DANIEL POTTHOFF

Railway Crew Rescheduling

Novel Approaches and Extensions

Railway Crew Rescheduling: Novel Approaches and Extensions

Daniel Potthoff

Railway Crew Rescheduling: Novel approaches and extensions

Bijsturen van rijdend personeel:

Nieuwe aanpakken en uitbreidingen

PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Erasmus Universiteit Rotterdam
op gezag van de rector magnificus

Prof.dr. H.G. Schmidt

en volgens besluit van het College voor Promoties.

De openbare verdediging zal plaatsvinden op

donderdag 21 oktober 2010 om 15:30 uur

door

Daniel Potthoff
geboren te Salzkotten, Duitsland.

ERASMUS UNIVERSITEIT ROTTERDAM

Promotiecommissie

Promotoren:	Prof.dr. A.P.M. Wagelmans
	Prof.dr. L.G. Kroon
Overige leden:	Prof.dr.ir. R. Dekker
	Prof.dr. L. Suhl
	Prof. J. Clausen
Copromotor:	Dr. D. Huisman

Erasmus Research Institute of Management – ERIM
Rotterdam School of Management (RSM)
Erasmus School of Economics (ESE)
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, 210
ERIM reference number: EPS–2010–210–LIS
ISBN 978–90–5892–250–2

Design: B&T Ontwerp en advies www.b-en-t.nl
Print: Haveka www.haveka.nl
Cover design: Florian Potthoff

This thesis was prepared with LaTeX2e.

Acknowledgements

The story behind this thesis began in September 2005 when I had a first meeting with my team of supervisors Albert Wagelmans, Leo Kroon, and Dennis Huisman. I still remember the good atmosphere and the fruitful discussions during that meeting. This resulted in an exciting PhD trajectory. Now, at the end of this trajectory, I would like to take the opportunity to thank the many people that helped me finishing this thesis.

Foremost, I would like to thank my daily supervisor Dennis Huisman for the tremendous amount of time he spent on my supervision and on our joint research. Dennis, because of your competence and your enthusiasm, working with you in close collaboration was a great pleasure for me. In writing this thesis I benefited quite a lot from your target-oriented approach. Furthermore, I appreciate you as travel companion. I especially enjoyed the trips in Thailand and Norway around the TRISTAN VI and VII conferences.

I am very grateful to my supervisors Albert Wagelmans and Leo Kroon. They guided and supported me during the last years. Albert, thank you for your excellent comments and remarks on several drafts. Assisting you in teaching mathematics for economists was an interesting experience and I learned a lot, not only about teaching, during this period. Leo, thank you very much for sharing your great enthusiasm about Operations Research and railways with me. I would also like to thank you for your invaluable comments on my thesis during all stages of its becoming.

Next, I would like to thank the members of the inner committee, Rommert Dekker, Leena Suhl, and Jens Clausen, for evaluating this thesis. Furthermore, I thank Bert Meerstadt, Ernesto Morgado, and Steef van de Velde, for their membership in the PhD committee.

Furthermore, I would like to thank all my co-authors: Jens Clausen, Guy Desaulniers, Dennis Huisman, Julie Jespersen-Groth, Leo Kroon, Gábor Maróti, Morten Nyhave Nielsen, and Luuk Veelenturf. Moreover, I want to thank Guy Desaulniers and Issmail Elhallaoui for their cooperation and hospitality during my visit to Montreal.

During the first two years of my research I spent a lot of time in the H-building of the Erasmus University Rotterdam (EUR). I really enjoyed my time there and I would like to thank my former colleagues Bram, Diana, Jeroen, Ludo, Remco, Rene, Merel, and Nees Jan for at least three of following: having lunch together, sharing information about all kind of things related and unrelated to the PhD research, doing exercises for PhD courses, and providing some distraction by playing the game *Koehandel.* Moreover, I thank Lars for doing exercises and for the good times at many conferences. I owe thanks to my officemate Çerağ for his enjoyable company during my four years at the EUR. Many thanks also to Martijn and Twan for bringing more life and the Dutch point of view into our office. Furthermore, I thank Gabriella for providing a roof during my first six weeks in the Netherlands. Many special thanks to Wilco for always being helpful. I also like to thank all members of the Econometric Institute. I also thank Vereniging Trustfonds EUR for supporting my research visit at the Group for Research in Decision Analysis (GERAD) in Montreal.

Due to the subject of my research I happened to also spend quite some time at the Innovations Group at Netherlands Railways. For the good work atmosphere, and many interesting discussions I would like to thank Dirk, Erwin, Joël, Luuk, Pieter-Jan, Ramon, Stefan, and the many master's students.

Now, I want to address a few words to my paranymphs. Robin, I would like to thank you for your kindness, your helpfulness, and especially for your patience and corrections when I started to try my Dutch on a real person. Gábor, thank you for the discussions about mathematics, programming, life in general, and for convincing me that Perl comes in handy on many occasions which inspired me to learn Python.

I would also like to thank my parents and my brother for their encouragement and understanding. Gabi, Bernd, I am deeply grateful for all our conversations: They have always been a great decision support. Florian, I owe you thanks for always being helpful and for designing the cover image.

Finally, a very special 'thank you' goes to Vera for cheering me up whenever there have been setbacks. I am very grateful for our joint time in Utrecht.

Utrecht, August 2010
Daniel Potthoff

Contents

Chapter 1

Introduction

Passenger railway transportation plays an important role in everyday life in many countries. Railway companies offer train services according to well designed timetables. Schedules for rolling stock and crew must not only make efficient use of the resources, they should also be robust and recoverable, since railway operations are exposed to unforeseen events such as infrastructure malfunctions, weather conditions, accidents, etc. Disruption management is the process of reacting to events that make it impossible to operate the timetable and the resource schedules as planned. Within disruption management, crew rescheduling is, next to timetable adjustment and rolling stock rescheduling, one of the major tasks.

This thesis deals with railway crew rescheduling. In particular we are interested in the development of mathematical models for crew rescheduling problems. Due to the operational context, we need to design efficient algorithms to solve these models to near optimality within a couple of minutes of computation time. These algorithms will be based on *Operations Research* techniques. We will evaluate the applicability of the proposed models and algorithms using real-life data from Netherlands Railways (Nederlandse Spoorwegen, or NS), the largest passenger operator in the Netherlands. The models and algorithms should, however, be applicable also to crew rescheduling problems at other companies. We will describe the dependencies and interactions of timetable adjustment, rolling stock rescheduling, and crew rescheduling, but models and algorithms for timetabling and rolling stock rescheduling are outside the scope of this thesis.

Let us now examine the role of passenger railway traffic in the Netherlands, a small and densely populated country in Western Europe. Railway traffic is the backbone of the Dutch public transport system. The market share of rail in commuter traffic during rush hours between the four major cities –Amsterdam, Rotterdam, The Hague, and Utrecht– is above 50 percent. Reliable railway operations are vital to the Dutch economy, because the already congested highway system between those cities would collapse if it also had to accommodate the commuters that are currently using the train.

The Dutch rail infrastructure is owned by the state. The state-owned, non-profit organization ProRail is responsible for maintaining and allocating the infrastructure. A number of freight operators make use of the same infrastructure as the passenger operators. However, about 95 percent of the trains are passenger trains. The yearly total amount of passenger transport by rail was 16.3 billion passenger kilometers in 2009 (Netherlands Railways (2010)).The Netherlands have the highest ratio of passenger kilometers over line kilometers within Europe. On an average workday about 1.2 million passengers travel by train.

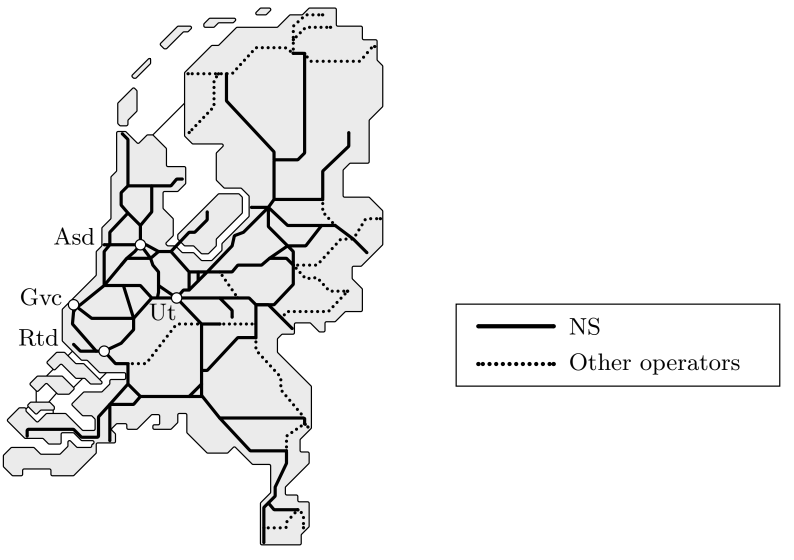


Figure 1.1: The Dutch passenger railway network as of January 2010 (now).

NS owns a license to operate passenger trains on all main lines until 2015. Approximately 90 percent of the passenger demand occurs on these lines. Moreover, NS operates passenger trains on some smaller lines. The completed network on which NS operates passenger trains is shown in Figure 1.1. In order to operate its timetable, NS employs around 6,000 crew members (train drivers and conductors) of which circa 1,000 train drivers and 1,300 conductors are on duty on an average workday. Each crew member is assigned to one of the 29 crew bases. NS has contracts with the Dutch government where, among others, targets for four key performance indicators, punctuality, percentage of canceled trains, percentage of maintained connections, and customer satisfaction are specified. The latter target itself consists of several points. If NS misses some of these targets, they have to pay a contractual fee and the government reserves the right to withdraw the license.

Therefore, good disruption management, including crew rescheduling, is vital not only for NS but for the Dutch economy as well. Its importance puts the reliability of railway traffic in the focus of Dutch society as illustrated by the following example.

During a period of ice and snow, the railway system almost broke down completely for several days in December 2009. The discussion about the causes and possible consequences made national headlines in the Dutch media on several days. On January 20, 2010 the so-called “rail chaos” was topic of a debate in the Dutch parliament. The aims of the debate were to find out about the reasons for the railway system to brake down and discuss about a possible course of action for the ministry of transportation. Frozen switches have been the major cause of the problems. However, it is evident from log files of some control centers from ProRail that have been published by a Dutch television station (RTL (2010)) that the problem was getting even worse since NS was not able to reschedule its train drivers. This resulted in a number of trains that did not depart, even though the infrastructure was available. This weather induced incident clearly indicates the importance of decision support tools for crew rescheduling in a disrupted situation which is the topic of this thesis. Of course, this is a very extreme example. Nonetheless, smaller but still significant disruptions happen several times per day.

1.1 Contributions of the thesis

The main results of this thesis belong to two categories. First we contribute to the scientific literature. This is done by presenting novel algorithms for railway crew rescheduling. Moreover, we consider two extensions to the basic crew rescheduling problem. The first extension integrates crew rescheduling with the possibility to delay the departure of some trains, which is called retiming, in order to allow better solutions. The second extension considers the uncertainty in the disrupted situation. The latter topic has not been discussed in the literature. We present new optimization models for these extensions along with sophisticated algorithms.

Our second contribution is that we provide computational evidence that the proposed models and algorithms can be applied in a practical setting. This follows from our computational experiments on real-life data from NS, where we could show that the proposed algorithms find good solutions to the considered optimization models within a couple of minutes of computation time on a desktop PC. This means that next to its scientific contribution, this thesis can be seen as a proof of concept for Operations Research based decision support for railway crew rescheduling. The application of decision support as presented in this thesis is a key to limit the consequences of disruptions. With decision support for crew rescheduling available, NS will be able to react much better to disruptions and consequently will be able to provide a better service to its customers.

1.2 Overview of the thesis

In this thesis we present mathematical formulations and algorithms to solve crew rescheduling problems that arise at passenger railway operators. The thesis is set up as follows.

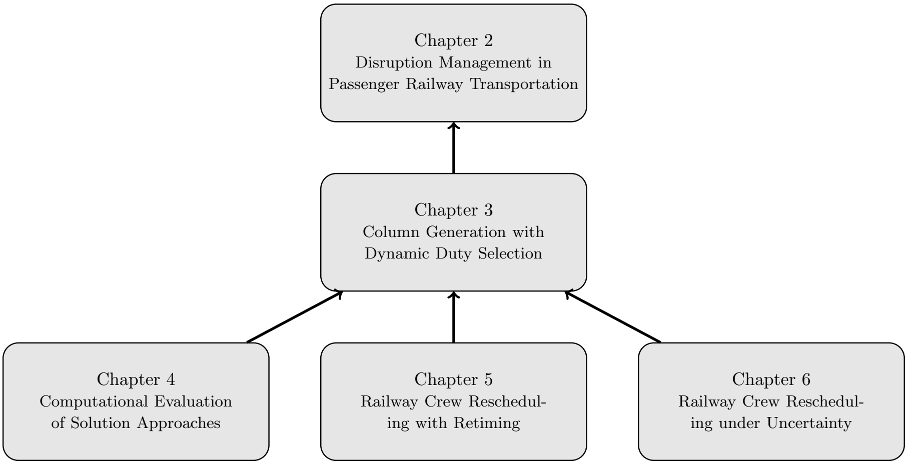


Figure 1.2: Graph showing the dependencies of Chapters 2–6.

In Chapter 2 we describe disruption management for passenger railway operators as a whole. We outline the three major tasks of the disruption management process, namely: (i) timetable update, (ii) rolling stock rescheduling, and (iii) crew rescheduling. Furthermore, we discuss the interdependencies between these tasks and we present an overview of the existing literature.

A novel algorithm to solve the crew rescheduling problem is presented in Chapter 3. First of all, we present a fast heuristic based on the combination of Lagrangian relaxation, Lagrangian heuristics and column generation. This heuristic is used in an iterative neighborhood exploration approach to tackle difficult problem instances using only a short amount of computation time. In the remainder this algorithm will be called Column Generation with Dynamic Duty Selection (CGDDS). We present computational results using real-life data from NS that illustrate the applicability of our solution approach.

In Chapter 4 we compare the solution approach presented in Chapter 3 with two alternative approaches. The first of these two is a greedy two phase heuristics that tries to mimic manual rescheduling. The second alternative approach is a heuristic based on dynamic constraint aggregation, a relatively new advanced column generation method. Comparing the results of the three methods for real-life data form NS we show that the greedy two phase heuristic fails to find good solutions for the larger disruptions.

Moreover, the quality of the solutions of this heuristic heavily depends on the availability of drivers on stand-by. The dynamic constraint aggregation heuristic is able to find good quality solutions, but in terms of computation time it is not competitive with the CGDDS approach. From the comparison of the three methods we conclude that the CGDDS approach is the best method for crew rescheduling since it finds solutions of good quality in a short amount of computation time.

Chapter 5 deals with an extension of the basic crew rescheduling problem. In this extension we allow to slightly change the timetable. The problem is motivated by the interdependencies between timetable adjustment and crew rescheduling as discussed in Chapter 2. The goal of the problem considered in Chapter 5 is to improve the overall disruption management process. We model the modifications to the timetable as a discrete choice of different departure times for some trains, or in other words we allow retiming for some trains. We propose a new solution approach based on the solution approach presented in Chapter 3. Our computational experiments, again with data provided by NS, point out the potential of crew rescheduling with retiming to improve the overall disruption management process.

In Chapter 6 we consider the uncertainty of the duration of a disruption and its consequences for crew rescheduling. This aspect is a new field in the scientific literature. We show that crew rescheduling under uncertainty can be seen as a two phase stochastic optimization problem with different scenarios representing the timetables that could be operated depending on the duration of the disruption. The real challenge in this problem lies in the fact that the computation time available for crew rescheduling is very limited. Therefore, we propose a quasi robust optimization model in order to take the uncertainty into account. Moreover, we show how the solution approach from Chapter 3 can be modified in order to solve the quasi robust optimization model. Finally, the quasi robust optimization approach is compared with an approach that only considers the expected scenario. Again, we use real-life data from NS for this comparison and we show that the quasi robust optimization approach gives more robust solutions for most of the instances.

The dependencies between Chapter 2–6 are shown in Figure 1.2. Chapter 3 builds on Chapter 2 and Chapters 4–6 build on Chapter 3. For the reader this suggests that after reading Chapters 2 and 3 the remaining chapters can be read in any order.

Finally, we summarize our main results in Chapter 7. We finish by giving some advice on steps to be taken to make Operations Research based decision support available for the crew dispatchers of NS.

Chapter 2

Disruption Management in Passenger Railway Transportation

2.1 Introduction

Many Europeans travel frequently by train, either to commute or in their leisure time. Therefore, the operational performance of railway systems is often discussed in the public debate. Travelers expect to arrive at a specific time at their destination. If they travel by rail, they expect to arrive more or less at the time published in the timetable. However, unforeseen events often take place, which cause delays or even cancellations of trains. As a result, passengers arrive later than expected at their final destinations. Due to missed connections, the delay of a passenger can be even much larger than the delays of his individual trains.

Due to the importance for the public on one hand and the deregulation of the railway market on the other, railway operators now put more emphasis on their operational performance than in the past. Furthermore, due to the separation of the management of the infrastructure and the operations in many European countries (including the Netherlands), several organizations are responsible for the performance of the railway system.

This chapter deals with passenger railway transport only. However, in addition to the passenger railway operator itself, the infrastructure manager and other (also cargo) operators have a strong influence on the performance of the railway services of that single operator. Therefore, the role and the objectives of the infrastructure manager and of the operators are also discussed.

We will focus on the situation for Netherlands Railways (NS), which is the main operator in the Netherlands, having the exclusive right to operate passenger trains on the so-called Dutch Main Railway Network until 2015, see Figure 2.1. NS operates a set of lines, where a *line* is defined as a route between a start and an end station and a number of intermediate stops, operated with a certain frequency, e.g. once or twice per hour. The route of the 500 intercity line from Groningen (Gn) to The Hague (Gvc) with stops in Assen (Asn), Zwolle (Zl), Amersfoort (Amf), Utrecht (Ut) and Gouda (Gd) is shown in Figure 2.1.

