railway operator has to come up with a new timetable and new rolling stock and crew schedules. To help the dispatchers in finding these new schedules, this thesis discusses several railway disruption management approaches. The approaches extend previous research in railway disruption management by reducing the number of assumptions, by adding more flexibility and by taking passenger behavior or uncertainty during disruptions into account. These extensions prove to be powerful in reducing the inconvenience passengers face by having less cancelled trains and less delays for passengers.

In Chapter 2 the timetable rescheduling problem has been investigated. From this research it can be concluded that, in case of a disruption, it is possible to find new timetables within computation times which are reasonable for practice. The model used is flexible and can handle different kinds of infrastructure layouts. In current practice, dispatchers use prescribed contingency plans if a disruption occurs. However, by the flexibility and short computation times, our model can be of great use for practice. Another advantage of the model is that it also considers the available rolling stock. This decreases the probability that additional trains need to be cancelled due to lack of rolling stock and thereby decreases the probability that additional rescheduling steps must be carried out.

Rolling stock rescheduling is discussed in Chapter 3. The chapter does not introduce a new model for rolling stock rescheduling. However, it comes up with a framework in which the timetable, the rolling stock schedule and the passenger behavior are considered together. Of the three schedules (timetable, rolling stock and crew) of a railway operator, the timetable and rolling stock schedules are the ones which influence the passenger flows. Chapter 3 is based on previous research in which only the rolling stock schedule was taken into account to improve passenger service during disruptions. However, computational tests of our framework on instances of Netherlands Railways showed that slight timetable adaptations, such as having an additional stop of a train at a certain station, can reduce the delays of passengers significantly. During disruptions, dispatchers should use our framework to decide which trains on the alternative routes should get an increased capacity and which should make an additional stop, such that passengers face as little delay as possible by the disruption.

If the operator has a new timetable and rolling stock schedule to handle the disruption, the last step is to find crew to operate all trains. In this thesis two chapters investigate how previous crew rescheduling approaches in railways can be improved. The first approach, discussed in Chapter 4, adds more flexibility while constructing the new crew schedule. The approach allows some trains to be slightly delayed such that new connections for crew become available. The results of tests with the approach on instances of Netherlands Railways demonstrate that allowing these small timetable modifications reduces the number of trains which need to be cancelled by lack of crew. This shows that dispatchers can be better off by adding the option to slightly modify the timetable.

The second crew rescheduling approach, discussed in Chapter 5, aims at creating more robust crew schedules. While rescheduling the crew duties, it takes into account that the disruption can take longer than estimated. At the moment it becomes clear that the disruption takes longer than estimated, the crew duties must be rescheduled again. The advantage of our quasi-robust rescheduling approach is that, if the disruption takes longer than planned, then there are less rescheduling costs in the second rescheduling phase. However, in the initial rescheduling the costs are a bit higher. Our approach provides an overview of solutions in which it is stated how many tasks need to be cancelled by lack of crew in each phase. In this way dispatchers can decide which kind of solution they prefer.

The rescheduling approaches discussed in this thesis demonstrate to perform well on practical instances. They demonstrate that they are able to find schedules in which passengers face less inconvenience than in schedules found by current approaches. First, the timetable rescheduling approach shows to be flexible in finding schedules which minimize cancellations and delays. Secondly, the rolling stock rescheduling approach shows that a combination of rescheduling the rolling stock and timetable leads to a drastic decrease of passenger delays in comparison with an approach which only considers the rolling stock schedule. At last, the two crew rescheduling approaches show that they can prevent cancellations by lack of crew if slight timetable adaptations are allowed or in case the uncertainty of the disruption is considered.

In addition to the solution quality, the approaches also perform well in terms of computation time. Therefore they can be of great advisory support for the dispatchers. The approaches were tested on instances of Netherlands Railways. However, the approaches can also be used in applications of other railway operators. In case of the crew rescheduling approaches they can even be applied in disruption management systems of other transport modes (e.g. airlines and bus/tram/metro systems) to improve the passenger service.

The next step should be to link the proposed approaches to the data systems of the railway operators such that the approaches can make use of the real-time data streams.

References

E. Abbink, M. Fischetti, L. Kroon, G. Timmer, and M. Vromans. Reinventing Crew Scheduling at Netherlands Railways. *Interfaces*, 35:393–401, 2005.

A. Abdelghany, G. Ekollu, R. Narasimhan, and K. Abdelghany. A Proactive Crew Recovery Decision Support Tool for Commercial Airlines during Irregular Operations. *Annals of Operations Research*, 127:309–331, 2004.

K. F. Abdelghany, A. F. Abdelghany, and G. Ekollu. An integrated decision support tool for airline schedule recovery during irregular operations. *European Journal of Operational Research*, 185:825–848, 2008.

R. Acuna-Agost, P. Michelon, D. Feillet, and S. Gueye. SAPI: Statistical Analysis of Propagation of Incidents. A new approach for rescheduling trains after disruptions. *European Journal of Operational Research*, 215:227 – 243, 2011a.

R. Acuna-Agost, P. Michelon, D. Feillet, and S. Gueye. A MIP-based local search method for the railway rescheduling problem. *Networks*, 57:69–86, 2011b.

B. Adenso-Díaz, M. Oliva González, and P. González-Torre. On-line timetable rescheduling in regional train services. *Transportation Research Part B: Methodological*, 33:387–398, 1999.

A. Albrecht, D. Panton, and D. Lee. Rescheduling rail networks with maintenance disruptions using Problem Space Search. *Computers & Operations Research*, 40:703 – 712, 2013.

J. E. Beasley. Lagrangian relaxation. In C. R. Reeves, editor, *Modern heuristic techniques for combinatorial problems*, pages 243–303. John Wiley & Sons, Inc., New York, 1993.

A. Ben-Tal and A. Nemirovski. Robust optimization - methodology and applications. *Mathematical Programming*, 92:453–480, 2002.

D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98:49–71, 2003.

J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, 1997.

M. Boccia, C. Mannino, and I. Vasilyev. The dispatching problem on multitrack territories: Heuristic approaches based on mixed integer linear programming. *Networks*, 62:315–326, 2013.

P. Brucker, S. Heitmann, and S. Knust. Scheduling railway traffic at a construction site. *OR Spectrum*, 24:19–30, 2002.

V. Cacchiani and P. Toth. Nominal and Robust Train Timetabling Problems. *European Journal of Operational Research*, 219:727–737, 2012.

V. Cacchiani, A. Caprara, L. Galli, L. Kroon, G. Maróti, and P. Toth. Railway Rolling Stock Planning: Robustness Against Large Disruptions. *Transportation Science*, 46:217–232, 2012.

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, Á. Marín, and G. Maróti. Recovery of disruptions in rapid transit networks. *Transportation Research Part E: Logistics and Transportation Review*, 53:15–33, 2013.

G. Caimi, M. Fuchsberger, M. Laumanns, and M. Lüthi. A model predictive control approach for discrete-time rescheduling in complex central railway station areas. *Computers & Operations Research*, 39:2578 – 2593, 2012.

A. Caprara, M. Fischetti, and P. Toth. A Heuristic Method for the Set Covering Problem. *Operations Research*, 47:730–743, 1999.

A. Caprara, L. Kroon, M. Monaci, M. Peeters, and P. Toth. Passenger Railway Optimization . In C. Barnhart and G. Laporte, editors, *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, pages 129 – 187. Elsevier, 2007.

A. Caprara, L. Kroon, and P. Toth. Optimization Problems in Passenger Railway Systems. In J. Cochran, L. Cox, P. Keskinocak, J. Kharoufed, and J. Cole Smith, editors, *Wiley Encyclopedia of Operations Research and Management Science*, volume 6, pages 3896–3905. John Wiley & Sons, Inc., 2011.

J. Clausen, A. Larsen, J. Larsen, and N. J. Rezanova. Disruption management in the airline industry - Concepts, models and methods. *Computers & Operations Research*, 37:809–821, 2010.