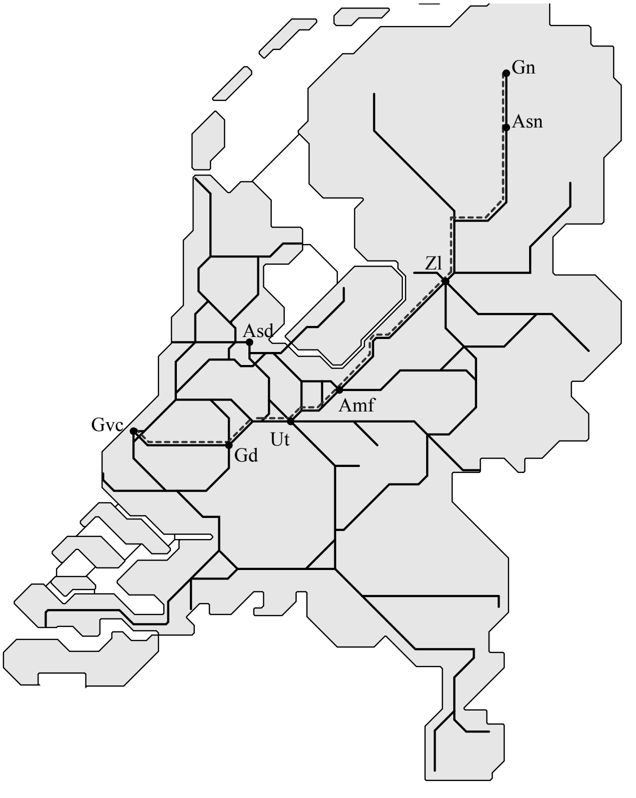


Figure 2.1: The network on which NS operates passenger trains (in 2007). The dashed line shows the 500 intercity line from Gronigen (Gn) to The Hague (Gvc) with its five intermediate stops.

Unfortunately, trains do not always run on time due to unexpected events. Examples are infrastructure malfunctions, rolling stock break downs, accidents, and weather conditions. Such events are called *disruptions*. The Dutch railway network has approximately 17 disruptions related to the infrastructure per day with an average duration of 1.8 hours. About 35% of these infrastructure related disruptions are due to to technical failures, while another 35% is related to third parties (e.g. accidents with other traffic). Next to the disruptions to infrastructure failures, there are also disruptions caused by the operators. The main reasons for the latter are passengers causing longer dwell times, rolling stock problems and delayed crew members. The proportion between the disruptions caused by the operators and the infrastructure is roughly 50-50 in the Netherlands.

Of course, infrastructure managers and operators try to avoid disruptions. Unfortunately, many of them are hard to influence. Therefore, it is very important to limit the consequences of these disruptions. A very common problem in railways is that, due to the strong interdependencies in the railway network and due to cost efficient resource schedules, disruptions are very likely to spread over the network in space and time. This well-known phenomenon is called *knock-on effect*. The key to a good performance of railways is to limit the knock-on effect and thereby to limit the impact of single disruptions. Therefore, operating plans should be robust and effective disruption management is required. In this chapter, we will only look at the second problem. In addition, note that the consequences for passengers can be limited by delaying connecting trains such that passengers can still have their connection even if their arriving train has a delay. This latter problem is known as delay management (Schöbel (2006, 2007)), however this topic falls outside the scope of the current chapter.

So far, Operations Research (OR) models have hardly been applied in practice for disruption management in railway systems. Nevertheless, it is our strong belief that OR models can play an important role to limit the impact of disruptions and thereby to improve the performance of railway systems. This belief is supported by the fact that nowadays OR models and techniques play a major role in several railway companies during the *planning* phase, where the focus is on a good balance of the service level offered to the passengers and the efficiency of the resources rolling stock and crew. The best example is probably the introduction of the new Dutch timetable, for which NS received the 2008 Franz Edelman Award (Kroon et al. (2009)). For an overview on these models and techniques, we refer to surveys of Assad (1980); Cordeau et al. (1998); Huisman et al. (2005b), and Caprara et al. (2007), and to the book of Geraets et al. (2007). Moreover OR models have proven to be quite effective already for supporting disruption management processes in the airline context, see e.g. Yu et al. (2003) and in many other fields (Yu and Qi (2004)).

With this chapter, which is partly based on Jespersen-Groth et al. (2009), we intend to give a comprehensive description of the problems arising in disruption management for railway systems. In this way we set the stage for the remaining chapters of this thesis which deal with crew rescheduling, one of the main subproblems in railway disruption management.

The remainder of this chapter is organized as follows. In Section 2.2 we give a description of disruption management for railway systems, including a description of organizations and actors involved in this process. In Sections 2.3-2.5, we discuss timetabling, rolling stock and crew aspects of the disruption management process. Section 2.6 deals with the advantages and possibilities of integrating some of these processes. Finally, we finish the chapter with some concluding remarks in Section 2.7.

2.2 Description of disruption management

For railway operations we define a *disruption* as an event or a series of events that lead to conflicts in the planned resource schedules for rolling stock, crew, etc.. By definition, a disruption is hence a cause rather than a consequence.

A disruption does not necessarily have immediate influence on the timetable - some disruptions like a track blockage renders the planned timetable immediately infeasible, while others as e.g. shortage of crew due to sickness may lead to cancellations either immediately, in the long run or not at all, depending on the amount of stand-by crew. Note that a disruption leads to a *disrupted situation*. Even though this is a slight abuse of terms, we will occasionally refer to the disrupted situation as the disruption itself.

Accordingly, we define *railway disruption management* as the joint approach of the involved organizations to deal with the impact of disruptions in order to ensure the best possible service for the passengers. This is done by modifying the timetable, and the rolling stock and crew schedules during and after the disruption. The involved organizations are the infrastructure manager and the operators.

Of course, one first has to answer the question if the situation is disrupted, i.e. if the deviation from the original plan is sufficiently large or not. Similar to the airline world (see Kohl et al. (2007)), this question is normally answered by dispatchers monitoring the operations. In the remainder of this chapter, this issue is not considered further.

The Sections 2.2.1 to 2.2.3 introduces a framework of organizations, actors and processes in disruption management, which is valid for several European railway systems. In Section 2.2.4 we discuss the organizational context of the disruption management process.

2.2.1 Organizations

The organizations directly involved in disruption management are the infrastructure manager and the railway operators. These organizations usually have contracts with the involved government. Moreover, they have a certain relationship with each other. These issues are described below.

The infrastructure manager has a contract with the government that obliges it to provide the railway operators with a railway network of a certain infrastructure capacity and reliability. The infrastructure manager has also the responsibility of maintaining the railway network as efficiently as possible.

A passenger railway operator obtains from the government a license to operate passenger trains on the network. The operator is contractually bound to provide a performance that exceeds certain specified thresholds on certain key performance indicators. For example, there may be thresholds for the number of train departures per station, for the (arrival) punctuality at certain stations, for the percentage of maintained connections, for the seating probability, etc. Here, the punctuality is the percentage of trains arriving within for example 3 or 5 minutes of their scheduled arrival time at certain stations. The realization figures on these performance indicators have to be reported to the government periodically. If an operator does not reach one of the thresholds, it has to pay a certain penalty to the government. If the performance is very poor, another operator may get the license to operate trains on the network.

As a consequence, usually the main objective of the railway operator is to meet all thresholds set in the contract with the government at minimum cost. The latter is due to the fact that the railway operators are commercially operating companies. Thus the number of rolling stock units on each train must match with the expected number of passengers. Deadheading of rolling stock units between depots and to and from maintenance facilities must be minimized. Furthermore, the number of crews needed to run the operations and to cover unforeseen demand must be minimized as well.

In more detail, an important objective of the operators in the disruption management process is to minimize the number of passengers affected by the disruption, and to minimize the inconvenience for the affected passengers. Indeed, small delays of trains are usually not considered as a bad service by the passengers, but large disruptions are. If passengers are too often confronted with large disruptions, which usually lead to long extensions of travel times and, even worse, to a lot of *uncertainty* about travel options and travel times, they may decide to switch to a different mode of transport. In relation to this, passenger operators usually prefer to return to the original timetable as soon as possible after a disruption. Indeed, the original timetable is recognizable for the passengers. Therefore, the original timetable provides a better service than a temporary ad hoc timetable during a disruption.

The passengers are the direct customers of the railway operators, and they are only indirect customers of the infrastructure manager. This may imply that the infrastructure manager has less knowledge of the expected passenger demand on each train and of the real-time passenger locations in the operations. The latter may prohibit a passenger focused dispatching, and may instead lead to a network capacity focused dispatching, i.e. dispatching focusing on supplying sufficient buffer times in the network to recover from disruptions.

Furthermore, each delay of a train may be attributed either to a railway operator or to the infrastructure manager, depending on the nature of the disruption. However, this creates a natural conflict between the organizations that may prohibit an effective communication and co-operation in the operations. The latter may be counter-productive for the operational performance of the railway system. Thus, although the infrastructure manager and the railway operators have the same general objective of providing railway services to the passengers of a high quality level, there are also conflicting elements in their objectives.

2.2.2 Actors

In railway disruption management, the actors are the dispatcher of the infrastructure manager and those of the railway operators. The major tasks to be carried out are *timetable adjustment*, *rolling stock rescheduling*, and *crew rescheduling*. Figure 2.2 shows how the responsibilities for the different elements are shared among the actors.

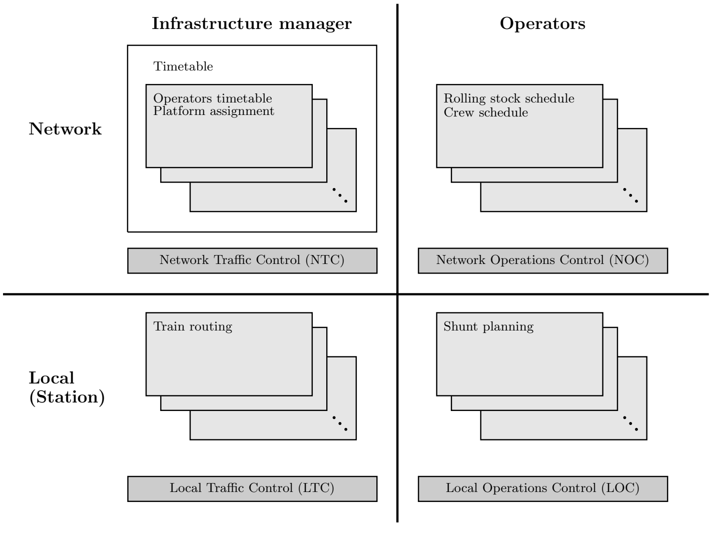


Figure 2.2: Schematic view of actors, timetables and resource schedules

The infrastructure manager controls and monitors all train movements in the railway network. *Network Traffic Control* (NTC) covers all tasks corresponding to the synchronization of the timetables of the different operators. NTC has to manage overtaking, rerouting, short turning, or canceling trains in order to prevent them from queuing up. The latter is a permanent threat at the basically one-dimensional railway infrastructure. Queuing up of trains immediately leads to extensions of travel times.

On a local level, the process is managed by the *Local Traffic Control* (LTC). For example, LTC is responsible for routing trains through railway stations and for platform assignments. Safety is ensured by headways and automatic track occupancy detection systems.

The *Network Operations Control* (NOC) of each passenger operator keeps track of the operations of the operator on a network level. The dispatchers of NOC are acting as decision makers for the operator in the disruption management process. Depending on the size of the operator, there are one or more dispatchers for rolling stock and crew, respectively. These dispatchers monitor and modify the rolling stock and crew movements. NOC dispatchers are the counterparts of the dispatchers of NTC.

Dispatchers of the *Local Operations Control* (LOC) of the railway operators are responsible for coordinating several local activities at the stations, such as shunting processes. They support NOC by evaluating whether changes to the rolling stock schedules can be implemented locally.

Train drivers and *conductors* are also important elements in the disruption management process. They are usually the first ones who are confronted with trains or passengers affected by a disruption. If train drivers and conductors work on different lines, they may carry a delay from one line to another.

In order to avoid this situation, the crew dispatchers may have to modify several duties. Besides making the decisions, the dispatchers also have to instruct and sometimes to convince the crew members to carry out the modifications, see Section 2.5.

2.2.3 Processes

NTC dispatchers constantly monitor the operations and have to decide if an actual situation is a disruption or will lead to a disruption in the near future. When this is the case, they start the disruption management process. Within this process, the original timetable may need to be changed. This is done by carrying out a dispatching plan.

Figure 2.3 displays the information flows between the different actors in this process.

First, NTC determines all trains that are affected by the disruption. NOC of the corresponding operators must then be informed about the disruption and its direct consequences. In the next step, the dispatchers have to find out to which extent it is still possible to run traffic on the involved route. Some pre-defined emergency scenarios give an indication about which trains should be overtaken, rerouted, short turned, or canceled. Using this information, an initial dispatching plan can be constructed. This dispatching plan must be evaluated by LTC. Almost simultaneously, the proposed dispatching plan is communicated to NOC of the operators. A complicating factor is the uncertainty about the duration of the disruption, for example NTC can only *estimate* how long it will take to repair a broken switch or signal.

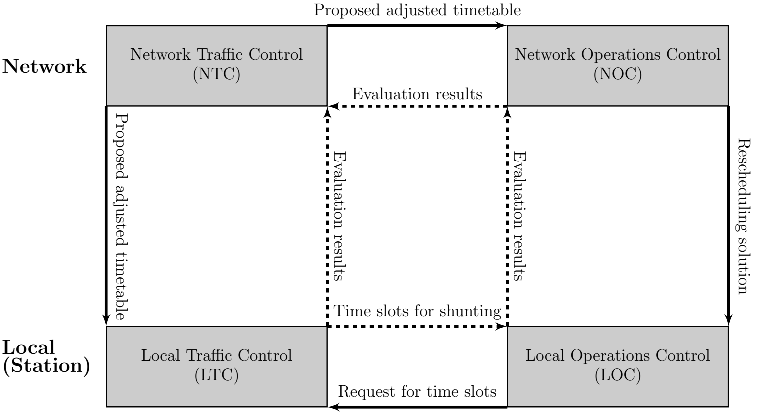


Figure 2.3: Information flow during the dispatching plan development

The dispatching plan may correspond to changes in the planned operations of several operators. As a whole, these changes are compatible with respect to the safety regulations. However, for the operators it may be impossible to operate the dispatching plan due to their resource schedules for rolling stock or crew. Therefore, the decision about the dispatching plan is taken in consultation between the infrastructure manager and the operators.

Hence, NOC dispatchers have to check whether it is possible for them to operate the proposed dispatching plan. In particular, they have to check whether they can adapt their resource schedules to the proposed dispatching plan. Furthermore, LOC has to verify that the modified timetable and the adapted resource schedules can be carried out locally. Because of the combinatorial nature of the resource schedules and the limited time available, not all rescheduling options can be evaluated. The rescheduling solutions represent a trade-off between the available time and the quality of the solution. The most important aspect is to find resource schedules that are feasible with respect to the proposed dispatching plan.

This evaluation procedure can basically have three different outcomes. First, NOC and LOC may find a rescheduling solution to the proposed dispatching plan where no additional cancellations or delays are needed. Second, they may find an initial solution, but trains have to be canceled in a second stage because rolling stock and/or crews are unavailable. A cancellation of a train has, however, a strong negative impact on the service level. Finally, NOC may come up with a request for changes to the proposed dispatching plan if this enables them to construct a much better solution.

Of course, not only one but several operators may ask for changes in the proposed dispatching plan. When these requests are conflicting, it is the responsibility of NTC to make a fair decision. This may involve another iteration of proposal and evaluation between NTC and the operators.

After the final decision about the dispatching plan has been taken by NTC, it is communicated to LTC and to the operators. LTC has to implement the new train routes and to change platform assignments. NOC has to inform the train drivers and conductors whose duties have been changed. LOC has to generate new shunting plans. LOC communicates directly with LTC to ask for time slots for shunting movements in the station area.

Furthermore, passengers need to be informed in trains, at stations, and via Internet and teletext about the changes in the timetable and alternative travel routes.

2.2.4 Organizational issues

The description in Section 2.2.2 of the actors in the disruption management process is a functional description, and not an organizational. For example, it suggests that all dispatchers of each of the mentioned actors are located in the same office. However, this need not be the case.

The Netherlands have been split up into 4 regions, and each region has its own NTC office and its own NOC office of NS. Moreover, there is a central NOC office of NS for coordinating the rolling stock rescheduling process. Similarly, there are 13 LTC offices and 13 LOC offices of NS. Obviously, this organizational split leads to a lot of additional communication within NTC and within NOC, which is counter-productive in the disruption management process. Therefore, it is discussed how to redesign the responsibilities of the NTC and the NOC offices. Moreover, it is investigated how the separation between the infrastructure manager and the operators can be reduced.

2.3 Timetable adjustments

2.3.1 Problem description

NTC has the overall responsibility of the railway operations and coordinates the disruption management process. When a disruption is recorded, NTC evaluates its effect and, if it is considered as severe, NTC tries to reschedule the timetable events affected by the disruption.

The severeness of a disruption is not easily assessed. It is described as a combination of how much time will pass until the operations are according to plan again and how many trains will be affected. The number of passengers that get delayed because of a disruption also contributes to its degree of severeness. Finally, it makes a large difference to the severeness whether the time intervals between trains on the same track (headways) are small or large. The effect caused by a blockage will be less on sections of the network with much time between consecutive trains than on sections with little time between the trains.

Timetables are constructed with included buffer time. Therefore, a timetable is able to absorb some disruptions. Buffer times are included in the dwell times, the running times, and the headways. When a disruption occurs, the buffer times in the timetable are used to gain time whenever possible. Thus they enable recovery from a disruption.

In general one can distinguish between disruptions with low and high impact on the timetable. Low level impact disruptions are those where recovery to the originally planned timetable is possible by using so-called dispatching rules. High level impact disruptions are those where recovery in this way is not possible, for example, if a complete blockage occurs at some part of the network. In such a case, more significant recovery measures are needed. These measures are presented in Section 2.3.2. Chapters 3–6 of this thesis will focus on these kind of disruptions. For a discussion of dispatching rules for low level disruptions we refer the interested reader to Jespersen-Groth et al. (2009).

A survey of optimization models for railway related problems is given by Cordeau et al. (1998). This survey describes various optimization models developed for railway problems. One of the described problems is the Train Dispatching Problem (TDP). TDP is the problem of minimizing delays by scheduling arrivals at bottlenecks of the railway network and overtakings, thereby taking into consideration operational costs. The velocity of trains is included in TDP as a decision variable (see D'Ariano (2008)). The paper of D'Ariano et al. (2007) describes how conflicts caused by timetable perturbations can be resolved in real-time.

Recently, a survey of algorithms and models for railway traffic scheduling and dispatching was given by Törnquist (2006). The problems mentioned are subdivided into tactical and operational scheduling and rescheduling. Of specific interest is rescheduling of trains, which focuses on the replanning of an existing timetable when a disruption has taken place.

2.3.2 Larger disruptions

For high impact disruptions, a set of emergency scenarios may exist, e.g. when tracks in one or both directions are completely blocked. These emergency scenarios describe for each section in the network and each direction an alternative timetable.

The immediate reaction to a high impact disruption is to apply an appropriate emergency scenario. On heavily utilized networks, the headways are so tight that the system will queue up immediately if no adequate measures are taken after a high impact disruption has occurred. Therefore, almost all railway traffic is canceled around the disrupted area. Trains may be turned around as closely as possible to this location. Otherwise, trains may be rerouted, but this requires sufficient capacity on the detour route. Finally, some lines may be canceled completely. Note that in practice the transformation from the planned timetable to the emergency scenario and back may involve some intermediate steps. E.g. by first not turning the trains anymore and then by restarting the service on the canceled train lines. Jespersen-Groth et al. (2006) present a model for calculating the order in which train lines should be restarted minimizing the time of the latest restart and taking the rolling stock inventories at the depots into account.

Example 2.1
As an example, consider a situation in which the tracks in both directions between stations Hoogeveen (Hgv) and Beilen (Bl) (see Figure 2.4) are blocked from 7:10 to 10:10. Three train lines use this route each with a frequency of once per hour: The earlier mentioned 500-line (intercity) between The Hague (Gvc) and Groningen, the 700-line (intercity) between Schiphol and Groningen, and the 9100-line (regional) from Zwolle to Groningen.

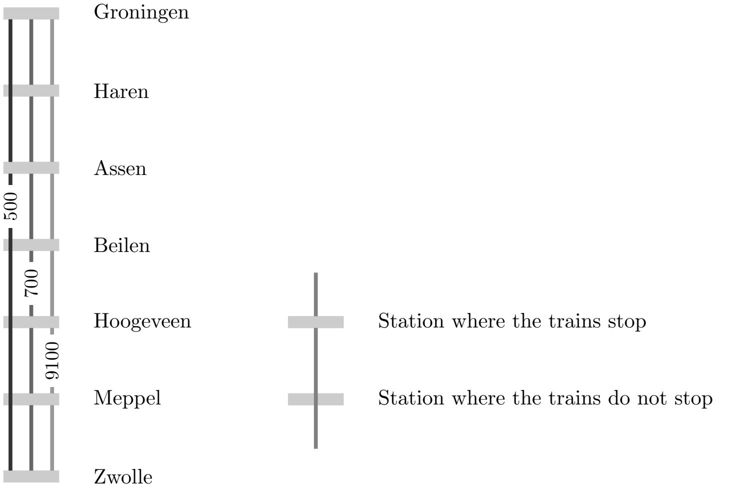


Figure 2.4: The train lines operated between Groningen (Gn) and Zwolle (Zl).

According to the emergency scenarios, the trains of the 500-line coming from Groningen are turned around in Assen (Asn) and the trains from The Hague are turned around in Hoogeveen, respectively. The same pattern is applied to the 700-line. The regional trains of the 9100-line from Zwolle are turned around in Meppel (Mp) and the trains from Groningen are turned around in Beilen, respectively. The resulting timetable and new *turns for the rolling stock are shown in Figure 2.5. Since there is no convenient alternative to go from Zwolle to Groningen by train during the time the route is blocked, bus-services between Beilen and Hoogeveen will be launched.*

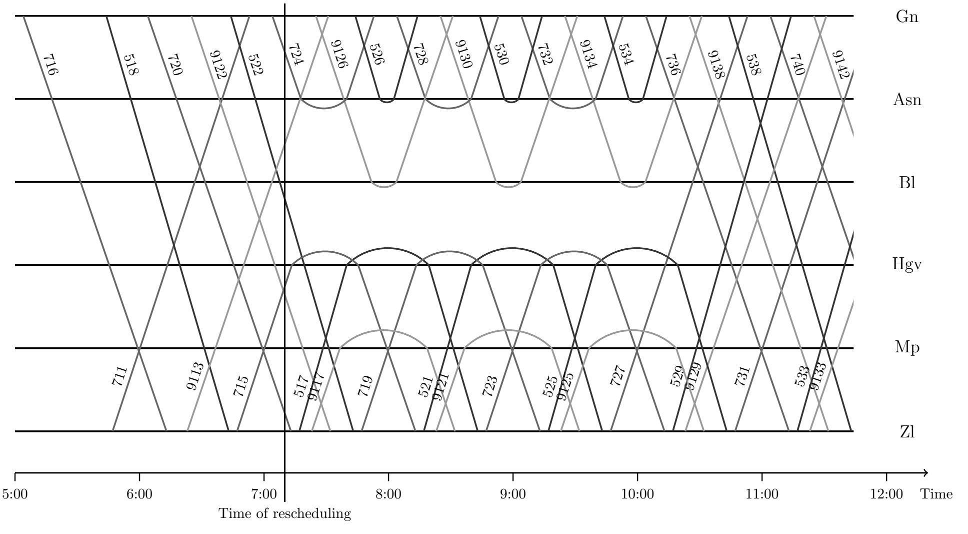


Figure 2.5: The adapted timetable between Groningen (Gn) and Zwolle (Zl).

2.4 Rolling stock rescheduling

2.4.1 Problem description

This section describes rolling stock rescheduling in a disrupted situation. Here the assumption is that, whenever this is necessary, the timetable has already been adjusted to the disrupted situation. The main goal is to decide how the rolling stock schedules can be adjusted to this new timetable at reasonable cost and with a minimum amount of passenger inconvenience.

The most characteristic feature of rolling stock is that it is bound to the tracks: rolling stock units cannot overtake one another, except at locations with parallel pairs of tracks. A broken rolling stock unit may entirely block the traffic – actually, this is a frequent cause of disruptions. Moreover, the operational rules of rolling stock units are largely determined by the shunting possibilities at the stations. Unfortunately, shunting is a challenging problem in itself, even for a medium-size station. Therefore, NOC must constantly keep contact with LOC and check whether or not their intended measures can be implemented in practice. The modifications may be impossible due to lack of shunting drivers or infrastructure capacity.

Timetable services must be provided with rolling stock of any type. Also, the assignment must fulfill some elementary requirements. For example, the rolling stock type must be compatible, and each train should not be longer than the shortest platform on its route. Especially in a disrupted situation, shunting operations are reduced as much as possible. In particular, shunting operations at locations or points in time where they do not occur in the original schedules are highly undesirable.

Railway operators usually keep a certain amount of rolling stock on stand-by. These units can be used only in case of disruptions. Moreover, many of the rolling stock units are idle between the peak hours, since the rolling stock capacity is usually too large for off-peak hours. If a disruption takes place during off-peak hours, these idle units can act as stand-by units.

In case of a disruption, the first dispatching task is to assign the available rolling stock units to train tasks. These decisions are taken under high time pressure, often guided by the emergency scenarios which tell how the trains have to turn. E.g. the emergency scenario for Example 2.1 says that the rolling stock from train 715 from Zwolle to Hoogeveen should go back to Zwolle as train 724. Whenever there is room for changes, the planners try to cover the seat demand as well as possible. In some cases, however, they are forced to cancel trains due to lacking rolling stock or to have a train that offers too little capacity.

After a disruption, it is preferable for the rolling stock schedules to return to the originally planned schedules as quickly as possible, since the feasibility of the originally planned schedules has been checked in detail. As a consequence of all these measures, the rolling stock units will not finish their daily duties at the locations where they were planned prior to the disruption. This is not a problem if two units of the same type get switched: rolling stock units of the same type can usually take each other's duty for the rest of the day. More likely, however, the numbers of units per type ending up in the evening at a station differ from the numbers of units per type that were planned to end up there. Thus, unless expensive deadheading trips are used, the traffic on the next day is influenced by the disruption. Modifications of the schedules for the busy peak hours of the next morning are highly undesirable. Therefore additional measures are taken such that at night the actual rolling stock balance is as close as possible to the planned balance.

A further important element in rolling stock rescheduling is maintenance of rolling stock. Train units need preventive maintenance after a certain number of kilometers or days, roughly once a month. Due to efficiency reasons, units are usually in service just until they reach a certain maintenance limit. Units that are close to this limit and have to undergo a maintenance check in the forthcoming couple of days are monitored permanently. The latter is particularly important during and after a disruption which may have distracted these units from their planned route towards a maintenance facility. NOC has to make sure that these units reach a maintenance facility in time. Usually, only a small number of rolling stock units is involved in planned maintenance routings. Other units of a given type are interchangeable, both in the planning stage and in the operations.

2.4.2 Current practice at NS

NS operates a dense railway system. This basically allows for many alternative rolling stock schedules through exchanges of train units. However, usually trains have short turn-around times, which rules out complex shunting operations at end points. Also, the shunting capacity (shunting area and crews) of stations is often a bottleneck. NS operates rolling stock units of several types. Moreover, a train may contain units of different types. In this case, the order of the train units in the train is important. On one hand, this allows adjusting the rolling stock types well to the passenger demand. In case of disruptions, however, the dispatchers have the additional task of monitoring and rebalancing exchanged rolling stock types.

NS uses a sophisticated computer system for rolling stock management, which provides automated tracking and tracing of the real-time positioning of individual units. The system, however, lacks algorithmic decision support tools; nearly all decisions have to be taken and to be fed to the system manually. As a consequence of the lack of decision support, the dispatchers focus on the immediately forthcoming time period only. Moreover, planning for a longer period of time may be a waste of effort since new disruptions can occur. Dispatchers identify possible conflicts, and handle them in order of urgency.

2.4.3 New developments

Compared to medium-term planning, there is a very scarce literature on real-time rolling stock rescheduling. In the recent years intensive research has been conducted to develop methods for the real-time problems as well.

Budai et al. (2009) study the Rolling Stock Balancing Problem. It is assumed that the timetable and a feasible rolling stock schedule are given. Moreover, the target rolling stock balance is given. This target is equal to the number of units per type that were originally supposed to arrive at the stations at the end of the planning horizon. The Rolling Stock Balancing Problem aims at modifying the input schedule in such a way that the realized end-of-day balance is as close to the target as possible.

Although the problem was first studied for the operational planning phase, it is also relevant in real-time rescheduling after a disruption when all immediate conflicts have been resolved (that is, there is a feasible schedule) but the realized end-of-day rolling stock balance differs from the target balance.

Budai et al. (2009) prove that an off-balance of a single train unit leads to an NP-hard optimization problem. Also, two heuristic algorithms are developed and compared to exact optimization methods. The computational results on real-life problem instances of NS indicate that the heuristic algorithms provide solutions of promising quality very quickly, within a few seconds.

Another track of research aims at applying an existing rolling stock circulation model of Fioole et al. (2006) for real-time planning. The basic model of Fioole et al. (2006) is a very flexible linear integer programming model that has been used by NS since 2004 for medium term planning. However, it cannot deal with uncertainties of the input data, and solving it by commercial MIP software can take several hours. Therefore Nielsen (2008) developed a rolling horizon based solution approach for dealing with real-time rescheduling problems of NS. The cornerstone of the method is the extension of the model of Fioole et al. (2006).

The main idea is to consider at any moment the forthcoming, say, 3 hours only. The extended MIP model is solved for this restricted time horizon based on the latest forecasts on the duration of the disruption. This optimization can indeed be performed in a few seconds. An hour later, or whenever new, relevant information arrives, the model is solved again for the forthcoming hours. This process is repeated until the end of the day. The algorithm is highly inspired by the current rolling stock disruption management.

The algorithm of Nielsen (2008) deals with three objective criteria: (*i*) cancellation of trips; (*ii*) deviation from the originally planned shunting process; and (*iii*) deviation from the originally planned end-of-day balance.

Criterion (*i*) is related to keeping a high service quality. Criterion (*ii*) enhances the chance that the found solution can be implemented in practice. Indeed, new, non-planned shunting operations can turn out to be impossible due to lacking shunting capacity. Finally, criterion (*iii*) tries to reduce the disruption's consequences for the next day.

While the first two criteria are easily incorporated in a rolling horizon framework, the deviation from the target rolling stock balance is conceptually more difficult: The end of the day is not visible until the very last iteration. Nielsen (2008) proposes the following heuristic way to cope with this issue. Consider a single iteration of the rolling horizon algorithm where the current horizon is from 12:00 till 15:00. Then one computes what should be the rolling stock balance at 15:00 according to the original, undisrupted schedule, and this is defined to be the target balance in the current iteration. Clearly, this guidance is inaccurate in the middle of the day, but it gets more and more precise as the end of the rolling horizon approaches the end of the day. Accordingly, the relative importance of criterion (*iii*) increases as the rolling horizon proceeds.

Nielsen (2008) reports computational results on several realistic problem instances of NS. These include disruptions on the so-called “Noord-Oost” case, a particularly complex rolling stock scheduling instance. The rolling horizon based algorithm found solutions with very little deviation from the undisrupted schedule, both in terms of shunting and in terms of rolling stock balance. On-going research focuses on making the algorithm fully comply with the restrictions of railway practice. This includes fine-tuning the algorithm as well as some extensions such as dealing with maintenance of rolling stock units.

Jespersen-Groth et al. (2008) and Jespersen-Groth (2009) propose to decompose the rolling stock rescheduling problem into three steps. In the first step a *position model* is solved that determines a suitable assignment of rolling stock compositions to train tasks. In the second, optional, step a *sequence model* assigns train units to sequences of train tasks that require a single unit. Finally, in the *routing model* train units are assigned to paths of train tasks. If the sequence model was used, the assignments found in the sequence model are fixed in the routing model. Experiments were conducted with data from DSB S-tog, the operator of local trains in the greater Copenhagen area. The results show that using the sequence model computation times for the routing model decrease dramatically. Good quality solutions for instances of realistic problem sizes could be found within minutes.

2.5 Crew rescheduling

2.5.1 Problem description

Recall that the recovery of the timetable, the rolling stock schedule, and the crew schedule is usually done in a sequential fashion. For an estimated duration of the disruption, a modified rolling stock schedule has been determined for a modified timetable. Both are input for the *operational crew rescheduling problem* (OCRSP), in which the crew schedule needs to be modified in order to have a driver and an appropriate number of conductors for each task of the modified timetable. Tasks can be either passenger train movements, empty train movements, or shunting activities. From this point on, we will focus on train drivers. The problem of rescheduling conductors is, however, quite similar.

All operations that need to be performed by a driver are represented by a task, where a task is an elementary sequence of activities starting and ending at a relief point. Relief points are the subset of all stations where drivers can transfer from one rolling stock unit to another one. The trains of the 500-line (see Figure 2.1), for example, are split up into the following tasks: Groningen–Zwolle, Zwolle–Amersfoort, Amersfoort–Utrecht, Utrecht–Gouda, and Gouda–The Hague. Within this thesis, 12345/i will refer to the i-th task of train “12345”. E.g. 522/b is the task Zwolle–Amersfoort of train 522.

On the day of operations the crew schedule is given by the original duties, each assigned to a driver. These original duties are either active duties, in the sense that they are a sequence of tasks, or reserve duties, meaning that a driver is on standby at a major station for a given time period and tasks can be assigned to the driver. All duties start and end at the same crew base, where crew bases are a subset of the relief points. For repositioning from one relief point to another, duties can also contain taxi trips or deadhead tasks. The latter means that a driver is traveling as a passenger on a task.

Due to a disruption on the day of operations the timetable is modified according to the estimated duration of the disruption. This could mean that the new timetable and the driver duties have become incompatible. In this case it is necessary to reschedule the drivers. Because NS operates very few night trains, for rescheduling it is generally sufficient to consider a crew schedule of a single day.

Given the point in time of rescheduling, for every unfinished original duty we need to find a replacement duty. A replacement duty is composed of all tasks of the associated original duty that started before the time of rescheduling, and a feasible completion. Feasible completions are (possibly empty) sequences of tasks such that the replacement duty satisfies the following rules.

- The replacement duty needs to start and end at the same crew base associated with the original duty. Furthermore, a replacement duty may end earlier or at the planned time. In addition, it is allowed to end up to 60 minutes later than the planned end time of the original duty.

- If in a replacement duty two tasks are performed after each other on different rolling stock units, then a minimum connection time between the two tasks needs to be respected. This connection time is less during rescheduling than in the planning phase.

- Every replacement duty longer than 5 1/2 hours must contain a meal break of at least 30 minutes. Meal breaks are possible only at relief points that have a canteen. Moreover, the working time before and after the break is not allowed to exceed 5 1/2 hours.

- Every driver is licensed to drive on certain parts of the railway network. Moreover, he is licensed to drive certain rolling stock types. These two attributes determine the set of tasks that can be performed by a replacement duty.

If an original duty is not affected by a disruption, one feasible completion is to follow the sequence of tasks as in the original duty.

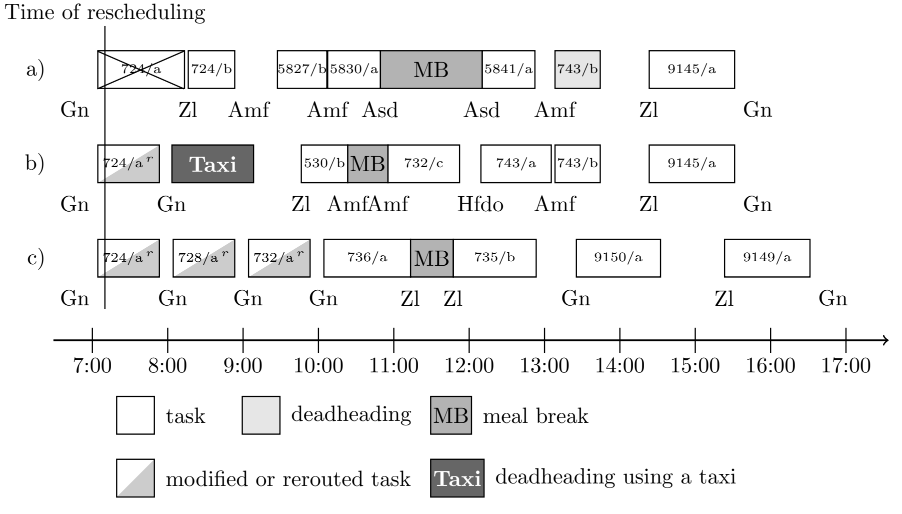


Figure 2.6: Examples of feasible completions for an affected original duty from crew base Groningen (Gn).