F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Evaluation of a green wave policy in real-time railway traffic management. *Transportation Research Part C: Emerging Technologies*, 17:607–616, 2009.

F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B: Methodological*, 44:175–192, 2010.

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.

F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Bi-objective conflict detection and resolution in railway traffic management. *Transportation Research Part C: Emerging Technologies*, 20:79–94, 2012.

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*, 183:643–657, 2007.

A. D'Ariano, F. Corman, D. Pacciarelli, and M. Pranzo. Reordering and local rerouting strategies to manage train traffic in real time. *Transportation Science*, 42:405–419, 2008a.

A. D'Ariano, D. Pacciarelli, and M. Pranzo. Assessment of flexible timetables in real-time traffic management of a railway bottleneck. *Transportation Research Part C: Emerging Technologies*, 16:232–245, 2008b.

G. Desaulniers, J. Desrosiers, and M. M. Solomon. Accelerating Strategies in Column Generation Methods for Vehicel Routing and Crew Scheduling Problems. In C. C. Ribeiro and P. Hansen, editors, *Essays and Surveys in Metaheuristics*, pages 309–324. Kluwer, Boston, 2001.

J. Desrosiers and M. E. Lübbecke. A Primer in Column Generation. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, pages 1–32. Springer, New York, 2005.

T. Dollevoet, D. Huisman, M. Schmidt, and A. Schöbel. Delay Management with Rerouting of Pasengers. *Transportation Science*, 46:74–89, 2012.

J. Dumas and F. Soumis. Passenger Flow Model for Airline Networks. *Transportation Science*, 42:197–207, 2008.

P. 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: 1281–1297, 2006.

M. L. Fisher. The Lagrangian Relaxation Method for Solving Integer Programming Problems. *Management Science*, 27:1–18, 1981.

D. Huisman, R. Jans, M. Peters, and A. Wagelmans. Combining Column Generation and Lagrangian Relaxation. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, pages 247–270. Springer, New York, 2005.

S. Irnich and G. Desaulniers. Shortest Path Problems with Resource Constraints. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, pages 33–65. Springer, New York, 2005.

J. Jespersen-Groth, D. Potthoff, J. Clausen, D. Huisman, L. G. Kroon, G. Maróti, and M. Nyhave Nielsen. Disruption Management in Passenger Railway Transportation. In R. K. Ahuja, R. H. Möhring, and C. D. Zaroliagis, editors, *Robust and Online Large-Scale Optimization*, pages 399–421. Springer, Berlin, 2009.

V. Johnson, L. Lettovský, G. L. Nemhauser, R. Pandit, and S. Querido. Final report to Northwest Airlines on the crew recovery problem. Technical report, The Logistic Institute, Georgia Institute of Technology, 1994.

P. Kall and S. Wallace. *Stochastic Programming*. John Wiley & Sons, Chichester, 1994.

S. Kanai, K. Shiina, S. Harada, and N. Tomii. An optimal delay management algorithm from passengers' viewpoints considering the whole railway network. *Journal of Rail Transport Planning & Management*, 1:25–37, 2011.

P. Kecman, F. Corman, A. D'Ariano, and R. Goverde. Rescheduling models for railway traffic management in large-scale networks. *Public Transport*, 5:95–123, 2013.

D. Klabjan, E. L. Johnson, G. L. Nemhauser, E. Gelman, and S. Ramaswamy. Airline Crew Scheduling with Time Windows and Plane-Count Constraints. *Transportation Science*, 36: 337–348, 2002.

L. Kroon, D. Huisman, E. Abbink, P.-J. Fioole, M. Fischetti, G. Maróti, L. Schrijver, A. Steenbeek, and R. Ybema. The New Dutch Timetable: The OR Revolution. *Interfaces*, 39:6–17, 2009.

L. Kroon, G. Maróti, and L. Nielsen. Rescheduling of Railway Rolling Stock with Dynamic Passenger Flows. *To appear in: Transportation Science*, 2014. doi: 10.1287/trsc.2013.0502.

C. Liebchen, M. Lübbecke, R. Möhring, and S. Stiller. The Concept of Recoverable Robustness, Linear Programming, and Railway Applications. In *Robust and Online Large-Scale Optimization*, pages 1–27. Springer, New York, 2009.

I. Louwerse and D. Huisman. Adjusting a railway timetable in case of partial or complete blockades. *European Journal of Operational Research*, 235:583 – 593, 2014.

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

R. M. Lusby, J. Larsen, M. Ehrgott, and D. M. Ryan. A set packing inspired method for real-time junction train routing. *Computers & Operations Research*, 40:713 – 724, 2013.

C. Mannino and A. Mascis. Optimal Real-Time Traffic Control in Metro Stations. *Operations Research*, 57:1026–1039, 2009.

A. Mercier and F. Soumis. An integrated aircraft routing, crew scheduling and flight retiming model. *Computers & Operations Research*, 34:2251–2265, 2007.

L. Nielsen, L. Kroon, and G. Maróti. A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, 220:496 – 509, 2012.

D. Potthoff. *Railway Crew Rescheduling: Novel Approaches and Extensions*. PhD thesis, Erasmus University, Rotterdam, The Netherlands, 2010.

D. Potthoff, D. Huisman, and G. Desaulniers. Column Generation with Dynamic Duty Selection for Railway Crew Rescheduling. *Transportation Science*, 44:493–505, 2010.

N. J. Rezanova. *The Train Driver Recovery Problem - Solution Method and Decision Support System Framework*. PhD thesis, Technical University of Denmark, 2009.

N. J. Rezanova and D. M. Ryan. The train driver recovery problem - A set partitioning based model and solution approach. *Computers & Operations Research*, 37:845–856, 2010.

M. Schachtebeck and A. Schöbel. To Wait or Not to Wait - And Who Goes First? Delay Management with Priority Decisions. *Transportation Science*, 44:307–321, 2010.

A. Schöbel. Capacity constraints in delay management. *Public Transport*, 1:135–154, 2009.

M. Stojković and F. Soumis. An Optimization Model for the Simultaneous Operational Flight and Pilot Scheduling Problem. *Management Science*, 47:1290–1350, 2001.

M. Stojković and F. Soumis. The operational flight and multi-crew scheduling problem. *Yugoslav Journal of Operations Research*, 15:25–48, 2005.

L. P. **Veelenturf**, D. Potthoff, D. Huisman, and L. G. Kroon. Railway Crew Rescheduling with Retiming. *Transportation Research Part C: Emerging Technologies*, 20:95 – 110, 2012.

L. P. **Veelenturf**, M. P. Kidd, V. Cacchiani, L. G. Kroon, and P. Toth. A railway timetable rescheduling approach for handling large scale disruptions. Technical Report ERS-2014-010-LIS, Erasmus Research Institute in Management (ERIM), 2014a. URL `http://hdl.handle.net/1765/51678`. Under review at: *Transportation Science*.

L. P. **Veelenturf**, L. G. Kroon, and G. Maróti. Passenger Oriented Railway Disruption Management by adapting Timetables and Rolling Stock Schedules. In B. M. E. Ozcan, E. K. Burke, editor, *Proceedings of the 10th International Conference of the Practice and Theory of Automated Timetabling (PATAT)*, York, United Kingdom, Aug 2014b. ISBN 978-0-9929984-0-0. Under review at: *Annals of Operations Research*.

L. P. **Veelenturf**, D. Potthoff, D. Huisman, L. G. Kroon, G. Maróti, and A. P. M. Wagelmans. A Quasi-Robust Optimization Approach for Crew Rescheduling. *Transportation Science*, forthcoming. doi: 10.1287/trsc.2014.0545.

J. Törnquist and J. A. Persson. N-tracked railway traffic re-scheduling during disturbances. *Transportation Research Part B: Methodological*, 41:342 – 362, 2007.

C. G. Walker, J. N. Snowdon, and D. M. Ryan. Simultaneous disruption recovery of a train timetable and crew roster in real time. *Computers & Operations Research*, 32:2077–2094, 2005.

M. Wiklund. Serious breakdowns in the track infrastructure - Calculation of the effects on rail traffic. In *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis - RailHannover 2007*, Hannover, Germany, Mar 2007.

Nederlandse Samenvatting (Summary in Dutch)

Helaas vinden er in spoorwegnetwerken dagelijks verstoringen plaats. Reizigers komen hierdoor later dan gepland aan op hun bestemming, wat voor hen erg vervelende gevolgen kan hebben. Spoorvervoerders moeten dan ook alles op alles zetten om te zorgen dat de reizigers zo min mogelijk last ondervinden van verstoringen. Aan de ene kant zal een spoorvervoerder (gezamenlijk met een spoorwegbeheerder) proberen om verstoringen te voorkomen. Het is echter niet altijd mogelijk om elke verstoring te voorkomen. Daardoor zal er aan de andere kant ook onderzoek gedaan moeten worden hoe er gezorgd kan worden dat reizigers zo min mogelijk hinder van verstoringen ondervinden.

In dit proefschrift wordt op dit laatste gefocust. In de verschillende hoofdstukken van dit proefschrift wordt onderzocht hoe in het geval van verstoringen de treinen het best bijgestuurd kunnen worden. In geval van een verstoring zal de spoorvervoerder een nieuwe dienstregeling en nieuwe materieel- en personeelsplannen moeten maken. Het is van groot belang dat bij het maken van de nieuwe plannen rekening gehouden wordt met de reizigers en met wat de reizigers willen en doen.

Hieronder volgen voor elk van de drie genoemde planningsproblemen voorbeelden die een indruk geven tegen wat voor problemen een spoorvervoerder aanloopt in geval van verstoringen. Een belangrijk onderdeel hierbij is dat er maar weinig tijd beschikbaar is om met nieuwe plannen te komen omdat de treinen ondertussen gewoon doorrijden. Verder zal worden aangegeven hoe deze problemen in dit proefschrift zijn onderzocht.

Bijsturing van de dienstregeling

Als er door een verstoring geen treinverkeer mogelijk is tussen Woerden en Utrecht zullen de treinen tussen Woerden en Utrecht uit moeten vallen of ze moeten worden omgeleid (bijvoorbeeld via Breukelen).

Als er gekozen wordt om de treinen uit te laten vallen is de vraag of de treinen alleen uitvallen op het traject tussen Woerden en Utrecht of misschien ook over een langer traject. Van invloed op deze beslissingen is de hoeveelheid beschikbare sporen op de stations. Als een trein die normaal van Den Haag via Woerden naar Utrecht gaat nu eindigt in Woerden, neemt het daar een spoor in beslag. Als er nu allemaal treinen in Woerden eindigen in plaats van in Utrecht zijn op een gegeven moment alle sporen bezet en dan kan er geen nieuwe trein meer binnenkomen. Om dit te voorkomen, kan de vervoerder besluiten dat het materieel van de treinen die in Woerden eindigen ingezet wordt voor treinen die normaal van Utrecht via Woerden naar Den Haag gaan, maar die nu in Woerden starten in plaats van in Utrecht. Als er een lange tijd zit tussen de aankomst in Woerden en het vertrek uit Woerden kan het nog steeds zijn dat er problemen ontstaan doordat sporen te lang bezet zijn. Het kan dus betekenen dat het niet anders mogelijk is dan dat de treinen al op een plek voor Woerden eindigen en teruggaan naar Den Haag.

Een andere optie is om de treinen om te laten rijden via Breukelen. Als daarvoor gekozen wordt, moet er rekening gehouden worden met het traject tussen Woerden en Breukelen en met het traject tussen Breukelen en Utrecht. Is daar wel ruimte op het spoor voor deze treinen? Uit veiligheidsoverwegingen moet bijvoorbeeld tussen elke twee treinen die achter elkaar op het zelfde spoor rijden een minimaal aantal minuten als buffer zitten. Verder hebben treinen ook verschillende snelheden en kan het gebeuren dat een snelle trein opgehouden wordt door een langzame trein.

In hoofdstuk 2 van dit proefschrift is onderzoek gedaan naar een wiskundig model dat in geval van een verstoring een nieuwe dienstregeling kan genereren welke rekening houdt met de capaciteiten van de stations en tussen de stations. Verder houdt het model er rekening mee dat er altijd materieel aanwezig moet zijn voor een trein. De aanpak is in het algemeen in staat om binnen een anderhalve minuut een nieuwe dienstregeling te genereren waarin het aantal treinen dat uitvalt en vertraagd wordt is geminimaliseerd.

Bijsturing van het materieel

In dit voorbeeld gaan we weer uit van een verstoring tussen Woerden en Utrecht waarbij nu de helft van het aantal sporen nog steeds beschikbaar is. Verder gaan we er vanuit dat het bijsturingsmodel van hoofdstuk 2 heeft ontdekt dat, als twee van de vier stoptreinen per uur uitvallen, de rest van de treinen nog steeds kunnen rijden.

Tussen Woerden en Utrecht ligt het station Vleuten waar alleen stoptreinen stoppen. Nu stoppen op Vleuten nog maar twee treinen per uur in plaats van vier. Dit zal betekenen dat deze resterende treinen nu ongeveer twee keer zoveel reizigers hebben als normaal. Om hier mee om te gaan heeft een spoorvervoerder twee opties.

De eerste optie is om de stoptreinen die nog steeds rijden langer te maken door het inzetten van langere materieelsamenstellingen. Dan passen er meer mensen in de trein. De trein kan echter niet onbeperkt langer gemaakt worden. Zo kan de trein bijvoorbeeld niet langer zijn dan het perron in Vleuten. Verder moet er ook gekeken worden of er überhaupt wel reserve materieel aanwezig is en of er wel voldoende tijd is om het reserve materieel van het rangeerterrein af te halen en te koppelen aan de huidige trein.

Een tweede optie is om treinen die normaal niet in Vleuten stoppen daar nu wel te laten stoppen. Hierdoor hoeven de reizigers in Vleuten niet te wachten tot de volgende stoptrein en komen ze eerder aan op hun bestemming. Maar uiteraard geldt ook hier dat er wel genoeg plaats moet zijn voor de reizigers in de trein die de extra stop maakt. Verder moet er rekening mee gehouden worden dat de extra stop voor een vertraging van de trein zorgt en dat reizigers in de trein daardoor bijvoorbeeld hun overstap kunnen missen.

In hoofdstuk 3 van het proefschrift wordt onderzocht hoe in het geval van een verstoring bepaald kan worden welke treinen een andere materieelsamenstelling dienen te krijgen en welke treinen een extra stop moeten maken. Hierbij wordt rekening gehouden met hoe de reizigers zullen reageren op de nieuwe dienstregeling en het nieuwe materieelplan. De besluiten over de dienstregeling en het materieelplan worden zodanig gemaakt dat de totale vertraging die reizigers ondervinden op hun route wordt geminimaliseerd.

Bijsturing van het personeel

Als er geen machinist gevonden kan worden voor een trein kan de trein niet rijden. Het personeelsplan heeft dus een invloed op de reismogelijkheden van de reizigers. Bij het maken van een nieuw personeelsplan tijdens een verstoring moet het aantal treinen waarvoor geen personeel gevonden kan worden dus geminimaliseerd worden. In dit proefschrift zijn hier twee onderzoeken naar gedaan.

Stel dat er een machinist is die eerst een trein moet rijden van Woerden naar Rotterdam, dan een trein van Rotterdam naar Woerden en tot slot een trein van Woerden naar Eindhoven. Nadat de machinist de trein van Woerden naar Rotterdam heeft gereden, wordt bekend dat er geen treinverkeer meer mogelijk is tussen Rotterdam en Woerden. Er zijn dan verschillende problemen die de spoorvervoerder moet oplossen.