Example 2.2
Figure 2.6.a shows how original duty Gn 7 *from crew base Groningen was planned. At 7:10, when the rescheduling takes place, the duty is performing task 724/a belonging to the 700-line. Since the route is blocked south of Beilen, the train is turned in Assen, from where it goes back as train 715 to Groningen (cf. Figure 2.5). This means that task 724/a is replaced by task 724/a*^{r} *which starts and ends in Groningen. This means that after performing his first task of the day the driver will be back in Groningen. It is not possible to get to Zwolle in time to perform task 724/b which was supposed to be the next task. Hence duty Gn* 7 *is not feasible anymore. Now we will show two examples of feasible completions of Gn* 7*. Given 7:10 as the time of rescheduling, Gn* 7 *must first complete task 724/a*^{r}*. Then it is possible to deviate from the plan. One possibility (Figure 2.6.b) would be to take a taxi from Groningen to Zwolle and drive task 530/b to Amersfoort. After a meal break (MB) in Amersfoort, task 732/c to Hoofddorp on the 700-line could be performed, followed by 743/a and 743/b also on the 700-line. Finally, the duty could finish with driving the regional train (9100-line) from Zwolle to Groningen (9145/a). Note, that in this feasible completion the last two tasks are the same as in the original duty, except that in the original duty the driver was deadheading as a passenger on task 743/b. Another possibility shown in Figure 2.6.c would be to continue driving rerouted tasks of the 700-line (728/a*^{r}*, 732/a*^{r}*) before going from Groningen to Zwolle*

(736/a), also on the 700-line. After a meal break in Zwolle the driver could perform task 735/b back to Groningen. Since it is allowed to end up to 60 minutes later, the duty could finish with driving two trains (9150/a, 9149/a) of the 9100-line to Zwolle and back.

Now we can formulate the OCRSP as follows. Given the modified timetable, the modified rolling stock circulations, and the planned crew schedule, find a new crew schedule that covers as many tasks as possible such that every original duty is assigned to one feasible completion. The objective of the OCRSP is a trade-off between different aspects, namely feasibility, operational costs, and robustness. We will now briefly discuss these aspects.

First of all, there is the feasibility aspect. It is not evident that all tasks can be covered by a solution. Given two solutions with different uncovered tasks, there may exist a preference for one of them, depending on the urgency and the expected numbers of passengers of the uncovered tasks. If a task cannot be covered, canceling it will lead to a feasible crew rescheduling solution. An additional cancellation, however, leads to more inconvenience for the passengers, which is against the general aim of disruption management. Moreover, such a cancellation has to be approved by the rolling stock dispatchers and the local planners, since it disturbs the rolling stock circulation. Because a cancellation is a change of the timetable, it has to be approved by NTC.

Operational costs are the second aspect in the objective. In the railway context, the crew payments are often based on fixed salaries. Nevertheless, some parts of a rescheduling solution influence the operational costs. Crew deadheading on trains can be considered to have no costs other than time, whereas using other transport options (e.g. taxis) for repositioning and taking home stranded crews is not free. Also, operator specific compensations for overtime work due to modified duties need to be considered.

The third aspect in the objective is stability. Humans are involved in the implementation of every rescheduling solution and can cause its failure. A crew dispatcher may, for example, forget to call a driver and inform him about the modifications in his duty. Therefore, a solution is considered to be more stable if the number of modified duties is smaller.

2.5.2 Current practice at NS

A closely related problem is crew rescheduling in short term planning. This occurs for instance due to timetable changes based on maintenance work on tracks. The resulting crew schedule is called a *special plan*. For the construction of special plans additional rules have to be taken into account. If a special plan is made prior to 72 hours before the day of operation, duties may start and end up to 30 minutes earlier (respectively later) compared to the planned schedule. Within the last 72 hours before the day of operation duties may start earlier or end later only if this is accepted by the crew member.

NS nowadays use optimization software to construct special plans. The dedicated approach (Huisman (2007)) has been integrated into the CREWS planning system (Morgado and Martins (1998)). The algorithm relies on a combination of column generation and Lagrangian relaxation.

For crew rescheduling on the day of operation NS is not using a decision support system. The crew dispatchers use an interactive software system. This provides them with information about the actually planned duties, and enables them to store their duty modifications in the system. The system informs them about delays of trains and about modifications in the timetable and rolling stock schedules. The system also indicates time and location conflicts in the duties. Recovery options, however, have to be found manually without algorithmic support. In the manual procedure, conflicts are resolved one at a time in order of urgency.

As mentioned earlier several agreements exist about the way duties may be modified on the day of operations. However, if a dispatcher finds an option outside these rules he might ask the affected drivers if they are willing to accept the changes to their duties. Experiments were carried out to inform crew members automatically via SMS about duty modifications. However, communicating modifications via telephone is still the common practice.

2.6 Integrated Recovery

The integrated recovery approach has received little attention up till now. To the best of our knowledge Walker et al. (2005) is the only paper presenting a model that manipulates the timetable and the crew schedule at the same time. The objective is to simultaneously minimize the deviation of the new timetable from the original one, and the cost of the crew schedule. One part of the model represents the timetable adjustment, the other part corresponds to a set partitioning model for the crew schedules. Both parts are linked in order to get a compatible solution. It should be mentioned that the railway systems addressed in the research is of a relatively simple structure.

The benefits of such an approach compared to the sequential approach may, however, be large in terms of quality of service, and the field is expected to become an active research field in the future.

In Chapter 5 we present a model that allows retiming for some tasks. This can be viewed as a partial integration of timetabling and crew rescheduling. We only consider a limited number of retiming possibilities. This keeps the resulting model tractable such that we can compute near optimal solutions in a very short amount of computational time. We show that with our approach less tasks need to be canceled compared to classical crew rescheduling without retiming.

2.7 Conclusions

Railway operators pay much attention to improve their operational performance. One of the key issues is to limit the number of delays by reducing the knock-on effect of single disruptions. To achieve this goal, effective disruption management is required. In this chapter, we have explained the role of the different organizations and actors in the disruption management process. An important issue here is that next to the operator itself, the infrastructure manager plays a major role in the disruption management process. The different objectives of both organizations on one hand and difficult communication schemes on the other hand, complicates the disruption management process a lot.

After the description of disruption management, we have discussed the three subproblems arising in railway disruption management: timetable adjustment, and rolling stock and crew rescheduling. To adjust the timetable, several dispatching rules are applied in practice. Unfortunately, no optimization techniques are involved to solve this problem currently. For the rescheduling of rolling stock and crew some first attempts have been made in the literature to come up with OR models and solution techniques. Current developments can be divided into two major categories. The first category includes attempts to integrate the already developed approaches into decision support systems and to finally use these tools in practice. The second category consists of research into extending the current approaches. Possible extensions include (partial) integration of timetabling and resource rescheduling (see Chapter 5), the consideration of passenger flows, and the uncertainty about the duration of the disruption as discussed in Chapter 6.

Chapter 3

Column Generation with Dynamic Duty Selection for Railway Crew Rescheduling

3.1 Introduction

In Chapter 2 we presented the disruption management process at a passenger railway operator. In this chapter we will present a mathematical model and an innovative solution approach to solve the OCRSP, introduced in Section 2.5, one of the three challenging problems within railway disruption management. The OCRSP is challenging because decisions have to be made quickly, while one has to deal with a large number of crews. As mentioned in Section 1, the crew schedule of Netherlands Railways (NS) contains around 1,000 duties for drivers on an average workday.

The contribution of this chapter, which is based on Potthoff et al. (2010), is twofold. We propose a new algorithm that heuristically solves the problem for dynamically selected subsets of the duties. This is our first contribution and this idea can be applied to many other domains of rescheduling as well. The heuristic follows ideas from Huisman (2007) for crew rescheduling in short-term planning. Our second contribution lies in the fact that we test our methods on real-life data of NS and we show that we can find good solutions in a reasonable amount of time. As a result, the methods that we propose can lay the foundations for algorithmic decision support for dispatchers in the operations control centers of NS.

The remainder of this chapter is organized as follows. In Section 3.2 a literature overview on crew rescheduling is given. A mathematical formulation and the outline of our solution approach is presented in Section 3.3. In Section 3.4, we present a heuristic based on column generation and Lagrangian relaxation to solve the OCRSP for a fixed subset of duties. In Section 3.5, we discuss how we can dynamically adjust the subsets of duties. Computational results are reported in Section 3.6. We finish with some concluding remarks in Section 3.8.

3.2 Literature review

During the last decade crew rescheduling, also known as crew recovery, has received a lot of attention in the airline literature. The application of a crew rescheduling decision support system at Continental Airlines (Yu et al. (2003)) won the Franz Edelman award. Stojković et al. (1998) published the first results for a rescheduling model dealing with crew pairing and rostering simultaneously. They apply a column generation approach for a preselected subset of crews. Column generation in combination with core problems defined by a selection of crews and/or time windows has also been used by Lettovský et al. (2000); Nissen and Haase (2006); Medard and Sawhney (2007).

Stojković and Soumis (2001) propose to consider time windows in order to allow retiming of flights. The situation that a crew (e.g. pilots and flight attendants) assigned to a flight do not need to stay together for the whole rescheduling period is modeled by Stojković and Soumis (2005) and Abdelghany et al. (2004). Recently, Abdelghany et al. (2008) extend the work of Abdelghany et al. (2004) by integrating aircraft and crew rescheduling.

For a recent review of airline crew recovery we refer the interested reader to Clausen et al. (2010).

To the best of our knowledge, the first attempt to come up with an approach including the aspect of railway crew rescheduling was made by Walker et al. (2005). The paper presents a model for simultaneous railway timetable adjustment and crew rescheduling. A timetabling part where the departure of tasks can be chosen within time windows is linked to a crew scheduling part where generic driver shifts are chosen. Here a generic driver shift is a sequence of tasks that is feasible with respect to the start and end locations of consecutive tasks. Shift length and task (piece-of-work) sequencing constraints ensure that the departure times are chosen such that only the break rule may be violated in the selected shifts. Breaks are added into the shifts 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. Since the model size would explode for a network as operated by NS, their approach is not applicable to the Dutch situation.

Crew rescheduling within disruption management was subject to research projects at NS. Experience from short term planning has already made clear that it is not possible to consider all duties and tasks in the rescheduling problem due to too long computation times. Therefore all studies on rescheduling on the day of operations consider only a small part of the crew schedule, given by a subset of the duties and a time window.

One such tailored solution method to solve the crew rescheduling problem was developed by Rezanova and Ryan (2010) and Rezanova (2009). The problem is formulated as a set partitioning problem and possesses strong integer properties. The proposed solution approach is therefore a depth-first search in a branch-and-price tree. The LP-relaxation of the problem is solved with a column and constraint generation algorithm. The problem is first initialized with a very small disruption neighborhood, which contains only duties that cover delayed, canceled or re-routed tasks and is limited by a recovery period. As long as, while solving the LP-relaxation, constraints are uncovered, the disruption neighborhood is extended by either adding more duties to the problem or by extending the recovery period. The algorithm was tested on instances based on historical disruptions using real-life crew schedules from DSB S-tog. The obtained results are very good in terms of solution quality as well as in terms of computation time. In order to deal with new information becoming available, the author(s) propose to use the crew rescheduling algorithm in a rolling time horizon approach similar to the one proposed by Nielsen (2008) for rolling stock rescheduling. However, tests about the effects of changing information are not reported.

Finally, there are some experiments at NS with multi-agent technology. In this approach each driver is represented by a driver-agent. If due to the disruption a driver-agent can no longer perform a certain task, this driver-agent starts a negotiation process with other driver-agents to transfer the task to another driver-agent. For more details, we refer to Abbink et al. (2009) and Mobach et al. (2009).

3.3 Mathematical model and solution approach

In the remainder of this chapter we use the following notation.

- S: Set of stations (in our case limited to relief points).
- D: Set of crew bases.
- N: Set of tasks which have not started at the time of rescheduling, where for every $i \in N$ we have:
  - $s_i^{\text{dep}}, t_i^{\text{dep}}$: Departure station and time.
  - $s_i^{\text{arr}}, t_i^{\text{arr}}$: Arrival station and time.
- $\Delta = \Delta_A \cup \Delta_R$: Set of unfinished original duties, where Δ_A are active and Δ_R are reserve duties, respectively. Moreover, for every $\delta \in \Delta$ we have:

- cs_δ: The station where the original duty δ is at the time of rescheduling or the arrival station of the task performed by the driver at the time of rescheduling.
- b_δ: The crew base where the original duty δ starts and ends.

- K^δ: Set of all feasible completions for original duty $\delta \in \Delta$. For every feasible completion $k \in K^\delta$ we have:
  - c_k^δ: Cost of feasible completion k for original duty δ. The cost of a feasible completion is zero if the duty is not modified. Otherwise, the cost is the sum of the cost for changing a duty, the cost for taxis, and the penalties for short connection times and overtime.
  - a_{ik}^δ: Binary parameter indicating if task i is covered by feasible completion k or not.
- f_i: Cost for canceling task i.

We can formulate the OCRSP using binary variables x_k^δ corresponding to the selection of the feasible completions of duty δ and binary variables z_i indicating if task i is canceled (1) or not (0).

$$\min \quad \sum_{\delta\in\Delta}\sum_{k\in K^\delta} c_k^\delta x_k^\delta + \sum_{i\in N} f_i z_i \tag{3.1}$$

$$\text{s.t.} \quad \sum_{\delta\in\Delta}\sum_{k\in K^\delta} a_{ik}^\delta x_k^\delta + z_i \geq 1 \qquad \forall i \in N \tag{3.2}$$

$$\sum_{k\in K^\delta} x_k^\delta = 1 \qquad \forall \delta \in \Delta \tag{3.3}$$

$$x_k^\delta, z_i \in \{0,1\} \qquad \forall \delta \in \Delta, \forall k \in K^\delta, \forall i \in N \tag{3.4}$$

In the above model, constraints (3.2) make sure that every task is either covered by a feasible completion or canceled. Furthermore, constraints (3.3) ensure that every original duty is assigned to exactly one feasible completion.

Note that, in the above model, deadheading can occur in two ways. Firstly, a feasible completion can explicitly use deadheading on tasks, e.g., if the driver of the original duty does not have the required route knowledge. In this case, the corresponding a_{ik}^δ coefficient is equal to 0. Secondly, a task can be overcovered in the solution of the model, then one of the drivers has to perform the driving, the other(s) deadhead on this task, but all coefficients a_{ik}^δ are equal to 1.

Recall from Chapter 1 that we can have about 1,000 original duties, of which about 90 are reserve duties, in (3.3). Moreover, the number of set covering constraints in (3.2) can be up to 10,000. The number of feasible completions for an original duty can range from only a few if the duty is almost finished when we reschedule, to millions if the duty has not started or has just started. If rescheduling is done on the day of operations, the emphasis is on obtaining the best possible solution within a couple of minutes of computation time rather than solving (3.1)–(3.4) to optimality.

Moreover, a local disruption like the one described in Example 2.1 affects only a limited number of original duties. Because we want to stay close to the planned schedule, it seems highly unlikely that an original duty covering tasks only in another part of the country will be modified in an optimal solution of (3.1)–(3.4) in this case. Therefore, it seems reasonable to consider a core problem containing only a subset of the original duties and tasks.

The advantage of a core problem is its reduced size, which will lead to shorter computation times. A drawback is that the solution quality might depend on the choice of the core problem. In particular, one might be able to reduce the number of canceled tasks by increasing the size of the core problem.

For the case of airline crew rescheduling this has been observed by Lettovský et al. (2000) and Nissen and Haase (2006). In both papers the core problems are generated using a set of parameters, which makes it possible for the dispatcher to solve the problem again with a larger core problem, if he is unsatisfied with the quality of the solution obtained so far. The drawback of this scheme is that computation times increase rapidly with the size of the core problems.

In order to overcome this drawback, we propose a different way, illustrated in Figure 3.1, of exploring promising parts of the solution space of (3.1)–(3.4). As in the other approaches, we start with an initial core problem. This initial core problem is defined such that it has a high probability of containing a good solution and is of a size that allows us to explore it within a small amount of time. If tasks need to be canceled in the solution obtained for the initial core problem, we try to cover them by exploring a neighborhood for each uncovered task in turn. We use a heuristic based on column generation and Lagrangian relaxation to explore the core problems. This heuristic is described in Section 3.4. In Section 3.5, we discuss how we define the core problems.

Starting with an initial feasible solution and trying to improve it iteratively by fixing a part of the solution and reoptimizing the remaining part has been proposed for several combinatorial problems. Examples are the heuristic of Caprara et al. (1999) for the set covering problem, the large neighborhood search (LNS) heuristics of Ropke and Pisinger (2006) and Prescott-Gagnon et al. (2009) for the vehicle routing problem with time windows, and of Pepin et al. (2009) for the multiple depot vehicle scheduling problem. The latter two papers have in common with this chapter that they use heuristic column generation for neighborhood exploration.