Doordat de treinen van Rotterdam naar Woerden uitvallen moet de vervoerder een manier vinden om de machinist toch op tijd in Woerden te krijgen om vanaf daar de trein van Woerden naar Eindhoven te rijden. Als dat niet lukt zal de vervoerder een manier moeten vinden om de machinist in ieder geval op zijn laatste bestemming (Eindhoven) te krijgen. De machinist mag niet zomaar op een willekeurig station zijn dienst eindigen. Voor de reizigers is het echter belangrijker dat, als de machinist niet op tijd in Woerden kan zijn om de trein tussen Woerden en Eindhoven te rijden, een andere machinist dit van hem overneemt. Anders moet die trein uitvallen.

In dit voorbeeld behandelen we slechts één trein die uitvalt tussen Rotterdam en Woerden. Maar men kan bedenken hoeveel puzzels de vervoerder moet oplossen als er meerdere treinen uitvallen waardoor meerdere diensten van machinisten (en materieel) geraakt worden.

In het verleden zijn er al modellen ontwikkeld voor het bijsturen van rijdend personeel. In hoofdstuk 4 is onderzocht wat het effect zal zijn als we deze modellen uitbreiden met de mogelijkheid om kleine dienstregelingswijzigingen toe te passen. Zo kan het voorkomen dat in de huidige modellen geen machinist gevonden kan worden voor de trein van Woerden naar Eindhoven, maar dat een kleine extra vertraging van deze trein er ineens voor kan zorgen dat het wel lukt om een machinist te vinden om de trein te rijden. Door deze kleine dienstregelingswijzigingen toe te staan blijkt het mogelijk om minder treinen uit te laten vallen door het ontbreken van een machinist.

Een andere uitbreiding op de bestaande bijsturingsmodellen voor personeel wordt besproken in hoofdstuk 5. Voor de bestaande modellen is het belangrijk om de exacte duur van de verstoring te weten. Op het moment dat de verstoring begint is het echter moeilijk in te schatten hoe lang de verstoring exact zal duren. Daarom nemen we de onzekerheid in de duur van de verstoring mee in het model dat besproken wordt in hoofdstuk 5. Dit maakt de nieuwe personeelsplannen meer robuust en leidt er toe dat, als de verstoring langer duurt dan verwacht, er dan minder treinen uitvallen door het ontbreken van machinisten.

Eindconclusie

In dit proefschrift zijn verschillende uitbreidingen van bestaande bijsturingsmodellen onderzocht. De modellen zijn getest op scenario's van de Nederlandse Spoorwegen (NS). Het blijkt dat de uitbreidingen zorgen voor minder uitval van treinen en minder vertragingen voor reizigers. De ontwikkelde methoden zijn in staat om in zeer korte tijd een nieuwe dienstregeling, materieel- en personeelsplanning te genereren. Dit maakt het mogelijk om deze modellen in de praktijk te gebruiken. Er zijn dan nog wel uitdagingen om de modellen gekoppeld te krijgen aan de real-time informatiesytemen.

Curriculum Vitae

Lucas Veelenturf was born July 22nd 1987 in Woerden, the Netherlands. In 2005 he graduated from high school education at *Minkema College Woerden*. For his Bachelor and Master of Science education he studied Econometrics and Management Science at *Erasmus School of Economics, Erasmus University Rotterdam*. During his study he applied research at *Netherlands Railways* for both his Bachelor and Master of Science thesis. After graduating for his master's degree in 2009, he started his PhD research in railway disruption management at *Rotterdam School of Management, Erasmus University*. He also spent three months at *University of Bologna* in the group of Professor Paolo Toth for a close research collaboration.

The research during his PhD study lead to several awards and publications. At the moment of writing he has two published papers in *Transportation Research Part B: Methodological* and *Transportation Research Part C: Emerging Technologies*, and one accepted paper in *Transportation Science*. Furthermore, two papers are in the reviewing process. At the INFORMS Annual Meetings of 2012 and 2013 he became second and third in the Student Paper Award competition of the Railway Application Section of INFORMS. He also got granted the Junior of the Year Award of Research School TRAIL in 2010.

Next to research he has a strong affiliation with teaching. He got his first teaching experience as a teaching assistant for Mathematics and Linear Algebra during his study at the Erasmus School of Economics. As a PhD student at Rotterdam School of Management, he had a lot of teaching activities (especially for the Mathematics course) which he enjoyed a lot.

Since September 2014, Lucas is as an Assistant Professor at the *School of Industrial Engineering* of *Eindhoven University of Technology*.

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT (ERIM)

ERIM PH.D. SERIES RESEARCH IN MANAGEMENT

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://hdl.handle.net/1765/1
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 LAST FIVE YEARS

Abbink, E., *Crew Management in Passenger Rail Transport*, Promotor(s): Prof.dr. L.G. Kroon & Prof.dr. A.P.M. Wagelmans, EPS-2014-325-LIS, http://hdl.handle.net/1765/1

Acar, O.A., *Crowdsourcing for innovation: Unpacking Motivational, Knowledge and Relational Mechanisms of Innovative Behavior in Crowdsourcing Platforms,* Promotor: Prof.dr. J.C.M. van den Ende, EPS-2014-321-LIS, http://hdl.handle.net/1765/1

Acciaro, M., *Bundling Strategies in Global Supply Chains*, Promotor(s): Prof.dr. H.E. Haralambides, EPS-2010-197-LIS, http://hdl.handle.net/1765/19742

Akpinar, E., *Consumer Information Sharing; Understanding Psychological Drivers of Social Transmission*, Promotor(s): Prof.dr.ir. A. Smidts, EPS-2013-297-MKT, http://hdl.handle.net/1765/50140

Alexiev, A., *Exploratory Innovation: The Role of Organizational and Top Management Team Social Capital*, Promotor(s): Prof.dr. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-208-STR, http://hdl.handle.net/1765/20632

Akin Ates, M., *Purchasing and Supply Management at the Purchase Category Level: Strategy, Structure, and Performance*, Promotor: Prof.dr. J.Y.F. Wynstra, EPS-2014-300-LIS, http://hdl.handle.net/1765/1

Almeida, R.J.de, *Conditional Density Models Integrating Fuzzy and Probabilistic Representations of Uncertainty,* Promotor Prof.dr.ir. Uzay Kaymak, EPS-2014-310-LIS, http://hdl.net/1765/1

Bannouh, K., *Measuring and Forecasting Financial Market Volatility using High-Frequency Data*, Promotor: Prof.dr.D.J.C. van Dijk, EPS-2013-273-F&A, http://hdl.handle.net/1765/38240

Benning, T.M., *A Consumer Perspective on Flexibility in Health Care: Priority Access Pricing and Customized Care*, Promotor: Prof.dr.ir. B.G.C. Dellaert, EPS-2011-241-MKT, http://hdl.handle.net/1765/23670

Ben-Menahem, S.M., *Strategic Timing and Proactiveness of Organizations*, Promotor(s): Prof.dr. H.W. Volberda & Prof.dr.ing. F.A.J. van den Bosch, EPS-2013-278-S&E, http://hdl.handle.net/1765/ 39128

Berg, W.E. van den, *Understanding Salesforce Behavior Using Genetic Association Studies,* Promotor: Prof.dr. W.J.M.I. Verbeke, EPS-2014-311-MKT, http://hdl.handle.net/1765/ 51440

Betancourt, N.E., *Typical Atypicality: Formal and Informal Institutional Conformity, Deviance, and Dynamics*, Promotor: Prof.dr. B. Krug, EPS-2012-262-ORG, http://hdl.handle.net/1765/32345

Binken, J.L.G., *System Markets: Indirect Network Effects in Action, or Inaction*, Promotor: Prof.dr. S. Stremersch, EPS-2010-213-MKT, http://hdl.handle.net/1765/21186

Blitz, D.C., *Benchmarking Benchmarks*, Promotor(s): Prof.dr. A.G.Z. Kemna & Prof.dr. W.F.C. Verschoor, EPS-2011-225-F&A, http://hdl.handle.net/1765/226244

Boons, M., *Working Together Alone in the Online Crowd: The Effects of Social Motivations and Individual Knowledge Backgrounds on the Participation and Performance of Members of Online Crowdsourcing Platforms*, Promotor: Prof.dr. H.G. Barkema, EPS-2014-306-S&E, http://hdl.net/1765/50711

Borst, W.A.M., *Understanding Crowdsourcing: Effects of Motivation and Rewards on Participation and Performance in Voluntary Online Activities*, Promotor(s): Prof.dr.ir. J.C.M. van den Ende & Prof.dr.ir. H.W.G.M. van Heck, EPS-2010-221-LIS, http://hdl.handle.net/1765/ 21914

Budiono, D.P., *The Analysis of Mutual Fund Performance: Evidence from U.S. Equity Mutual Funds*, Promotor: Prof.dr. M.J.C.M. Verbeek, EPS-2010-185-F&A, http://hdl.handle.net/1765/18126

Burger, M.J., *Structure and Cooptition in Urban Networks*, Promotor(s): Prof.dr. G.A. van der Knaap & Prof.dr. H.R. Commandeur, EPS-2011-243-ORG, http://hdl.handle.net/1765/26178

Byington, E., *Exploring Coworker Relationships: Antecedents and Dimensions of Interpersonal Fit, Coworker Satisfaction, and Relational Models*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2013-292-ORG, http://hdl.handle.net/1765/41508

Camacho, N.M., *Health and Marketing; Essays on Physician and Patient Decision-making*, Promotor: Prof.dr. S. Stremersch, EPS-2011-237-MKT, http://hdl.handle.net/1765/23604

Cankurtaran, P. *Essays On Accelerated Product Development,* Promotor: Prof.dr.ir. G.H. van Bruggen, EPS-2014-317-MKT, http://hdl.handle.net/1765/1

Caron, E.A.M., *Explanation of Exceptional Values in Multi-dimensional Business Databases*, Promotor(s): Prof.dr.ir. H.A.M. Daniels & Prof.dr. G.W.J. Hendrikse, EPS-2013-296-LIS, http://hdl.handle.net/1765/50005

Carvalho, L., *Knowledge Locations in Cities; Emergence and Development Dynamics*, Promotor: Prof.dr. L. van den Berg, EPS-2013-274-S&E, http://hdl.handle.net/1765/ 38449

Carvalho de Mesquita Ferreira, L., *Attention Mosaics: Studies of Organizational Attention*, Promotor(s): Prof.dr. P.M.A.R. Heugens & Prof.dr. J. van Oosterhout, EPS-2010-205-ORG, http://hdl.handle.net/1765/19882

Cox, R.H.G.M., *To Own, To Finance, and to Insure; Residential Real Estate Revealed*, Promotor: Prof.dr. D. Brounen, EPS-2013-290-F&A, http://hdl.handle.net/1765/40964

Defilippi Angeldonis, E.F., *Access Regulation for Naturally Monopolistic Port Terminals: Lessons from Regulated Network Industries*, Promotor: Prof.dr. H.E. Haralambides, EPS-2010-204-LIS, http://hdl.handle.net/1765/19881

Deichmann, D., *Idea Management: Perspectives from Leadership, Learning, and Network Theory*, Promotor: Prof.dr.ir. J.C.M. van den Ende, EPS-2012-255-ORG, http://hdl.handle.net/1765/ 31174

Desmet, P.T.M., *In Money we Trust? Trust Repair and the Psychology of Financial Compensations*, Promotor: Prof.dr. D. De Cremer & Prof.dr. E. van Dijk, EPS-2011-232-ORG, http://hdl.handle.net/1765/23268

Dietvorst, R.C., *Neural Mechanisms Underlying Social Intelligence and Their Relationship with the Performance of Sales Managers*, Promotor: Prof.dr. W.J.M.I. Verbeke, EPS-2010-215-MKT, http://hdl.handle.net/1765/21188

Dollevoet, T.A.B., *Delay Management and Dispatching in Railways*, Promotor: Prof.dr. A.P.M. Wagelmans, EPS-2013-272-LIS, http://hdl.handle.net/1765/38241

Doorn, S. van, *Managing Entrepreneurial Orientation*, Promotor(s): Prof.dr. J.J.P. Jansen, Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2012-258-STR, http://hdl.handle.net/1765/32166

Douwens-Zonneveld, M.G., *Animal Spirits and Extreme Confidence: No Guts, No Glory*, Promotor: Prof.dr. W.F.C. Verschoor, EPS-2012-257-F&A, http://hdl.handle.net/1765/31914

Duca, E., *The Impact of Investor Demand on Security Offerings*, Promotor: Prof.dr. A. de Jong, EPS-2011-240-F&A, http://hdl.handle.net/1765/26041

Duursema, H., *Strategic Leadership; Moving Beyond the Leader-follower Dyad*, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2013-279-ORG, http://hdl.handle.net/1765/ 39129

Eck, N.J. van, *Methodological Advances in Bibliometric Mapping of Science*, Promotor: Prof.dr.ir. R. Dekker, EPS-2011-247-LIS, http://hdl.handle.net/1765/26509

Essen, M. van, *An Institution-Based View of Ownership*, Promotor(s): Prof.dr. J. van Oosterhout & Prof.dr. G.M.H. Mertens, EPS-2011-226-ORG, http://hdl.handle.net/1765/22643

Feng, L., *Motivation, Coordination and Cognition in Cooperatives*, Promotor: Prof.dr. G.W.J. Hendrikse, EPS-2010-220-ORG, http://hdl.handle.net/1765/21680

Fourné, S. P. L. Managing Organizational Tensions: A Multi-level Perspective on Exploration, Exploitation, and Ambidexterity, Promotor(s): Prof.dr. J.J.P. Jansen, Prof.dr. S.J. Magala & dr. T.J.M.Mom, EPS-2014-318-S&E, http://hdl.handle.net/1765/21680

Gharehgozli, A.H., *Developing New Methods for Efficient Container Stacking Operations*, Promotor: Prof.dr.ir. M.B.M. de Koster, EPS-2012-269-LIS, http://hdl.handle.net/1765/ 37779

Gils, S. van, *Morality in Interactions: On the Display of Moral Behavior by Leaders and Employees*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2012-270-ORG, http://hdl.handle.net/1765/ 38028

Ginkel-Bieshaar, M.N.G. van, *The Impact of Abstract versus Concrete Product Communications on Consumer Decision-making Processes*, Promotor: Prof.dr.ir. B.G.C. Dellaert, EPS-2012-256-MKT, http://hdl.handle.net/1765/31913

Gkougkousi, X., *Empirical Studies in Financial Accounting*, Promotor(s): Prof.dr. G.M.H. Mertens & Prof.dr. E. Peek, EPS-2012-264-F&A, http://hdl.handle.net/1765/37170

Glorie, K.M., *Clearing Barter Exchange Markets: Kidney Exchange and Beyond*, *Promotor(s):* Prof.dr. A.P.M. Wagelmans & Prof.dr. J.J. van de Klundert, EPS-2014-329-LIS, http://hdl.handle.net/1765/1

Hakimi, N.A, *Leader Empowering Behaviour: The Leader's Perspective: Understanding the Motivation behind Leader Empowering Behaviour*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2010-184-ORG, http://hdl.handle.net/1765/17701

Hensmans, M., *A Republican Settlement Theory of the Firm: Applied to Retail Banks in England and the Netherlands (1830-2007)*, Promotor(s): Prof.dr. A. Jolink & Prof.dr. S.J. Magala, EPS-2010-193-ORG, http://hdl.handle.net/1765/19494

Hernandez Mireles, C., *Marketing Modeling for New Products*, Promotor: Prof.dr. P.H. Franses, EPS-2010-202-MKT, http://hdl.handle.net/1765/19878

Heyde Fernandes, D. von der, *The Functions and Dysfunctions of Reminders*, Promotor: Prof.dr. S.M.J. van Osselaer, EPS-2013-295-MKT, http://hdl.handle.net/1765/41514