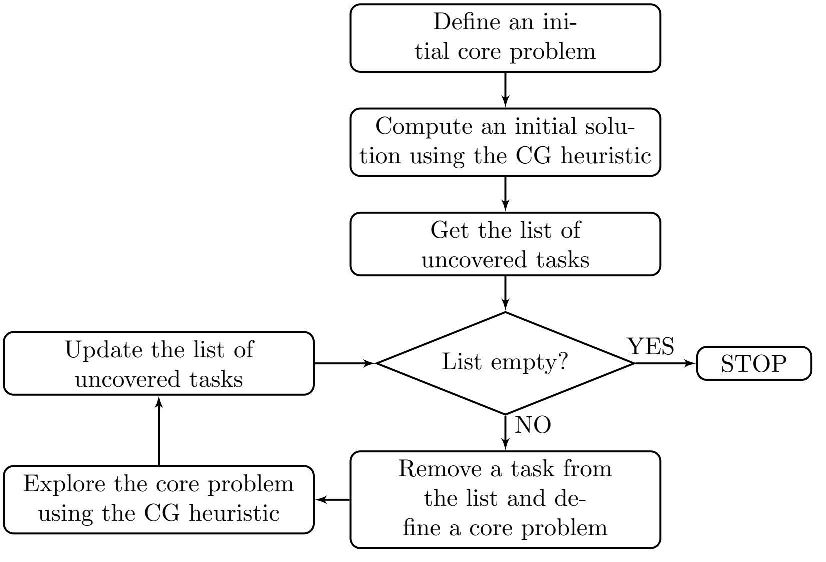


Figure 3.1: Overview of the algorithm.

3.4 Exploring the core problems

We compute a lower bound and near optimal solutions for the core problems with a column generation based heuristic. We will first describe the building blocks of our heuristic, before we present it in Section 3.4.3.

3.4.1 Combining column generation and Lagrangian relaxation

A core problem is given by a subset $\bar{\Delta}$ of the original duties and a subset $\bar{N}$ of the tasks. Given $\bar{\Delta}$, $\bar{N}$ contains the tasks that are covered by at least one $\delta \in \bar{\Delta}$ plus the tasks uncovered in the current solution. More formally a core problem reads:

$$\min \quad \sum_{\delta \in \bar{\Delta}} \sum_{k \in \bar{K}^\delta} c_k^\delta x_k^\delta + \sum_{i \in \bar{N}} f_i z_i \tag{3.5}$$

$$\text{s.t.} \quad \sum_{\delta \in \bar{\Delta}} \sum_{k \in \bar{K}^\delta} a_{ik}^\delta x_k^\delta + z_i \geq 1 \qquad \forall i \in \bar{N} \tag{3.6}$$

$$\sum_{k \in \bar{K}^\delta} x_k^\delta = 1 \qquad \forall \delta \in \bar{\Delta} \tag{3.7}$$

$$x_k^\delta, z_i \in \{0,1\} \qquad \forall \delta \in \bar{\Delta}, \forall k \in \bar{K}^\delta, \forall i \in \bar{N} \tag{3.8}$$

where $\bar{K}^\delta \subseteq K^\delta$. This subset contains all feasible completions that are represented by a path in graph $\bar{G}^\delta$, to be discussed in Section 3.4.2. To find good feasible solutions to (3.5) subject to (3.6)–(3.8) fast, we use a Lagrangian heuristic similar to the one proposed by Huisman (2007). Therefore, we relax the covering constraints (3.6) in a Lagrangian way introducing nonnegative Lagrangian multipliers $\lambda_i, i \in \bar{N}$. The Lagrangian subproblem then becomes:

$$\begin{aligned}\Theta(\lambda) = \min &\sum_{\delta\in\bar{\Delta}}\sum_{k\in\bar{K}^\delta} c_k^\delta x_k^\delta + \sum_{i\in\bar{N}} f_i z_i + \sum_{i\in\bar{N}} \lambda_i(1 - \sum_{\delta\in\bar{\Delta}}\sum_{k\in\bar{K}^\delta} a_{ik}^\delta x_k^\delta - z_i)\\ \text{s.t.}\quad &(3.7) \text{ and } (3.8)\end{aligned}$$

which can be rewritten as

$$\begin{aligned}\Theta(\lambda) = \min &\sum_{i\in\bar{N}} \lambda_i + \sum_{\delta\in\bar{\Delta}}\sum_{k\in\bar{K}^\delta} (c_k^\delta - \sum_{i\in\bar{N}} \lambda_i a_{ik}^\delta) x_k^\delta + \sum_{i\in\bar{N}} (f_i - \lambda_i) z_i \\ \text{s.t.}\quad &(3.7) \text{ and } (3.8)\end{aligned} \tag{3.9}$$

The Lagrangian subproblem is separable and therefore its optimal solution can be found with the following procedure. In order to not violate constraints (3.7) we set $x_k^\delta = 1$ for one $k \in \arg\min\{\bar{c}_k^\delta(\lambda) : k \in \bar{K}^\delta\}$ for each $\delta \in \bar{\Delta}$, where $\bar{c}_k^\delta(\lambda) := c_k^\delta - \sum_{i\in\bar{N}} \lambda_i a_{ik}^\delta$ is the reduced cost of feasible completion k. All other x_k^δ variables are set to 0. Furthermore, for each $i \in \bar{N}$, we set $z_i = 1$ if $f_i - \lambda_i < 0$ and $z_i = 0$ otherwise.

Now the Lagrangian dual problem is to find

$$\Theta^* = \max \Theta(\lambda), \quad \lambda \geq 0$$

Because the number of feasible completions for every driver can still be huge we combine Lagrangian relaxation with column generation. We assume that the reader is familiar with the general ideas of column generation, for a short introduction into this topic we refer to Section A.2.1. We thus consider a *restricted master problem* (*RMP*) of (3.7)–(3.9) containing only a subset of the x_k^δ variables. In the n^{th} column generation iteration the x_k^δ variables in the RMP are given by $\cup_{\delta\in\bar{\Delta}}\{x_k^\delta : k \in \bar{K}_n^\delta\}$, where $\bar{K}_n^\delta \subseteq \bar{K}^\delta$ is a subset of feasible completions. Let Θ_n^* be the optimal value of the associated Lagrangian dual problem. For every RMP we use subgradient optimization (see e.g. Fisher (1981)) to approximate Θ_n^*. Let λ_n^* be the corresponding multiplier vector. We solve a pricing problem for every original duty $\delta \in \bar{\Delta}$ to check if Θ_n^* is a good approximation of Θ^*. Otherwise, we need to add feasible completions to the RMP in order to potentially improve on Θ_n^*. The pricing problems are modeled as resource constrained shortest path problems in dedicated graphs as described later in Section 3.4.2. Let $r_n^\delta := \min\{\bar{c}_k^\delta(\lambda_n^*) : k \in \bar{K}_n^\delta\}$ be the smallest Lagrangian reduced cost of the already generated feasible completions for original duty δ and $p_n^\delta := \min\{\bar{c}_k^\delta(\lambda_n^*) : k \in \bar{K}^\delta\}$ the optimal value of the pricing problem for δ. Then the feasible completion k corresponding to p_n^δ should be added to the RMP if $p_n^\delta - r_n^\delta < 0$. Moreover, $LB_n := \Theta_n^* + \sum_{\delta \in \bar{\Delta}} (p_n^\delta - r_n^\delta)$ is a lower bound on Θ^*.

Furthermore, when the subgradient method terminates, we invoke a greedy procedure to find feasible solutions to the core problem. This procedure, which takes as input a multiplier vector, is repeated up to *maxMulti* times using the multiplier vectors obtained in the last *maxMulti* iterations of the subgradient algorithm. In our experiments, *maxMulti* was set to 100 or 200. The greedy procedure is presented in Algorithm 1. 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. Moreover, we set all $z_i = 1$ (Line 2). We initialize $\hat{\lambda}$ with the current vector of multipliers λ (Line 3). Then, we choose for every original duty the best feasible completion with respect to quasi reduced cost depending on $\hat{\lambda}$ (Lines 4–6). If there are uncovered tasks, we try to cover these using reserve duties where the feasible completion selected in the loop of Lines 4–6 does not cover any tasks. We will refer to these reserve duties as idle reserve duties and they are determined in Line 9. In Line 8 we have set multipliers corresponding to the uncovered tasks to f_i. Given suitable values for the objective function, this means that feasible completions covering any of the uncovered tasks will have negative pseudo reduced cost and will be very attractive when we (possibly) revise the selection of the feasible completion for the idle reserve duties in Lines 10–13.

1 Order the original duties $\delta \in \bar{\Delta}$ by increasing reduced cost of the x_k^δ variables that were set to 1 in the Lagrangian subproblem solution;
2 Set $z_i = 1$ for all $i \in \bar{N}$;
3 Set $\hat{\lambda} = \lambda$;
4 **foreach** $\delta \in \bar{\Delta}$ **do**
5   Choose $k^*(\delta) \in \arg\min\{\bar{c}_k^\delta(\hat{\lambda}) : k \in \bar{K}_n^\delta\}$ 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 **if** $\exists i \in \bar{N}$ *with* $z_i = 1$ **then**
8   Set $\hat{\lambda}_i = f_i$ if $z_i = 1$ for all $i \in \bar{N}$;
9   Construct the set of idle reserve duties $\bar{\Delta}_I = \{\delta \in \bar{\Delta}_R : a_{ik^*(\delta)}^\delta = 0 \;\; \forall i \in \bar{N}\}$;
10   **foreach** $\delta \in \bar{\Delta}_I$ **do**
11   Set $x_{k^*(\delta)}^\delta = 0$;
12   Choose $q^*(\delta) \in \arg\min\{\bar{c}_k^\delta(\hat{\lambda}) : k \in \bar{K}_n^\delta\}$ and set $x_{q^*(\delta)}^\delta = 1$;
13   Set $\hat{\lambda}_i = 0$ and $z_i = 0$ for all $i \in \bar{N}$ with $a_{iq^*(\delta)}^\delta = 1$;

Algorithm 1: Greedy procedure to construct feasible solutions.

When we explore a new core problem, we warm start the RMP with columns generated earlier if possible. In order to do so, we store all generated columns in a column pool.

If a new core problem contains original duties that have been considered in other core problems, we scan the column pool and add columns to the RMP if all tasks covered by the column are included in the new core problem.

3.4.2 Pricing problems

For every original duty $\delta \in \bar{\Delta}$ we build a graph $\bar{G}^{\delta}$ in which every feasible completion k that satisfies the following criterion is represented by a path in $\bar{G}^{\delta}$: Every task i covered by k as well as every task that is used for deadheading in k belongs to $\bar{N}$.

In these graphs we use several types of nodes and arcs in order to model the feasible completions. The source of graph $\bar{G}^{\delta}$ captures the position of original duty δ at the time of rescheduling. There are three possibilities: The duty might not have started (i). If the duty has started, the driver is either performing a task (ii), or he transfers at a station (iii). The sink node of $\bar{G}^{\delta}$ corresponds to the end of an original duty.

Besides the source and sink, we introduce a pair of nodes for the departure and the arrival of each task i. These nodes are connected by an arc representing driving task i. The weight for a task arc $w_i = \mathit{costRole}(i, \delta) - \lambda_i$. Where

$$\mathit{costRole}(i, \delta) = \begin{cases} \mathit{costOwnRole} & \text{, if } i \text{ appears in } \delta \text{ with the same role} \\ \mathit{costOtherRole} & \text{, otherwise.} \end{cases}$$

A copy of the arc is used to model deadheading of a driver on task i, if the driver is not allowed to drive task i due to his route and/or rolling stock licenses. The weight for these deadhead arcs is given by $w_i = \mathit{costRole}(i, \delta)$.

A transfer arc from the arrival node of a task i to the departure node of a task j exists, if a driver can perform task j immediately after task i. In general, this is possible if task j starts at the end station of task i and if either the time between the arrival and the departure is larger than the minimum connection time, or the two tasks are operated with the same rolling stock. The weight for the transfer arc between tasks i and j, $w_{ij} = \mathit{costTransfer}(i, j)$. Here

$$\mathit{costTransfer}(i, j) = \begin{cases} \mathit{costPlanned} & \text{, if the transfer is in any original duty } \delta \\ \mathit{costNew} & \text{, otherwise.} \end{cases}$$

Transfer arcs have a property indicating if this transfer can be used as a meal break. This is the case if the transfer takes place at a station that has a canteen and the transfer time is long enough.

From some stations there are taxi connections to other stations for given periods of the day. This occurs for example if the shunting area is located far from a station or crew base. In this case drivers travel by taxi between the stations and the shunting areas to perform pull-out and pull-in tasks. Moreover, alternative ways of transportation might be used during rescheduling to reposition drivers. These deadhead transfers are also modeled by taxi arcs although they could be bus trips or trips on trains of other operators in reality. For taxi arcs, $w_{ij} = costTaxi(i, j)$ is used to penalize the repositioning.

All arcs leaving the source have weight $w_{si} = costModifyDuty$. Moreover, overtime can be penalized via the weight on arcs entering the sink $w_{it} = costOvertime$.

Constructing the graphs in this way, not every path corresponds to a feasible completion because it might violate the meal break rule. Therefore, we solve the subproblems as resource constrained shortest path problems (see Irnich and Desaulniers (2005)). As resources we use the working time before and after the meal break. Moreover, given a vector of Lagrangian multipliers λ, the cost of every path corresponds to the reduced cost of the feasible completion.

3.4.3 The column generation based heuristic

Our column generation based heuristic using the building blocks as described in Section 3.4.1–3.4.2 is outlined in Algotihm 2. It can be seen as a depth first search 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 *maxPP* of the duties. In Line 9 we use three criteria to decide 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 column generation for a node we check in Line 11 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.

For selecting the columns that we fix in a given node, we use information about how often a column appeared in a Lagrangian subproblem solution while solving the last RMP. For every feasible completion k for every original duty δ, we compute the ratio $R_k^\delta = \frac{s_k^\delta}{U}$, where s_k^δ is the number of times x_k^δ was set to 1 in a Lagrangian subproblem solution during subgradient optimization of the last RMP and U is the number of iterations performed by the subgradient algorithm. We order the feasible completions by decreasing values of R_k^δ. We start with setting the feasible completion with the largest value of R_k^δ to 1 and all other feasible completions from the same original duty to 0. For the same node, we then continue with the feasible completion with the next largest value of R_k^δ as long as $R_k^\delta \geq 0.7$ and the number of original duties for which we fixed the feasible completions in this node is less than *maxFix* percent of the original duties (*maxFix* was set to 10 in our experiments). This scheme is closely related to the α-fixing procedure proposed by Holmberg and Yuan (2000).

$stopFix = \text{false}, LB_F = -\infty, UB^* = \infty, UB_F = 0$;
while *stopFix = false* **do**
  stopColGen = false;
  while *stopColGen = false* **do**
    Compute the lower bound Θ_n^* for the RMP with subgradient optimization;
    Call *GREEDY* with at most *maxMV* multiplier vectors and update UB^*;
    Solve pricing problems and add promising feasible completions;
    Compute LN_n if all pricing problems have been solved;
    if *any stopping criteria for column generation is met* **then**
      stopColGen = true, $LB_F = UB_F + LB_n$;
  if *any stopping criteria for fixing is met* **then**
    stopFix = true;
  else
    Fix the feasible completions for at most *maxFix* original duties and update UB_F;

Algorithm 2: The column generation heuristic to explore a core problem.

3.5 Defining the core problems

3.5.1 Initial core problem

After initial experiments we came up with the following selection of the subset of original duties $\bar{\Delta}$ for the initial core problem. This selection is a good compromise between computation time and solution quality.

We build $\bar{\Delta}$ in four steps. In the first step, we add all tasks which are canceled or modified (rerouted) to N_1. Secondly, we build a set N_2 where we add an unmodified task j if it has the same pair of start and end stations as one of the tasks in N_1 and if its departure time t_j^{dep} lies in the interval $[t_0, t_1 + 60 \text{ minutes}]$, where t_0 is the earliest departure time of a task $i \in N_1$ with $s_i^{\text{dep}} = s_j^{\text{dep}}$ and $s_i^{\text{arr}} = s_j^{\text{arr}}$, t_1 is the latest arrival time of a task $i' \in N_1$ with $s_{i'}^{\text{dep}} = s_j^{\text{dep}}$ and $s_{i'}^{\text{arr}} = s_j^{\text{arr}}$. In the third step we add a task j to N_3 if it is part of the same train as one of the tasks in $N_1 \cup N_2$. Finally, we define the subset of original duties $\bar{\Delta} := \Delta_R \cup \{\delta \in \Delta_A : \delta \text{ covers at least one task in } N_1 \cup N_2 \cup N_3\}$. Note that we include all reserve duties in the initial core problem.

3.5.2 Neighborhoods for uncovered tasks

Given our crew rescheduling problem, the largest improvement in the objective, and the one we are mainly interested in, is covering tasks that have not been covered in the solution of the initial core problem. Therefore, we are interested in neighborhood operators which, given an uncovered task, define a neighborhood such that exploring the neighborhood could lead to a crew schedule that covers more tasks.

In the first step we select a number of candidates. These duties can possibly cover the uncovered task. Usually this would leave other tasks uncovered and in order to assign them to other duties we select in step two for each candidate duty a number of similar duties that offer possibilities to swap parts of the duties.

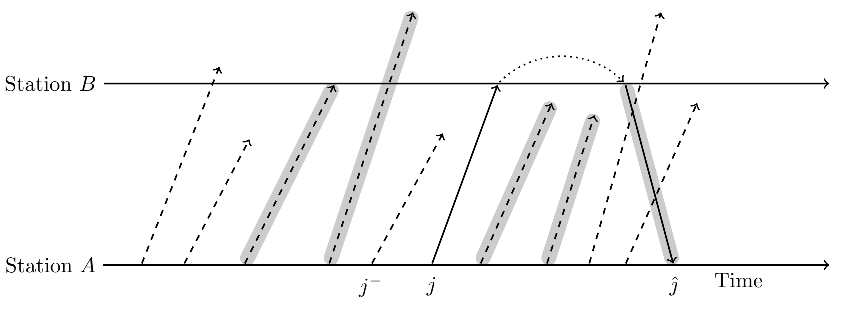


Figure 3.2: Selecting replacement duties that cover tasks leaving from station A just before and after task j.

The candidates in the first step are selected as follows (see Figure 3.2). Given the departure time and station (A in the example) of the uncovered task j we look at task j^- that departs from the same station the closest before task j. Then we consider the replacement duty σ that covers j^- in the current solution and check heuristically, considering rolling stock and route knowledge, if σ could cover j. If yes, then we select σ as a candidate and continue with the next task that departs from station A before j^- until we have selected r candidates. We repeat the procedure considering tasks that depart from station A after task j.

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

In Figure 3.2, we have marked in gray the tasks covered by replacement duties that have been selected. Note that, because of missing route knowledge, task j^- is not marked.

In the second step we select for every candidate the s most similar duties that have not been selected yet. We define similarity between duties in terms of the number of stations that are visited around the same time. We also add a bonus if they share the same current station and crew base. The idea behind this measure is that two duties can possibly swap parts if they have a departure from the same station around the same time and both reach another station later in their duty again around the same time. Given a candidate σ, another duty τ and the set of tasks N_σ and N_τ covered by the duties, we compute the similarity as

$$S(\sigma, \tau) := B(\sigma, \tau) + \sum_{i \in N_\sigma} \sum_{j \in N_\tau} \gamma_{ij}$$

where

$$\gamma_{ij} = \begin{cases} 1 & \text{if } s_i^{\text{dep}} = s_j^{\text{dep}} \text{ and } |t_i^{\text{dep}} - t_j^{\text{dep}}| \leq \omega \\ 0 & \text{otherwise} \end{cases}$$

indicates that tasks i and j depart from the same station within at most ω minutes from each other. The bonus function $B(\sigma, \tau) := B_b(\sigma, \tau) + B_{cs}(\sigma, \tau)$ sums up the bonus for the same crew base B_b and same current station B_{cs}, respectively. For our experiments we use

$$B_b(\sigma, \tau) := \begin{cases} 0.6 & \text{if } b_\sigma = b_\tau \\ 0 & \text{otherwise.} \end{cases}$$

$B_{cs}(\sigma, \tau)$ is defined accordingly.

3.6 Computational experiments

We implemented our solution approach in C++ and compiled it with the Visual C++ 8.0 compiler. We ran our experiments on an *Intel Pentium D* processor with 2 GB RAM clocked at 3.4 GHz.

For the objective function we specified the following cost coefficients. The value of f_i depends on the type of the task. Canceling a task from a station A to another station B would make the underlying rolling stock schedule infeasible, therefore we set $f_i = 20,000$

for these tasks. Under the mild assumption that the rolling stock assigned to a task from A to A can be moved to the shunting area and pulled out again, these tasks leave the rolling stock schedule intact. Since this situation is preferred from the point of view of the overall disruption management process, we set the corresponding f_i to $3,000$. The cost of each feasible completion of a duty is zero if the duty is unchanged or the sum of penalties depending on the way the duty is changed. We used the following values for penalties: $costModifyDuty = 400$ if a duty is changed, $costOtherRole = 50$ for every task that is not assigned to its original duty, $costNew = 1$ for every transfer between two tasks that was not used in the original plan by some duty and $costTaxi = 1,000$ if the driver has to be repositioned using a taxi. In the experiments we had no penalties for short connection times and overtime.

3.6.1 Instances

As a starting point for our instances we remodeled five scenarios, spread over the country, that happened in the past. All scenarios lasted about three hours. Therefore, we chose an estimated duration of 3 hours for our remodeled instances. For every historical scenario, we generated a second disruption with the same estimated duration but at a different time of the day. We modified the timetable following the main ideas behind the emergency scenarios. Because rescheduling of rolling stock is in itself a difficult optimization problem, we considered a simplified rolling stock schedule, which can easily be adapted to the new timetable. For the original duties we used a crew schedule that was operated by NS on a workday somewhere in September 2007.

A general description of the 10 cases is given in Table 3.1. The disruptions around Abcoude, which is located between Utrecht and Amsterdam, and around 's-Hertogenbosch, which is located south of Utrecht, involve heavily used routes. More than 50 original duties are affected by the disruptions in these instances. The involved routes in the instances at Beilen and Lelystad are not so heavily used, but the route blockages disconnect some ends of the railway network from the remaining part. The instances at Zoetermeer also involve a heavily used route, but in these instances a reduced number of trains can be operated on the involved route, because it is not completely blocked.

3.6.2 Results for initial core problems

In a first series of experiments, we applied our algorithm to the described scenarios but we solved only the initial core problems. These core problems were constructed as described in Section 3.5.1. In addition, we included a number of reserve duties. Recall that the crew schedule of NS contains about 90 reserve duties on an average workday. At the time a large disruption occurs and rescheduling takes place, not all reserve duties might

Location	ID	Time	Type	Affected duties
Abcoude	Ac A	11:00-14:00	two sided blockage	59
Abcoude	Ac B	16:30-19:30	two sided blockage	53
Beilen	Bl A	07:00-10:00	two sided blockage	15
Beilen	Bl B	16:00-19:00	two sided blockage	15
's-Hertogenbosch	Ht A	08:00-11:00	two sided blockage	55
's-Hertogenbosch	Ht B	15:30-18:30	two sided blockage	51
Lelystad	Lls A	04:00-07:00	two sided blockage	25
Lelystad	Lls B	13:00-16:00	two sided blockage	22
Zoetermeer	Ztm A	08:00-11:00	reduced number of trains	21
Zoetermeer	Ztm B	11:30-14:30	reduced number of trains	25

Table 3.1: Summary of the different instances.

be available for rescheduling. One reason is that reserve duties are also used when a train driver missed a connection because of a delay. Moreover, another disruption could have happened earlier and reserve duties might have been used in order to recover from this disruption. In order to take this into account somehow during our experiments, we derived three sets of reserve duties R_1–R_3 from a given initial plan R_0 in the following way. In the set R_1 (R_2), every reserve duty from R_0 had a probability of 50 % (25 %) to be included in R_1 (R_2) as well. Based on drawing a single random number between 0 and 1 for every reserve duty in R_0 we obtained the sets R_1 and R_2. Note here that the drawing for R_2 was done independently of the drawing for R_1. This procedure resulted in sets R_1 with 46 reserve duties and R_2 with 20 reserve duties. Finally, set R_3 does not contain any reserve duty at all.

In Tables 3.2–3.4 we report the results for the three sets of reserve duties R_1–R_3, respectively. Here the columns have the following meanings. The first column is the *Id* of the instance. $|\bar{\Delta}|$ is the number of original duties in the initial core problem (finished reserve duties are excluded) and $|\bar{N}|$ is the number of set covering constraints in (3.6). Column *LB* reports the value of LB_n when terminating the column generation. *UB* is the value of the best feasible solution. *GAP* is the percentage gap between *LB* and *UB*. *Time* is the computation time in seconds. In the last five columns, we give some insight into the best feasible solution. *A-A* and *A-B* denote the number of tasks of the types A-A and A-B that need to be canceled. *Taxi* is the number of additional taxi trips used. *MD* is the number of active original duties that are feasible but modified in the solution. *UR* is the number of reserve duties that cover tasks in the solution. Note that the total number of original duties that are modified is the sum of *MD*, *UR* and the number of affected duties (see Table 3.1).

Id	$\vert\bar{\Delta}\vert$	$\vert\bar{N}\vert$	LB	UB	GAP (%)	Time (s)	A-B	A-A	Taxi	MD	UR
Ac_A	163	602	63089	63857	1.2	201	1	0	1	79	8
Ac_B	143	734	35795	36134	0.9	255	0	0	0	67	5
Bl_A	80	234	10581	10581	0.0	26	0	0	1	17	2
Bl_B	63	186	9532	9532	0.0	18	0	0	0	18	3
Ht_A	133	614	37911	38094	0.5	193	0	0	5	59	1
Ht_B	119	665	62900	62922	< 0.1	177	1	1	4	59	4
Lls_A	81	239	17623	17646	0.1	33	0	0	3	30	5
Lls_B	161	456	21796	21796	0.0	107	0	0	4	27	1
Ztm_A	112	508	11840	11935	0.8	87	0	0	0	23	1
Ztm_B	110	423	34736	34736	0.0	66	1	0	0	27	1

Table 3.2: Results for initial core problems with 46 reserve duties (R_1).

Id	$\vert\bar{\Delta}\vert$	$\vert\bar{N}\vert$	LB	UB	GAP (%)	Time (s)	A-B	A-A	Taxi	MD	UR
Ac_A	137	602	45280	46072	1.7	148	0	0	1	78	5
Ac_B	124	734	36734	38267	4.2	230	0	0	0	66	3
Bl_A	54	234	10583	10583	0.0	17	0	0	1	17	2
Bl_B	44	186	10417	10487	0.7	18	0	0	0	19	3
Ht_A	107	614	38680	39298	1.6	200	0	0	6	60	2
Ht_B	97	665	67343	67426	0.1	137	1	2	5	57	2
Lls_A	55	239	18996	19254	1.4	23	0	0	5	28	3
Lls_B	133	456	22445	22445	0.0	102	0	0	5	27	1
Ztm_A	86	508	11936	11936	0.0	49	0	0	0	23	1
Ztm_B	84	423	34936	34936	0.0	49	1	0	0	26	0

Table 3.3: Results for initial core problems with 20 reserve duties (R_2).

First of all, we observe that all computation times are less than 5 minutes. The computation time mainly depends on the size of the core problems in terms of the number of set covering constraints. For example, for set R_1, computation times range from 18 seconds for Bl_B to 255 seconds for Ac_B.

Moreover, the number of canceled tasks is at most 3 for the experiments with reserve duties and at most 4 for the experiments without reserve duties. The number of instances where all tasks are covered in the solution, is equal to 7, 8, and 6 for sets R_1, R_2 and R_3, respectively. It is not surprising that without reserve duties (R_3) the number of instances where all tasks can be covered is less compared to the experiments with reserve duties (R_1 and R_2). Interestingly, with R_2 we can cover all tasks in more instances compared to R_1, while the absolute number of reserve duties is only 20 compared to 46 in R_1. This indicates that it is important where and when reserve duties are available for rescheduling. Furthermore, we observe that at most 8 reserve duties are used.

Id	$\lvert\bar{\Delta}\rvert$	$\lvert\bar{N}\rvert$	LB	UB	GAP (%)	Time (s)	A-B	A-A	Taxi	MD	UR
Ac_A	117	602	71083	72399	1.9	142	1	1	1	82	-
Ac_B	111	734	37100	38919	4.9	261	0	0	0	65	-
Bl_A	34	234	11516	11590	0.6	20	0	0	3	15	-
Bl_B	31	186	13105	15200	16.0	17	0	1	1	20	-
Ht_A	87	614	38701	39799	2.8	206	0	0	6	58	-
Ht_B	83	665	70323	70379	0.1	219	1	3	5	55	-
Lls_A	35	239	20575	21698	5.5	20	0	0	8	25	-
Lls_B	115	456	23112	23752	2.8	106	0	0	5	29	-
Ztm_A	66	508	12290	12290	0.0	40	0	0	0	23	-
Ztm_B	64	423	34936	34936	0.0	45	1	0	0	26	-

Table 3.4: Results for initial core problems without reserve duties (R_3).

Furthermore, we see that the impact of crew rescheduling on the whole crew schedule differs significantly. The impact is limited considering the experiments with reserve duties. Without reserve duties more duties are modified for most of the instances. In some cases the absence of reserve duties can be compensated by using more additional taxi trips and/or modifying more duties.

3.6.3 Results with neighborhood exploration

We have seen that the solutions to the core problems are good in terms of the number of canceled tasks especially when reserve duties are available. However, we would like to see if some of the uncovered tasks can be covered when we explore a neighborhood as defined in Section 3.5.2. In the following, we only consider the instances where cancellation of tasks occurs in the solution of the initial core problem.

We present our results in Tables 3.5–3.8. In these tables the first column is the *Id* of the instances. Column *It* is the number of the core problem exploration in the overall algorithm (see Figure 3.1), where a 1 corresponds to the initial core problem and a number greater than 1 corresponds to a neighborhood exploration. *Fixed* provides the total cost of the fixed duties when the current core problem is solved. $|\bar{\Delta}|$ and $|\bar{N}|$ are the number of original duties and set covering constraints in the core problem. *LB*, *UB*, and *GAP* give the values for LB_n, the best feasible solution cost and the percentage gap. The next four columns show the status of the overall algorithm. *Sol*, which is equal to *Fixed* + *UB*, is the objective value of the new crew schedule. TT is the total computation time of the algorithm. *A-B* and *A-A* are the number of canceled tasks of the corresponding types.

We first tried a relatively small neighborhood, where r and s were set to 3. When considering R_1 as the set of reserve duties we can improve the solution of the initial core problems in 2 out of 3 cases and find solutions that cover all tasks (see Table 3.5).

Exploring the neighborhoods of the uncovered tasks took between 12 and 35 seconds. With R_2 as the set of reserve duties we can improve the solution in 1 of the 2 cases (Table 3.6).

For the case Ht_B with R_1, we also tried to obtain a better solution by exploring a larger neighborhood of the uncovered task. We tried the settings $r = s = 6$ and $r = s = 9$. The first setting increased the size of the core problems to $|\bar{\Delta}| = 111$ and $|\bar{N}| = 767$ in iteration 2 and $|\bar{\Delta}| = 122$ and $|\bar{N}| = 829$ in iteration 3. The total computation time was 432 seconds. For the second setting we observed $|\bar{\delta}| = 188$ and $|\bar{N}| = 1535$, and $|\bar{\delta}| = 244$ and $|\bar{N}| = 1527$ for the core problems in iterations 3 and 4 respectively. The computation took 1258 seconds. However, we could not find better solutions in terms of the number of canceled tasks. We obtained similar results for the same experiments with Ht_B and R_2.

Id	It	Fixed	$\lvert\bar{\Delta}\rvert$	$\lvert\bar{N}\rvert$	LB	UB	GAP (%)	Sol	TT (s)	A-B	A-A
Ac_A	1	0	163	602	63089	63857	1.2	63857	201	1	0
Ac_A	2	39791	74	241	4851	5621	15.9	45412	236	0	0
Ht_B	1	0	119	665	62900	62922	< 0.1	62922	177	1	1
Ht_B	2	33439	55	209	29483	29483	0.0	62922	189	1	1
Ht_B	3	33842	60	284	29080	29080	0.0	62922	209	1	1
Ztm_B	1	0	110	423	34736	34736	0.0	34736	66	1	0
Ztm_B	2	13679	72	142	1558	1558	0.0	15237	83	0	0

Table 3.5: Results for neighborhood exploration with $r = 3, s = 3$ using 46 reserve duties (R_1).

Id	It	Fixed	$\lvert\bar{\Delta}\rvert$	$\lvert\bar{N}\rvert$	LB	UB	GAP (%)	Sol	TT (s)	A-B	A-A
Ht_B	1	0	97	665	67343	67426	0.1	67426	137	1	2
Ht_B	2	33892	36	218	33534	33534	0.0	67426	154	1	2
Ht_B	3	34799	40	303	32627	32627	0.0	67426	173	1	2
Ht_B	4	35401	36	239	32025	32025	0.0	67426	184	1	2
Ztm_B	1	0	84	423	34936	34936	0.0	34936	49	1	0
Ztm_B	2	14383	47	139	1056	1056	0.0	15439	61	0	0

Table 3.6: Results for neighborhood exploration with $r = 3, s = 3$ using 20 reserve duties (R_2).

In Table 3.7, we present the results of the neighborhood exploration with $r = 4$ and $s = 4$ when no reserve duties are present (R_3). For 3 out of the 4 considered instances we can significantly improve on the solutions of the initial core problems. Moreover, for Ztm_B we were able to find a solution that covers all tasks.

Id	It	Fixed	$\lvert\bar{\Delta}\rvert$	$\lvert\bar{N}\rvert$	LB	UB	GAP (%)	Sol	TT (s)	A-B	A-A
Ac_A	1	0	117	602	71083	72399	1.9	72399	142	1	1
Ac_A	2	39709	55	275	32087	32641	1.7	72350	175	1	1
Ac_A	3	42919	52	347	11939	11939	0.0	54858	194	0	1
Bl_B	1	0	31	186	13105	15200	16.0	15200	17	0	1
Bl_B	2	6323	36	304	7402	8826	19.2	15149	38	0	1
Bl_B	3	7379	36	307	6346	7770	22.4	15149	60	0	1
Bl_B	4	6323	37	307	7402	8826	19.2	15149	83	0	1
Ht_B	1	0	83	665	70323	70379	0.1	70379	219	1	3
Ht_B	2	32941	39	389	37438	37438	0.0	70379	239	1	3
Ht_B	3	28933	45	452	38397	38397	0.0	67330	280	1	2
Ht_B	4	32892	39	387	34438	34438	0.0	67330	297	1	2
Ztm_B	1	0	64	423	34936	34936	0.0	34936	45	1	0
Ztm_B	2	12876	43	275	2563	2563	0.0	15439	62	0	0

Table 3.7: Results for neighborhood exploration with $r = 4, s = 4$ without reserve duties (R_3).

We run our algorithm again after increasing r and s to 6. With this setting, which generates larger neighborhoods, we found better solutions for 4 of the 4 instances (see Table 3.8). Moreover, we found solutions covering all tasks for 3 of the 4 instances.

Comparing the results with the two different choices of r and s we can see that we can obtain better results by spending more time in exploring larger neighborhoods.

Id	It	Fixed	$\lvert\bar{\Delta}\rvert$	$\lvert\bar{N}\rvert$	LB	UB	GAP (%)	Sol	TT (s)	A-B	A-A
Ac_A	1	0	117	602	71083	72399	1.9	72399	142	1	1
Ac_A	2	33683	103	562	18398	19537	6.2	53220	288	0	0
Bl_B	1	0	31	186	13105	15200	16.0	15200	17	0	1
Bl_B	2	5318	67	525	8088	8088	0.0	13406	73	0	0
Ht_B	1	0	83	665	70323	70379	0.1	70379	219	1	3
Ht_B	2	26567	82	810	43812	43812	0.0	70379	335	1	3
Ht_B	3	20906	88	798	46025	46025	0.0	66931	462	1	2
Ht_B	4	24817	83	848	39267	39267	0.0	64084	553	1	1
Ztm_B	1	0	64	423	34936	34936	0.0	34936	45	1	0
Ztm_B	2	12876	84	612	2563	2563	0.0	15439	96	0	0

Table 3.8: Results for neighborhood exploration with $r = 6, s = 6$ without reserve duties (R_3).

3.7 Application: The Vleuten case

In March 2009 the crew rescheduling algorithm proposed in Section 3.3 was put to a test (Kroon and Huisman (2009)). On Monday, March 23, 2009, two carriages of a freight train derailed near station Vleuten (Vtn). Due to this accident, the railway infrastructure of the route between Woerden (Wdn) and Utrecht (Ut) was damaged over 5 kilometers. It took nearly a week before the repair works had been completed and the route could be used again at full capacity.