Heyden, M.L.M., *Essays on Upper Echelons & Strategic Renewal: A Multilevel Contingency Approach*, Promotor(s): Prof.dr. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2012-259-STR, http://hdl.handle.net/1765/32167

Hoever, I.J., *Diversity and Creativity: In Search of Synergy*, Promotor(s): Prof.dr. D.L. van Knippenberg, EPS-2012-267-ORG, http://hdl.handle.net/1765/37392

Hoogendoorn, B., *Social Entrepreneurship in the Modern Economy: Warm Glow, Cold Feet*, Promotor(s): Prof.dr. H.P.G. Pennings & Prof.dr. A.R. Thurik, EPS-2011-246-STR, http://hdl.handle.net/1765/26447

Hoogervorst, N., *On The Psychology of Displaying Ethical Leadership: A Behavioral Ethics Approach*, Promotor(s): Prof.dr. D. De Cremer & Dr. M. van Dijke, EPS-2011-244-ORG, http://hdl.handle.net/1765/26228

Huang, X., *An Analysis of Occupational Pension Provision: From Evaluation to Redesign*, Promotor(s): Prof.dr. M.J.C.M. Verbeek & Prof.dr. R.J. Mahieu, EPS-2010-196-F&A, http://hdl.handle.net/1765/19674

Hytönen, K.A. *Context Effects in Valuation, Judgment and Choice*, Promotor(s): Prof.dr.ir. A. Smidts, EPS-2011-252-MKT, http://hdl.handle.net/1765/30668

Iseger, P. den, *Fourier and Laplace Transform Inversion with Application in Finance,* Promotor: Prof.dr.ir. R.Dekker, EPS-2014-322-LIS, http://hdl.handle.net/1765/1

Jaarsveld, W.L. van, *Maintenance Centered Service Parts Inventory Control*, Promotor(s): Prof.dr.ir. R. Dekker, EPS-2013-288-LIS, http://hdl.handle.net/1765/ 39933

Jalil, M.N., *Customer Information Driven After Sales Service Management: Lessons from Spare Parts Logistics*, Promotor(s): Prof.dr. L.G. Kroon, EPS-2011-222-LIS, http://hdl.handle.net/1765/22156

Kagie, M., *Advances in Online Shopping Interfaces: Product Catalog Maps and Recommender Systems*, Promotor(s): Prof.dr. P.J.F. Groenen, EPS-2010-195-MKT, http://hdl.handle.net/1765/19532

Kappe, E.R., *The Effectiveness of Pharmaceutical Marketing*, Promotor(s): Prof.dr. S. Stremersch, EPS-2011-239-MKT, http://hdl.handle.net/1765/23610

Karreman, B., *Financial Services and Emerging* Markets, Promotor(s): Prof.dr. G.A. van der Knaap & Prof.dr. H.P.G. Pennings, EPS-2011-223-ORG, http://hdl.handle.net/1765/ 22280

Kil, J.C.M., *Acquisitions Through a Behavioral and Real Options Lens*, Promotor(s): Prof.dr. H.T.J. Smit, EPS-2013-298-F&A, http://hdl.handle.net/1765/50142

Klooster, E. van't, *Travel to Learn: The Influence of Cultural Distance on Competence Development in Educational Travel,* Promotors: Prof.dr. F.M. Go & Prof.dr. P.J. van Baalen, EPS-2014-312-MKT, http://hdl.handle.net/1765/151460

Koendjbiharie, S.R., *The Information-Based View on Business Network Performance Revealing the Performance of Interorganizational Networks,* Promotors: Prof.dr.ir. H.W.G.M. van Heck & Prof.mr.dr. P.H.M. Vervest, EPS-2014-315-LIS, http://hdl.handle.net/1765/1

Koning, M., *The Financial Reporting Environment: taking into account the media, international relations and auditors,* Promotor(s): Prof.dr. P.G.J.Roosenboom & Prof.dr. G.M.H. Mertens, EPS-2014-330-F&A, http://hdl.handle.net/1765/1

Konter, D.J., *Crossing borders with HRM: An inquiry of the influence of contextual differences in the adaption and effectiveness of HRM*, Promotor: Prof.dr. J. Paauwe, EPS-2014-305-ORG, http://hdl.handle.net/1765/1

Korkmaz, E. *Understanding Heterogeneity in Hidden Drivers of Customer Purchase Behavior,* Promotors: Prof.dr. S.L. van de Velde & dr. R.Kuik, EPS-2014-316-LIS, http://hdl.handle.net/1765/1

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://hdl.handle.net/1765/1

Lam, K.Y., *Reliability and Rankings*, Promotor(s): Prof.dr. P.H.B.F. Franses, EPS-2011-230-MKT, http://hdl.handle.net/1765/22977

Lander, M.W., *Profits or Professionalism? On Designing Professional Service Firms*, Promotor(s): Prof.dr. J. van Oosterhout & Prof.dr. P.P.M.A.R. Heugens, EPS-2012-253-ORG, http://hdl.handle.net/1765/30682

Langhe, B. de, *Contingencies: Learning Numerical and Emotional Associations in an Uncertain World*, Promotor(s): Prof.dr.ir. B. Wierenga & Prof.dr. S.M.J. van Osselaer, EPS-2011-236-MKT, http://hdl.handle.net/1765/23504

Larco Martinelli, J.A., *Incorporating Worker-Specific Factors in Operations Management Models*, Promotor(s): Prof.dr.ir. J. Dul & Prof.dr. M.B.M. de Koster, EPS-2010-217-LIS, http://hdl.handle.net/1765/21527

Leunissen, J.M., *All Apologies: On the Willingness of Perpetrators to Apoligize*, Promotor: Prof.dr. D. De Cremer, EPS-2014-301-ORG, http://hdl.handle.net/1765/1

Liang, Q., *Governance, CEO Indentity, and Quality Provision of Farmer Cooperatives*, Promotor: Prof.dr. G.W.J. Hendrikse, EPS-2013-281-ORG, http://hdl.handle.net/1765/1

Liket, K.C., *Why 'Doing Good' is not Good Enough: Essays on Social Impact Measurement*, Promotor: Prof.dr. H.R. Commandeur, EPS-2014-307-S&E, http://hdl.handle.net/1765/51130

Loos, M.J.H.M. van der, *Molecular Genetics and Hormones; New Frontiers in Entrepreneurship Research*, Promotor(s): Prof.dr. A.R. Thurik, Prof.dr. P.J.F. Groenen & Prof.dr. A. Hofman, EPS-2013-287-S&E, http://hdl.handle.net/1765/ 40081

Lovric, M., *Behavioral Finance and Agent-Based Artificial Markets*, Promotor(s): Prof.dr. J. Spronk & Prof.dr.ir. U. Kaymak, EPS-2011-229-F&A, http://hdl.handle.net/1765/ 22814

Lu, Y., *Data-Driven Decision Making in Auction Markets, Promotors:* Prof.dr.ir.H.W.G.M. van Heck & Prof.dr.W.Ketter, EPS-2014-314-LIS, http://hdl.handle.net/1765/1

Markwat, T.D., *Extreme Dependence in Asset Markets Around the Globe*, Promotor: Prof.dr. D.J.C. van Dijk, EPS-2011-227-F&A, http://hdl.handle.net/1765/22744

Mees, H., *Changing Fortunes: How China's Boom Caused the Financial Crisis*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2012-266-MKT, http://hdl.handle.net/1765/34930

Meuer, J., *Configurations of Inter-Firm Relations in Management Innovation: A Study in China's Biopharmaceutical Industry*, Promotor: Prof.dr. B. Krug, EPS-2011-228-ORG, http://hdl.handle.net/1765/22745

Mihalache, O.R., *Stimulating Firm Innovativeness: Probing the Interrelations between Managerial and Organizational Determinants*, Promotor(s): Prof.dr. J.J.P. Jansen, Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2012-260-S&E, http://hdl.handle.net/1765/32343