Initially the route between Woerden and Utrecht was blocked completely. On Tuesday, March 24, it was possible to run trains on one track. This situation lasted until the evening of Sunday, March 29. For this time period an alternative timetable was operated for the 2000-line between The Hague (Gvc) and Utrecht. It was decided that the trains of the 2000-line from Utrecht towards The Hague should go over the planned route, but the trains in the opposite direction should run on an alternative route via Breukelen (Bkl) where they can turn towards Utrecht (Firgure 3.3). As a consequence of this rerouting, the timetable of other trains had to be modified as well.

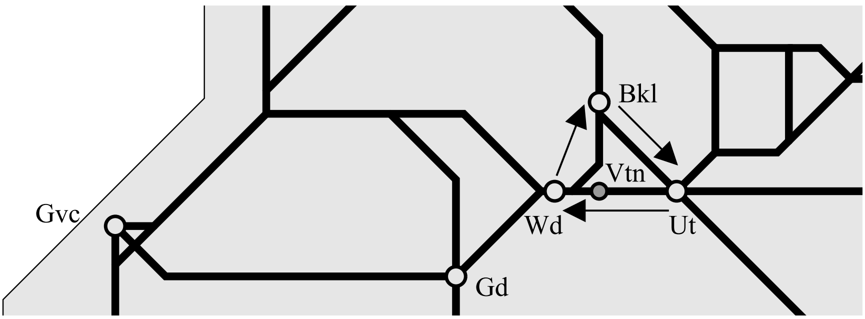


Figure 3.3: The routes for the 2000-line when only one track was open between Woerden (Wd) and Utrecht (Ut) due to the damage caused by the derailing of a cargo train near station Vleuten (Vtn).

On Monday and Tuesday the duties for the train drivers and for the conductors have been rescheduled manually by the dispatchers. For Wednesday and Thursday the duties of the drivers were rescheduled with the method described in Section 3.3. In total, on each of the considered days, around 260 duties had become infeasible due to the timetable modifications. The algorithm was able to find good feasible solution in about 1 hour of computation time. However, the neighborhood exploration was not needed since all tasks were covered after the initial core problem was solved. For the convenience of the drivers it was decided to not allow any duties to end later than planned. The algorithm of Section 3.3 was designed to take rolling stock and route knowledge into account at an individual level. However, at NS this information is not available in any electronic format. Fortunately, the computed solutions contained only few conflicts with respect to the rolling stock or route knowledge which could be resolved by hand relatively easily.

At that time, it was not possible to import the solution of the algorithm into the computer system used at the Network Operations Control of NS. Therefore two dispatchers manually typed in all new duties during the nights before the new duties should be operated.

The duties for the conductors have still been rescheduled manually. Therefore, we are able to compare the algorithmic approach with the current manual process. During the night, four dispatchers managed to reschedule all infeasible conductor duties that started before 13:00. The remaining infeasible duties had to be rescheduled during the operations. This resulted in a lot of communication during the operations. Furthermore, many duties for the conductors did finish later than the regular time.

All crew duties for the last three days (Friday until Sunday), when only one track was available, had be rescheduled using the CREWS crew scheduling system (see Morgado and Martins (1998)). However, the lead time of using the CREWS system was too long to reschedule the duties on Wednesday and Thursday.

The Vleuten case is not an example of real-time crew rescheduling, but it is close enough to draw some conclusions about algorithmic decision support for disruption management at NS. The case clearly revealed the advantages of automated decision support for crew rescheduling: It leads to better solutions in less time. We therefore strongly recommend an integration of algorithmic decision support into the computer systems used at Network Operations Control.

3.8 Summary and conclusion

We have proposed an algorithm to solve the OCRSP. Given a disruption and a real-life crew schedule from NS, we have shown how to select a subset of the original duties in the crew schedule such that we can find solutions of good quality within a short amount of time. This was achieved by combining column generation and Lagrangian relaxation into a heuristic algorithm. The proposed column fixing also enables us to obtain good solutions for the larger instances.

Furthermore, we developed an extension, namely exploring neighborhoods of tasks which could not be covered with the initial selection of duties. We have shown that with this extension, it is possible to reduce the number of canceled tasks in many cases. This is an important improvement compared to algorithms which rely on an a priori defined core problem. In our experiments, considering two sets of reserve duties, we can cover all tasks in 9 out of 10 instances. In the case where we do not consider any reserve duties, we can increase the number of instances where no tasks need to be canceled from 6 after solving the initial core problem to 9 by our neighborhood exploration scheme.

We believe that the idea of neighborhood exploration can be used in other areas of rescheduling as well. Moreover, our algorithm can easily be extended by new neighborhood definitions. This could further improve the performance of the algorithm.

Chapter 4

Computational Evaluation of Solution Approaches for Railway Crew Rescheduling

4.1 Introduction

In this chapter we will compare the CGDDS solution approach for operational crew rescheduling developed in Chapter 3 with two alternative solution approaches. This is motivated by two main reasons.

First, it would be very interesting to compare the CGDDS approach to a manual solution approach. Unfortunately a comparison is not so easy for several reasons. First of all, the solutions obtained by the dispatchers do not satisfy all constraints that we take into account. Although this situation is highly undesirable due to the negative impact on the operations and on driver satisfaction, it is sometimes too hard for a dispatcher to find a feasible solution at all. Second, the instances presented in Chapter 3.6 are instances with one disruption. However, in practice there are usually several disruptions on a day. The manual rescheduling of the driver duties for such severe disruptions as considered in our experiments can keep dispatchers busy for several hours. This makes it unlikely that no other disruption has occurred in the meantime. In addition, in the computer system used by NS it is only possible to access the information on how the duties have been performed. Since a duty may have been modified more than once, we cannot get the dispatcher's solution for the single disruption that we study in this thesis. To overcome these problems we will present a new heuristic for railway crew rescheduling that tries to mimic the manual solution approach of the dispatchers.

The second reason for considering alternatives to the CGDDS method is the following. By considering core problems, the CGDDS approach explores only a part of the solution space. Naturally the question arises how good the solutions of the CGDDS approach are. We are going to evaluate the solutions by comparing them to solutions and lower bounds that we obtain from solving larger core problems.

We believe that the merits of the CGDDS approach are that it finds very good solutions in acceptable computation time. We seek to confirm this by answering the following three questions:

(i) Can an improvement heuristic that mimics the manual approach of the dispatchers provide solutions of good quality in much shorter computation time?

(ii) How good are the solutions of the CGDDS algorithm compared to lower bounds obtained by considering larger core problems?

(iii) How does dynamic constraint aggregation (DCA), a state-of-the-art column generation method presented in Elhallaoui et al. (2005, 2008), perform in terms of solution quality and time?

In Elhallaoui et al. (2005, 2008) it was shown that the computation time needed to solve linear relaxations of vehicle and crew scheduling problems can be reduced significantly by using DCA. This motivates the following application of DCA as a substitution for the CGDDS approach. The intuition is to aggregate as many tasks as possible according to the still feasible original duties and to let DCA by disaggregation determine which parts of the search space should be explored. So instead of working with small core problems, we consider much more original duties and tasks. As we will show later on, such an approach suffers from long computation times if a standard branch-and-price heuristic is used to solve these problems.

This chapter is organized as follows. In Section 4.2 we present an original improvement heuristic based on the manual approach of the dispatchers. In Section 4.3 we show how crew rescheduling problems can be solved by DCA. We will discuss how we adapted the pricing problems and how we defined the initial clusters for the initial aggregation of the constraints. An extensive computational comparison is provided in Section 4.4. We finish with some concluding remarks in Section 4.5. Note that we reuse the notation introduced in Chapter 3.

4.2 A heuristic mimicking the manual approach

We will present a heuristic for crew rescheduling that uses the concept of shortest paths with resource constraints and tries to mimic the way dispatchers manually reschedule the crew duties. From interviews with dispatchers of NS we concluded that their manual rescheduling process roughly follows a two phase approach. In the first phase the dispatchers try to construct a feasible crew schedule in a greedy way. In this phase the dispatchers need to find a feasible completion for every infeasible original duty. When constructing these feasible completions they seek to use parts of the infeasible duties to cover as many tasks as possible. When they have found a feasible completion for every duty and there are tasks left uncovered, they attempt to cover these by utilizing the reserve duties. At the end of the first phase they have a feasible replacement duty for every original duty and probably a number of uncovered tasks. In the second phase, the improvement phase, the dispatchers seek to resolve the uncovered tasks one by one. When trying to assign a task to a duty they try to avoid newly uncovered tasks if possible. If this is not possible, a new feasible completion that covers the uncovered task under consideration would be accepted under a certain condition. This condition is that they would accept to leave another task uncovered, if the latter task starts later than the task in their focus. The motivation behind this is to move the problems to a later point in time which gives the dispatchers more time to resolve the new problems later on.

Example 4.1
An example of such a situation is shown in Figure 4.1. The dispatchers want to cover task 21740/a which starts in Utrecht (Ut) at 13:17. Figure 4.1.a shows the original duty "Nm 12". This could be replaced by the feasible replacement duty shown in Figure 4.1.b which covers task 21740/a instead of task 9842/a. Since the start time of task 9842/a is 13:36 the dispatchers would choose this option if they could not find a better one.

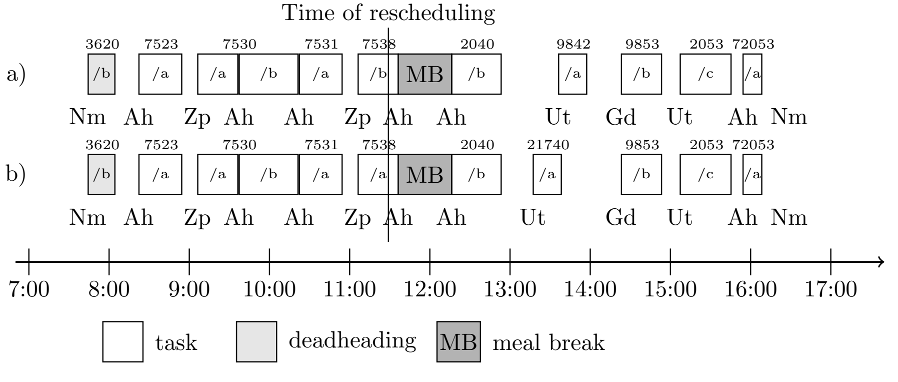


Figure 4.1: An original duty that does not cover tasks 21740/a and a possible replacement duty that covers task 21740/a instead of task 9842/a.

We will mimic the manual solution method of the dispatchers with a heuristic called two phase repeated shortest path problem with resource constraints (2P-RSPPRC). The feasible completions that we will consider are computed as solutions to auxiliary shortest path problems with resource constraints (SPPRC) on a weighted directed acyclic graph similar to the ones used in the column generation pricing problems described in Section 3.4.2. The length of a path will measure the attractiveness of a feasible completion and is dependent on the phase and the uncovered task in the focus. Next to the task arcs whose weights will be determined dynamically, only the taxi arcs will have non-zero weights as specified in Section 3.4.2. The resource constraints guarantee feasibility of the corresponding replacement duties with respect to the rules described in Section 2.5.1.

4.2.1 Phase 1: Repairing the infeasible duties

We formalize the idea of the first phase in the manual approach as follows (see Algorithm 3). In Line 1 we construct auxiliary graphs for all infeasible duties Δ_C and all idle reserve duties Δ_I. The set of tasks $\hat{N}$ which is used for constructing the graphs is an input to the algorithm. Via this mechanism we can control the size of the graphs. Let $\hat{N}_C$ be the tasks that had been assigned to the infeasible duties. In Line 2, we order the infeasible original duties Δ_C by increasing remaining duty time. The weights w_i on the task arcs are set in Line 3. After some initial experiments we defined

$$bonusTask(i, \delta) = \begin{cases} 19,900 & \text{if } i \in \hat{N}_C, s_i^{\text{arr}} \neq s_i^{\text{dep}} \text{ and } i \text{ was covered by } \delta' \neq \delta \\ 2,900 & \text{if } i \in \hat{N}_C, s_i^{\text{arr}} = s_i^{\text{dep}} \text{ and } i \text{ was covered by } \delta' \neq \delta \\ 20,000 & \text{if } i \text{ was covered by } \delta \text{ and } s_i^{\text{arr}} \neq s_i^{\text{dep}} \\ 3,000 & \text{if } i \text{ was covered by } \delta \text{ and } s_i^{\text{arr}} = s_i^{\text{dep}} \\ 0 & \text{otherwise (not covered by any } \delta' \in \Delta_C) \end{cases}$$

The values in $bonusTask(i, \delta)$ are inspired by the values for not covering a task of type A-A and A-B as specified in Section 3.6. We found that giving a bit more priority to tasks that have been originally assigned to a duty is beneficial. We then solve the auxiliary SPPRC for every original duty. We assign the feasible completion that is represented by the solution to the SPPRC to the original duty (Line 5). In Line 6 we set the arc weights for the task arcs corresponding to the tasks covered by the feasible completion to 0. In Lines 7–9 we assign for every idle reserve duty the "best" feasible completion according to the auxiliary SPPRC.

After Phase 1, a feasible completion has been assigned for every original duty. Moreover, we have a list of tasks N_u that are not covered by any of the chosen feasible completions. This new crew schedule with uncovered tasks is the input for the second phase of the 2P-RSPPRC heuristic.

Build the auxiliary graphs for duties $\delta \in \Delta_C \cup \Delta_I$ based on the tasks in $\hat{N}$;
Order the original duties $\delta \in \Delta_C$ by increasing remaining duty time;
Set $w_i = -bonusTask(i, \delta)$ for all $i \in \hat{N}$;
foreach $\delta \in \Delta_C$ **do**
    Compute an optimal solution to the SPPRC. Let $k^*(\delta)$ be the corresponding feasible completion;
    Set $w_i = 0$ for all $i \in \hat{N}$ with $a^{\delta}_{ik^*(\delta)} = 1$;
foreach $\delta \in \Delta_I$ **do**
    Compute an optimal solution to the SPPRC. Let $k^*(\delta)$ be the corresponding feasible completion;
    Set $w_i = 0$ for all $i \in \hat{N}$ with $a^{\delta}_{ik^*(\delta)} = 1$;

Algorithm 3: Phase 1 of the 2P-RSPPRC method

4.2.2 Phase 2: Improving the crew schedule

As mentioned earlier, in Phase 2 we try to improve the current solution to the OCRSP. As in Phase 1, the SPPRC on auxiliary graphs will be the main tool. Considering all tasks at the same time would make the auxiliary graphs too large, therefore we are a looking for a good compromise between the quality of the solutions of the heuristic and the required computational time. To this end, we consider a horizon, that is a subset of tasks and original duties, that can be redefined dynamically during Phase 2. The definition of a horizon given above is quite general, for our implementation we rely on the neighborhood definition as presented in Section 3.5.2. A horizon is then built as the union of tasks and duties of the neighborhoods of a list of uncovered tasks.

Algorithm 4 shows the details of the improvement phase. Let N_T be a list of uncovered tasks sorted by increasing departure times. In Line 2 we remove the first task from list N_T. In this iteration we will focus on this task u. The function *updateHorizon*() indicates if a new horizon should be built or not. The definition of *updateHorizon*() determines how often a new horizon is constructed. If needed, the new horizon H and the resulting auxiliary graphs are constructed in Line 4–5. One extreme implementation of the functions *updateHorizon*() and *buildHorizon*() would be that *updateHorizon*() returns *true* only for the first $u \in N_T$ and defines a horizon based on the complete list $L = N_T$. The other extreme would be that *updateHorizon*() would always return *true* and to pass only task u as argument to *buildHorizon*(). Next, the arc weights w_i are updated in order to favor

1 **while** $size(N_T) > 0$ **do**
2 $\quad u = pop(N_T)$;
3 $\quad$ **if** updateHorizon(u) **then**
4 $\quad\quad H = buildHorizon(L \subseteq N_T)$;
5 $\quad\quad$ Build the auxiliary graphs based on H;
6 $\quad$ Set $w_i = -bonus(i, u, \delta)$ for all $i \in N_H$;
7 $\quad$ **foreach** $\delta \in \Delta_H$ **do**
8 $\quad\quad$ Compute an optimal solution to the SPPRC. Let $k^*(\delta)$ be the corresponding feasible completion;
9 $\quad\quad$ Select $q^*(\delta) \in \arg\min\{k^*(\delta) : \delta \in \Delta_H\}$;
10 $\quad\quad$ **if** *u is still uncovered* **then**
11 $\quad\quad\quad$ Add u to N_U;
12 $\quad\quad$ Update and sort N_T;

Algorithm 4: Phase 2 of the 2P-RSPPRC method

feasible completions covering task u using the function $bonus(i, u, \delta)$ (Line 6). Here

$$bonus(i,u,\delta) = \begin{cases} 19,990 & \text{if } i \in N_U \text{ and } s_i^{\text{arr}} \neq s_i^{\text{dep}} \\ 2,990 & \text{if } i \in N_U \text{ and } s_i^{\text{arr}} = s_i^{\text{dep}} \\ 22,300 & \text{if } i = u \text{ and } s_i^{\text{arr}} \neq s_i^{\text{dep}} \\ 5,300 & \text{if } i = u \text{ and } s_i^{\text{arr}} = s_i^{\text{dep}} \\ 20,000 & \text{if } i \text{ is covered by } \delta \text{ and } s_i^{\text{arr}} \neq s_i^{\text{dep}} \\ 3,000 & \text{if } i \text{ is covered by } \delta \text{ and } s_i^{\text{arr}} = s_i^{\text{dep}} \\ 0 & \text{otherwise } (i \notin N_U \text{ and not covered by } \delta) \end{cases}$$

The values in $bonus(i, u, \delta)$ have been chosen in order to obtain the desired behavior of the algorithm. As starting point we took the penalty values for not covering a task of type A-A and A-B as specified in Section 3.6. For task u we increase the value to $22,300$ and $5,300$ respectively. This means that it will be attractive to cover task u in exchange for a task of the same type, even if this requires two additional taxi trips. Tasks $i \in N_U$ are made less attractive than tasks covered by the original duty under consideration. In Line 8, we solve the SPPRC on the auxiliary graph for the original duties in the horizon. We guarantee to still cover all tasks starting before task u by choosing an appropriate source node for every original duty. In Line 9 we modify the crew schedule by replacing the current feasible completion for one original duty δ. In Line 11 we add task u to the list of permanently uncovered tasks N_U, if it is not covered. The temporary list of uncovered tasks N_T is updated and sorted in Line 12. Thereby we only consider tasks $i \notin N_U$.