Milea, V., *New Analytics for Financial Decision Support*, Promotor: Prof.dr.ir. U. Kaymak, EPS-2013-275-LIS, http://hdl.handle.net/1765/ 38673

Naumovska, I. *Socially Situated Financial Markets:a Neo-Behavioral Perspective on Firms, Investors and Practices,* Promoter(s) Prof.dr. P.P.M.A.R. Heugens & Prof.dr. A.de Jong, EPS-2014-319-S&E, http://hdl.handle.net/1765/ 1

Nielsen, L.K., *Rolling Stock Rescheduling in Passenger Railways: Applications in Short-term Planning and in Disruption Management*, Promotor: Prof.dr. L.G. Kroon, EPS-2011-224-LIS, http://hdl.handle.net/1765/22444

Nijdam, M.H., *Leader Firms: The Value of Companies for the Competitiveness of the Rotterdam Seaport Cluster*, Promotor(s): Prof.dr. R.J.M. van Tulder, EPS-2010-216-ORG, http://hdl.handle.net/1765/21405

Noordegraaf-Eelens, L.H.J., *Contested Communication: A Critical Analysis of Central Bank Speech*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2010-209-MKT, http://hdl.handle.net/1765/21061

Nuijten, A.L.P., *Deaf Effect for Risk Warnings: A Causal Examination applied to Information Systems Projects*, Promotor: Prof.dr. G. van der Pijl & Prof.dr. H. Commandeur & Prof.dr. M. Keil, EPS-2012-263-S&E, http://hdl.handle.net/1765/34928

Oosterhout, M., van, *Business Agility and Information Technology in Service Organizations*, Promotor: Prof,dr.ir. H.W.G.M. van Heck, EPS-2010-198-LIS, http://hdl.handle.net/1765/19805

Osadchiy, S.E., *The Dynamics of Formal Organization: Essays on Bureaucracy and Formal Rules*, Promotor: Prof.dr. P.P.M.A.R. Heugens, EPS-2011-231-ORG, http://hdl.handle.net/1765/23250

Otgaar, A.H.J., *Industrial Tourism: Where the Public Meets the Private*, Promotor: Prof.dr. L. van den Berg, EPS-2010-219-ORG, http://hdl.handle.net/1765/21585

Ozdemir, M.N., *Project-level Governance, Monetary Incentives and Performance in Strategic R&D Alliances*, Promotor: Prof.dr.ir. J.C.M. van den Ende, EPS-2011-235-LIS, http://hdl.handle.net/1765/23550

Peers, Y., *Econometric Advances in Diffusion Models*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2011-251-MKT, http://hdl.handle.net/1765/ 30586

Pince, C., *Advances in Inventory Management: Dynamic Models*, Promotor: Prof.dr.ir. R. Dekker, EPS-2010-199-LIS, http://hdl.handle.net/1765/19867

Porck, J.P., *No Team is an Island*, Promotor: Prof.dr. P.J.F. Groenen & Prof.dr. D.L. van Knippenberg, EPS-2013-299-ORG, http://hdl.handle.net/1765/50141

Porras Prado, M., *The Long and Short Side of Real Estate, Real Estate Stocks, and Equity*, Promotor: Prof.dr. M.J.C.M. Verbeek, EPS-2012-254-F&A, http://hdl.handle.net/1765/30848

Potthoff, D., *Railway Crew Rescheduling: Novel Approaches and Extensions*, Promotor(s): Prof.dr. A.P.M. Wagelmans & Prof.dr. L.G. Kroon, EPS-2010-210-LIS, http://hdl.handle.net/1765/21084

Poruthiyil, P.V., *Steering Through: How Organizations Negotiate Permanent Uncertainty and Unresolvable Choices*, Promotor(s): Prof.dr. P.P.M.A.R. Heugens & Prof.dr. S. Magala, EPS-2011-245-ORG, http://hdl.handle.net/1765/26392

Pourakbar, M. *End-of-Life Inventory Decisions of Service Parts*, Promotor: Prof.dr.ir. R. Dekker, EPS-2011-249-LIS, http://hdl.handle.net/1765/30584

Pronker, E.S., *Innovation Paradox in Vaccine Target Selection*, Promotor(s): Prof.dr. H.R. Commandeur & Prof.dr. H.J.H.M. Claassen, EPS-2013-282-S&E, http://hdl.handle.net/1765/39654

Retel Helmrich, M.J., *Green Lot-Sizing*, Promotor: Prof.dr. A.P.M. Wagelmans, EPS-2013-291-LIS, http://hdl.handle.net/1765/41330

Rietveld, C.A., *Essays on the Intersection of Economics and Biology*, Promotor(s): Prof.dr. P.J.F. Groenen, Prof.dr. A. Hofman, Prof.dr. A.R. Thurik, Prof.dr. P.D. Koellinger, EPS-2014-320-S&E, http://hdl.handle.net/1765/1

Rijsenbilt, J.A., *CEO Narcissism; Measurement and Impact*, Promotor: Prof.dr. A.G.Z. Kemna & Prof.dr. H.R. Commandeur, EPS-2011-238-STR, http://hdl.handle.net/1765/ 23554

Roelofsen, E.M., *The Role of Analyst Conference Calls in Capital Markets*, Promotor(s): Prof.dr. G.M.H. Mertens & Prof.dr. L.G. van der Tas RA, EPS-2010-190-F&A, http://hdl.handle.net/1765/18013

Roza, M.W., *The Relationship between Offshoring Strategies and Firm Performance: Impact of Innovation, Absorptive Capacity and Firm Size*, Promotor(s): Prof.dr. H.W. Volberda & Prof.dr.ing. F.A.J. van den Bosch, EPS-2011-214-STR, http://hdl.handle.net/1765/22155

Rubbaniy, G., *Investment Behavior of Institutional Investors*, Promotor: Prof.dr. W.F.C. Verschoor, EPS-2013-284-F&A, http://hdl.handle.net/1765/ 40068

Schellekens, G.A.C., *Language Abstraction in Word of Mouth*, Promotor: Prof.dr.ir. A. Smidts, EPS-2010-218-MKT, http://hdl.handle.net/1765/21580

Shahzad, K., *Credit Rating Agencies, Financial Regulations and the Capital Markets*, Promotor: Prof.dr. G.M.H. Mertens, EPS-2013-283-F&A, http://hdl.handle.net/1765/39655

Sotgiu, F., *Not All Promotions are Made Equal: From the Effects of a Price War to Cross-chain Cannibalization*, Promotor(s): Prof.dr. M.G. Dekimpe & Prof.dr.ir. B. Wierenga, EPS-2010-203-MKT, http://hdl.handle.net/1765/19714

Sousa, M., *Servant Leadership to the Test: New Perspectives and Insight,* Promotors: Prof.dr. D. van Knippenberg & Dr. D. van Dierendonck, EPS-2014-313-ORG, http://hdl.net/1765/1

Spliet, R., *Vehicle Routing with Uncertain Demand*, Promotor: Prof.dr.ir. R. Dekker, EPS-2013-293-LIS, http://hdl.handle.net/1765/41513

Srour, F.J., *Dissecting Drayage: An Examination of Structure, Information, and Control in Drayage Operations*, Promotor: Prof.dr. S.L. van de Velde, EPS-2010-186-LIS, http://hdl.handle.net/1765/18231

Staadt, J.L., *Leading Public Housing Organisation in a Problematic Situation: A Critical Soft Systems Methodology Approach*, Promotor: Prof.dr. S.J. Magala, EPS-2014-308-ORG, http://hdl.handle.net/1765/50712

Stallen, M., *Social Context Effects on Decision-Making; A Neurobiological Approach*, Promotor: Prof.dr.ir. A. Smidts, EPS-2013-285-MKT, http://hdl.handle.net/1765/ 39931

Tarakci, M., *Behavioral Strategy; Strategic Consensus, Power and Networks*, Promotor(s): Prof.dr. P.J.F. Groenen & Prof.dr. D.L. van Knippenberg, EPS-2013-280-ORG, http://hdl.handle.net/1765/ 39130

Teixeira de Vasconcelos, M., *Agency Costs, Firm Value, and Corporate Investment*, Promotor: Prof.dr. P.G.J. Roosenboom, EPS-2012-265-F&A, http://hdl.handle.net/1765/37265

Tempelaar, M.P., *Organizing for Ambidexterity: Studies on the Pursuit of Exploration and Exploitation through Differentiation, Integration, Contextual and Individual Attributes*, Promotor(s): Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-191-STR, http://hdl.handle.net/1765/18457

Tiwari, V., *Transition Process and Performance in IT Outsourcing: Evidence from a Field Study and Laboratory Experiments*, Promotor(s): Prof.dr.ir. H.W.G.M. van Heck & Prof.dr. P.H.M. Vervest, EPS-2010-201-LIS, http://hdl.handle.net/1765/19868

Tröster, C., *Nationality Heterogeneity and Interpersonal Relationships at Work*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2011-233-ORG, http://hdl.handle.net/1765/23298

Tsekouras, D., *No Pain No Gain: The Beneficial Role of Consumer Effort in Decision Making*, Promotor: Prof.dr.ir. B.G.C. Dellaert, EPS-2012-268-MKT, http://hdl.handle.net/1765/ 37542

Tzioti, S., *Let Me Give You a Piece of Advice: Empirical Papers about Advice Taking in Marketing*, Promotor(s): Prof.dr. S.M.J. van Osselaer & Prof.dr.ir. B. Wierenga, EPS-2010-211-MKT, hdl.handle.net/1765/21149

Vaccaro, I.G., *Management Innovation: Studies on the Role of Internal Change Agents*, Promotor(s): Prof.dr. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-212-STR, hdl.handle.net/1765/21150

Vagias, D., *Liquidity, Investors and International Capital Markets*, Promotor: Prof.dr. M.A. van Dijk, EPS-2013-294-F&A, http://hdl.handle.net/1765/41511

Veelenturf, L.P., *Disruption Management in Passenger Railways: Models for Timetable, Rolling Stock and Crew Rescheduling,* Promotor: Prof.dr. L.G. Kroon, EPS-2014-327-LIS, http://hdl.handle.net/1765/1

Verheijen, H.J.J., *Vendor-Buyer Coordination in Supply Chains*, Promotor: Prof.dr.ir. J.A.E.E. van Nunen, EPS-2010-194-LIS, http://hdl.handle.net/1765/19594

Venus, M., *Demystifying Visionary Leadership; In Search of the Essence of Effective Vision Communication*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2013-289-ORG, http://hdl.handle.net/1765/ 40079

Visser, V., *Leader Affect and Leader Effectiveness; How Leader Affective Displays Influence Follower Outcomes*, Promotor: Prof.dr. D. van Knippenberg, EPS-2013-286-ORG, http://hdl.handle.net/1765/40076

Vlam, A.J., *Customer First? The Relationship between Advisors and Consumers of Financial Products*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2011-250-MKT, http://hdl.handle.net/1765/30585

Waard, E.J. de, *Engaging Environmental Turbulence: Organizational Determinants for Repetitive Quick and Adequate Responses*, Promotor(s): Prof.dr. H.W. Volberda & Prof.dr. J. Soeters, EPS-2010-189-STR, http://hdl.handle.net/1765/18012

Waltman, L., *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*, Promotor(s): Prof.dr.ir. R. Dekker & Prof.dr.ir. U. Kaymak, EPS-2011-248-LIS, http://hdl.handle.net/1765/26564

Wang, Y., *Information Content of Mutual Fund Portfolio Disclosure*, Promotor: Prof.dr. M.J.C.M. Verbeek, EPS-2011-242-F&A, http://hdl.handle.net/1765/26066

Wang, Y., *Corporate Reputation Management; Reaching Out to Find Stakeholders*, Promotor: Prof.dr. C.B.M. van Riel, EPS-2013-271-ORG, http://hdl.handle.net/1765/ 38675

Weenen, T.C., *On the Origin and Development of the Medical Nutrition Industry*, Promotors: Prof.dr. H.R. Commandeur & Prof.dr. H.J.H.M. Claassen, EPS-2014-309-S&E, http://hdl.handle.net/1765/51134

Wolfswinkel, M., *Corporate Governance, Firm Risk and Shareholder Value of Dutch Firms*, Promotor: Prof.dr. A. de Jong, EPS-2013-277-F&A, http://hdl.handle.net/1765/ 39127

Xu, Y., *Empirical Essays on the Stock Returns, Risk Management, and Liquidity Creation of Banks*, Promotor: Prof.dr. M.J.C.M. Verbeek, EPS-2010-188-F&A, http://hdl.handle.net/1765/18125

Yang, S.Y., *Information aggregation efficiency of prediction market*, Promotor: Prof.dr.ir. H.W.G.M. van Heck, EPS-2014-323-LIS, http://hdl.handle.net/1765/1

Zaerpour, N., *Efficient Management of Compact Storage Systems*, Promotor: Prof.dr. M.B.M. de Koster, EPS-2013-276-LIS, http://hdl.handle.net/1765/38766

Zhang, D., *Essays in Executive Compensation*, Promotor: Prof.dr. I. Dittmann, EPS-2012-261-F&A, http://hdl.handle.net/1765/32344

Zhang, X., *Scheduling with Time Lags*, Promotor: Prof.dr. S.L. van de Velde, EPS-2010-206-LIS, http://hdl.handle.net/1765/19928

Zhou, H., *Knowledge, Entrepreneurship and Performance: Evidence from Country-level and Firm-level Studies*, Promotor(s): Prof.dr. A.R. Thurik & Prof.dr. L.M. Uhlaner, EPS-2010-207-ORG, http://hdl.handle.net/1765/20634

Zwan, P.W. van der, *The Entrepreneurial Process: An International Analysis of Entry and Exit*, Promotor(s): Prof.dr. A.R. Thurik & Prof.dr. P.J.F. Groenen, EPS-2011-234-ORG, http://hdl.handle.net/1765/23422

Erasmus Research Institute of Management - ERiM

DISRUPTION MANAGEMENT IN PASSENGER RAILWAYS

MODELS FOR TIMETABLE, ROLLING STOCK AND CREW RESCHEDULING

Every day a significant number of people choose for the railways as a comfortable and sustainable way of transportation. In order to accommodate the journeys of a large number of railway passengers, extensive planning is necessary. Unfortunately, the execution of the plans is frequently disrupted by unexpected events. For railway operators it is quite a challenge to deal with these disruptions as even small deviations from the plan can have large influences on the timetable, the rolling stock schedule and the crew schedule. More severely, these events reduce the available transport capacity and interrupt the mobility of the passengers.

This thesis discusses several models and solution approaches for railway disruption management based on algorithmic techniques from Operations Research. The main focus is to reduce the inconvenience passengers experience during disruptions. This is achieved by improving the disruption management approaches for timetable, rolling stock and crew rescheduling proposed within the scientific community. The existing models are extended by introducing greater flexibility, e.g. allowing small delays in the crew rescheduling or addition stops in the rolling stock rescheduling. As a result fewer trains are cancelled during disruptions and passengers have more options to reach their destination. Although some inconvenience will remain, as much as possible is mitigated.

Design & layout: B&T Ontwerp en advies (www.b-en-t.nl) Print: Haveka (www.haveka.nl)

ERiM

The Erasmus Research Institute of Management (ERIM) is the Research School (Onderzoekschool) 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 undertaken 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 forefront of creating new business knowledge.

ERIM PhD Series
Research in Management

Erasmus Research Institute of Management - ERiM
Rotterdam School of Management (RSM)
Erasmus School of Economics (ESE)
Erasmus University Rotterdam (EUR)
P.O. Box 1738, 3000 DR Rotterdam,
The Netherlands

Tel. +31 10 408 11 82
Fax +31 10 408 96 40
E-mail info@erim.eur.nl
Internet www.erim.eur.nl