4.2.3 Computational results

We implemented the 2P-RSPPRC heuristic in C++ and compiled it with the Visual C++ 8.0 compiler. We ran our experiments on an *Intel Pentium D* processor with 2 GB RAM clocked at 3.4 GHz.

In Table 4.1 we present the results for the test instances considering 46 reserve duties. The first column shows the name of the instance. *Obj* displays the objective value of the solution at the end of the phase. *A-B* and *A-A* are the number of tasks of type A-B and A-A respectively that are uncovered at the end of the phase. Column T reports the computation time in seconds for the corresponding phase. For Phase 2 we tested two settings for the horizon update strategy. For the first setting *SET1* we considered up to three tasks when selecting the tasks and original duties for building the auxiliary graphs ($|L| = \min\{|N_T|, 3\}$). We set the parameters for constructing the neighborhoods to $r^{\text{earlier}} = 2$, $r^{\text{later}} = 6$ and $s = 3$. As can be seen from Table 4.1 the computation times for phase 2 are too long for practical purpose with this setting. Therefore, we tested a second setting *SET2*. Here we build a new horizon for every focus task u ($|L| = 1$). Moreover, we set $r^{\text{earlier}} = 2$, $r^{\text{later}} = 6$ and $s = 2$. Given this setting we tried three further settings where respectively r^{earlier}, r^{later} and s had been reduced by one as opposed to setting *SET2*. For each of the new three settings we observed at least one instance with more uncovered tasks of type A-B and therefore we decided to not report these results.

	Phase 1				Phase 2							
					SET1				*SET2*			
Id	Obj	A-B	A-A	T (s)	Obj	A-B	A-A	T (s)	Obj	A-B	A-A	T (s)
Ac_A	332745	14	9	40	149651	5	7	935	207579	8	7	265
Ac_B	166478	6	7	61	135824	5	3	313	141822	5	5	121
Bl_A	27917	0	3	4	21070	0	1	32	21070	0	1	19
Bl_B	13852	0	0	3	13852	0	0	0	13852	0	0	0
Ht_A	159920	4	13	42	76704	1	5	1183	112253	2	10	263
Ht_B	135856	3	11	44	68981	0	9	561	68974	0	9	93
Lls_A	27811	0	0	5	27811	0	0	0	27811	0	0	0
Lls_B	54309	1	2	27	54309	1	2	4	54309	1	2	5
Ztm_A	16777	0	0	33	16777	0	0	0	16777	0	0	0
Ztm_B	97052	4	0	17	37714	1	0	51	37260	1	0	38

Table 4.1: Results of the 2P-RSPPRC heuristic with 46 reserve duties.

Inspecting the results for Phase 1, we see that for three instances Bl_B, Lls_A, and Ztm_A all tasks are covered. The computation time for phase 1 is at most 61 seconds. However, more than 10 tasks are uncovered for Ac_A, Ac_B, Ht_A, and Ht_B. Except for instance Lls_B the number of uncovered tasks could be reduced in phase 2 with both settings *SET1* and *SET2*. However, 5 or more tasks of type A-B remain uncovered for

Ac_A and Ac_B. This indicates that these instances are too complicated to find good solutions with a simple heuristic as 2P-RSPPRC.

In Table 4.2 we present the results for the case when no reserve duties are available. The columns have the same meaning as in Table 4.1. First of all we observe that without reserve duties, there is no instances without uncovered tasks after Phase 1. The computation time for Phase 1 ranges form 2 to 45 seconds. Now 30 or more tasks remain uncovered for Ac_A, Ac_B, Ht_A, and Ht_B. Recall from Table 3.1 that in these instances more than 50 original duties have become infeasible. In Phase 2 the number of uncovered tasks could be reduced for all instances but Lls_B. However, there is no instance where all tasks are covered after Phase 2. For most instances we find better results with setting *SET1*. An exception is Ztm_A where the solution found using setting *SET2* is much better. The computation time for this setting ranges from 2 to 4794 seconds. With setting *SET2* the computation time is typically much smaller. The longest computation time with this setting is 305 seconds for Ac_A. For both settings we observe that the 2P-RSPPRC heuristic fails to produce good solutions if no reserve duties are available. This is especially true for the difficult instances Ac_A, Ac_B, Ht_A, and Ht_B. In Section 4.5 we will draw our final conclusions after comparing the 2P-RSPPRC heuristic with the CGDDS approach and a heuristic branch-and-price approach (see Section 4.4).

	Phase 1				Phase 2							
					SET1				*SET2*			
Id	Obj	A-B	A-A	T (s)	Obj	A-B	A-A	T (s)	Obj	A-B	A-A	T (s)
Ac_A	619612	28	15	36	331682	13	18	4794	423816	18	16	305
Ac_B	410394	17	17	45	324883	13	14	2809	349864	14	16	246
Bl_A	84551	3	6	3	83905	3	6	216	83905	3	6	39
Bl_B	101492	4	5	2	81753	3	5	151	82498	3	5	30
Ht_A	354796	14	16	33	205493	7	13	1891	263285	10	12	206
Ht_B	360774	13	23	34	190018	5	19	1689	196628	5	21	250
Lls_A	122146	5	3	3	63375	2	3	1665	103110	4	3	53
Lls_B	43548	1	2	20	43548	1	2	2	43548	1	2	2
Ztm_A	256296	12	4	23	122778	5	6	1426	96880	4	4	136
Ztm_B	155887	7	3	12	26926	0	6	580	39410	1	4	62

Table 4.2: Results of the 2P-RSPPRC heuristic without reserve duties.

4.3 Dynamic constraint aggregation

Dynamic constraint aggregation (DCA) is an advanced column generation method for large-scale vehicle and crew scheduling applications. The conceptual idea was presented in the thesis of Villeneuve (1999). The first implementation was provided by Elhallaoui et al. (2005). For crew scheduling applications, DCA is motivated by the observation that in optimal solutions duties are composed of several clusters of tasks, where a cluster corresponds to consecutive tasks on the same vehicle. For many applications of rescheduling on the day of operations the new schedules should not deviate too much from the planned schedules, like it is the case for the OCRSP we consider in this thesis. This means that one expects to find that duties in a rescheduling solution should be composed of clusters of tasks that also appeared in the planned duties.

The idea behind DCA is to not treat every task on its own by a set partitioning constraint in the mathematical model, but to aggregate the tasks in one cluster and represent them in the mathematical model by one aggregated constraint. The resulting model is referred to as the *aggregated* master problem. During the algorithm only a subset of the columns is considered in the aggregated restricted master problem (ARMP). Since the number of constraints in the ARMP is much smaller, it can be solved quicker. The aggregation is dynamically redefined during the solution procedure.

Elhallaoui et al. (2008) present an enhancement of the original DCA method called multi-phase DCA method. In this method a multi-phase partial pricing strategy is used to reduce overall computational times. The motivation behind the multi-phase partial pricing is as follows. The redefinition of the aggregation should be based solely on columns which are only slightly incompatible with the current aggregation in order to keep the series of aggregations considered during the algorithm more similar. Given an aggregation, the incompatibility score of a column is equal to the number of additional clusters that are needed for the column to become compatible. In phase h only columns with an incompatibility score less than h are considered, and hence only these need to be generated in the pricing problems.

4.3.1 The mathematical model

In order to apply DCA we formulate the OCRSP as a set partitioning problem with side constraints. We use the same notation as in Section 3.3 and replace the "$\geq$" sign in Formulation (3.1)–(3.4) with a "$=$" sign. We then obtain the same formulation that was used in Rezanova and Ryan (2010) which we will refer to as OCRSP-SP.

$$\min \quad \sum_{\delta\in\Delta}\sum_{k\in K^\delta} c_k^\delta x_k^\delta + \sum_{i\in N} f_i z_i \tag{4.1}$$

$$\text{s.t.} \quad \sum_{\delta\in\Delta}\sum_{k\in K^\delta} a_{ik}^\delta x_k^\delta + z_i = 1 \qquad \forall i \in N \tag{4.2}$$

$$\sum_{k\in K^\delta} x_k^\delta = 1 \qquad \forall \delta \in \Delta \tag{4.3}$$

$$x_k^\delta, z_i \in \{0,1\} \qquad \forall \delta \in \Delta, \forall k \in K^\delta, \forall i \in N \tag{4.4}$$

Constraints (4.2) make sure that every task is either covered by exactly one feasible completion or is canceled. The assignment constraints (4.3) are the same as Constraints (3.3).

The optimal solution value of both formulations is exactly the same under the considered objective function. Every feasible solution of Formulation (3.1)–(3.4) can be transformed into a solution with the same objective value that is feasible w.r.t. (4.2)–(4.4). This can be seen as follows. Let us consider a solution of Formulation (3.1)–(3.4), where $\sum_{\delta\in\Delta}\sum_{k\in K^\delta} a_{ik}^\delta x_k^\delta + z_i$ is > 1 for some $j \in N$. This means that 2 or more feasible completions with $x_k^\delta = 1$ cover task j. Let us refer to these feasible completions as $\hat{X}$. In order to obtain a solution to Formulation (4.2)–(4.4), we have to decide in which feasible completion the task should appear as driving task and in which it should appear as deadheading task. Note that for every feasible completion $k \in \hat{X}$ there exists a feasible completion l where $a_{il}^\delta = a_{ik}^\delta \forall i \neq j \in N$. Moreover, $c_l^\delta = c_k^\delta$ if task j was not a driving task in the original duty δ, and $c_l^\delta \geq c_k^\delta$ if task j was a driving task in δ. Therefore, the only interesting case for the decision about driving and deadheading is when task j was a driving task in one of the original duties δ for a $k \in \hat{X}$. In order to not change the objective function we have to the decisions in this case as follows. Task j must be the driving task in the feasible completion of the original duty δ in which it was a driving task and a deadheading task in the other feasible completions.

4.3.2 Outline of the DCA method

In Algorithm 5 we show the pseudo-code of the multi-phase DCA method to solve a linear relaxation of e.g. the OCRSP-SP. An initial aggregation for the problem needs to be provided as input (see Line 1). Next, the phase h is set to 0 in Line 2. Then DCA is performed considering only columns that have an incompatibility score less or equal to h (Lines 5–13). After solving the ARMP (Line 6), a disaggregated dual vector is computed in Line 7 by solving an auxiliary shortest path problem (see Elhallaoui et al. (2005) for details). The pricing problems return negative reduced cost columns that are at most h incompatible with the current aggregation, if any exist (Line 8). If no such columns exist, the algorithm moves on to the next phase in Line 13. If such negative reduced cost columns have been found, in Line 11 the aggregation may be updated and columns are added to the ARMP. The decision about a partition update depends on the reduced cost of the negative reduced cost columns that are returned by the pricing problems. Let $\bar{c}$ be the smallest reduced cost of a negative reduced cost column that is compatible with the current partition and let $\tilde{c}$ be the smallest reduced cost of a negative reduced cost column that is not compatible with the current partition. In the case that compatible as well as incompatible columns with negative reduced cost have been found, the partition is updated if $\bar{c}/\tilde{c} \leq \textit{thresholdPartitionUpdate}$. Moreover, the partition is not updated if all negative reduced cost columns are compatible and the partition is always updated if all negative reduced cost columns are incompatible. After testing some values between 0 and 1 we decided to set $\textit{thresholdPartitionUpdate} = 0.67$ for our experiments.

1 Determine an initial aggregation;
2 Set $h = 0$;
3 **while** $\textit{stopMDCA} = \textit{false}$ **do**
4     $\textit{stopColGen}$ = false;
5   **while** $\textit{stopColGen} = \textit{false}$ **do**
6   Solve the ARMP to obtain a primal and a dual solution;
7   Compute a disaggregated dual vector;
8   Solve the pricing problems with the disaggregated dual vector;
9   **if** $\exists$ *negative reduced cost columns with an incompatibility score* $\leq h$ **then**
10   Update the aggregation and the ARMP if necessary;
11   Add columns to the ARMP;
12   **else**
13             $\textit{stopColGen}$ = true;
14   **if** $h < \textit{maxPhase}$ **then**
15   Set $h = h + 1$ and move to next phase;
16   **else**
17         $\textit{stopMDCA}$ = true;

Algorithm 5: The multi-phase dynamic constraint aggregation (DCA) method

4.3.3 The initial clusters

An initial aggregation needs to be specified as input for the DCA method. Recall that one motivation behind using DCA is that we expect that many clusters of tasks present in the planned duties should also turn up in an optimal solution. This is especially true for original duties which are still feasible, since given our objective function we try to avoid changing them. Therefore, we construct one or two clusters from a still feasible original duty. Due to the available DCA implementation, we cannot have a meal break within a cluster. So if necessary, we construct one cluster including all tasks before and one cluster including all tasks after the planned meal break.

For original duties that are not feasible anymore, we cluster the tasks as follows. We check for every driving task t_i, or its rerouted substitute, if it is still possible to perform the next planned driving task t_j, if the meal break was not planned between the two tasks, and if there is no rolling stock change between the two tasks. If these three conditions are satisfied, then we add t_j to the cluster of t_i.

In Figure 4.2 we show the four clusters, c_1, c_2, c_3 and c_4, we constructed from the tasks that had been assigned to the original duties “Ehv 122” and “Mt 107”.

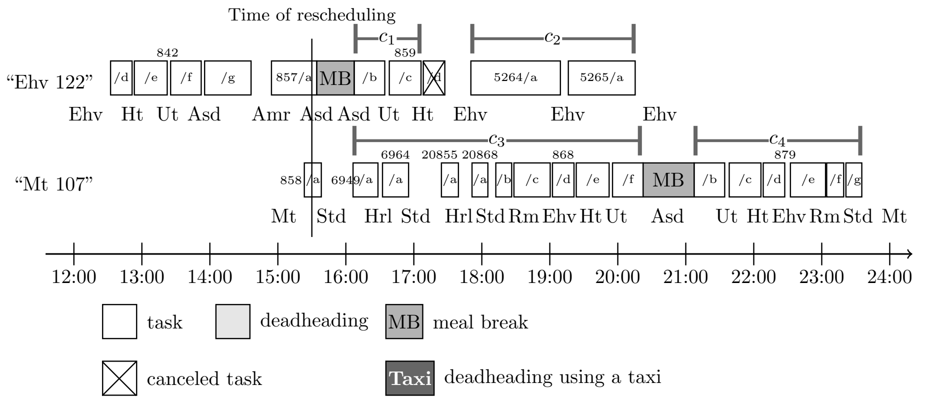


Figure 4.2: The initial clusters c_1, c_2, c_3 and c_4 are generated from the affected original duty “Ehv 122” and the not affected original duty “Mt 107” in case Ht_2.

We also tried two other ways of generating the initial clusters, but we obtained the better results for the initial clusters as described above. The first alternative way generates less initial clusters. For original duties that are feasible we generate the initial clusters as described above. For the infeasible original duties we do not consider rolling stock changes. In the second alternative which generates a larger number of initial clusters we applied the rules that we described above for infeasible original duties, to all original duties.

Treatment of the assignment constraints

So far we have only discussed how to cluster the set partitioning constraints that correspond to tasks. However, constraints (4.3) in Formulation (4.1)–(4.4) are also set partitioning constraints and can also be part of clusters in the aggregated model. Assuming that many duties that are still feasible will not be changed, it seems promising to add the duty constraints to one of the clusters that have been constructed from the original duty. One could choose between two options, either add the duty constraint for duty δ to the cluster of the task the driver has to perform next, or add the duty constraint to the cluster of the last task the driver has to perform. We have chosen the second option for the following reason. It seems more likely that, if a duty will be modified, this modification will take place relatively soon after the time of rescheduling, because that is where the conflicts in the planned crew schedule are likely to be. Moreover, given that it is desirable to return to the planned duty, it is likely that in a duty, even if it is modified, the last task does not change. This means that we hope that the initial cluster does not have to be broken between the constraint for the last task and the duty constraint during the DCA method. We did a number of experiments with the duty constraints added to the initial clusters. In most cases this resulted in longer computation times for solving the LP-relaxation and therefore we decided not to use this option in our final experiments.

4.3.4 The pricing problems

A connection graph for the pricing problems is built for every original duty. A few adjustments compared to the graphs described in Section 3.4.2 have been made in order to be able to use the standard resource constraint shortest path solver available in *GENCOL*. GENCOL is a column generation library developed at the *GERAD* (Group for Research in Decision Analysis). The main features are discussed in Desaulniers et al. (1998). In order to model the meal break rule, we use two resources *break* and *timespan*. For every node in the graph we associated resource windows with these resources. The resource windows are the same for every node, namely [0 1] for *break* and [0 330] for *timespan*. The consumption for the resource *break* is 1 for *break arcs* and 0 for all other arc types. For the resource *timespan* the consumption of an arc r_{ij} is equal to the duration of the related activity, or -330 for arcs corresponding to meal breaks. The resource extension function for the resource *break* simply adds the consumption of an arc r_{ij} to the total consumption of the partial path R_i. The resource extension function for the resource *timespan* is given by $R_j = \max\{0, R_i + r_{ij}\}$. In Figure 4.3 we show a part of the graph used in the pricing problem for driver “Gn 7”. Note that the feasible completions shown in Figure 2.6 correspond to paths in Figure 4.3.

The weight of arc (i, j) consists of a penalty as described in Section 3.4.2, from which the dual value of an associated constraint is subtracted. Constraints (4.2) are associated with *task* arcs and constraints (3.3) are associated either with all arcs leaving the source, or with all arcs entering the sink.

The current DCA implementation in *GENCOL* does not support arcs parallel to arcs that correspond to constraints in a cluster, like the *task* arcs. Therefore, we introduced *dummy* task *departure* and *arrival* nodes and used them as tail, respectively head nodes for the *deadhead* arcs.

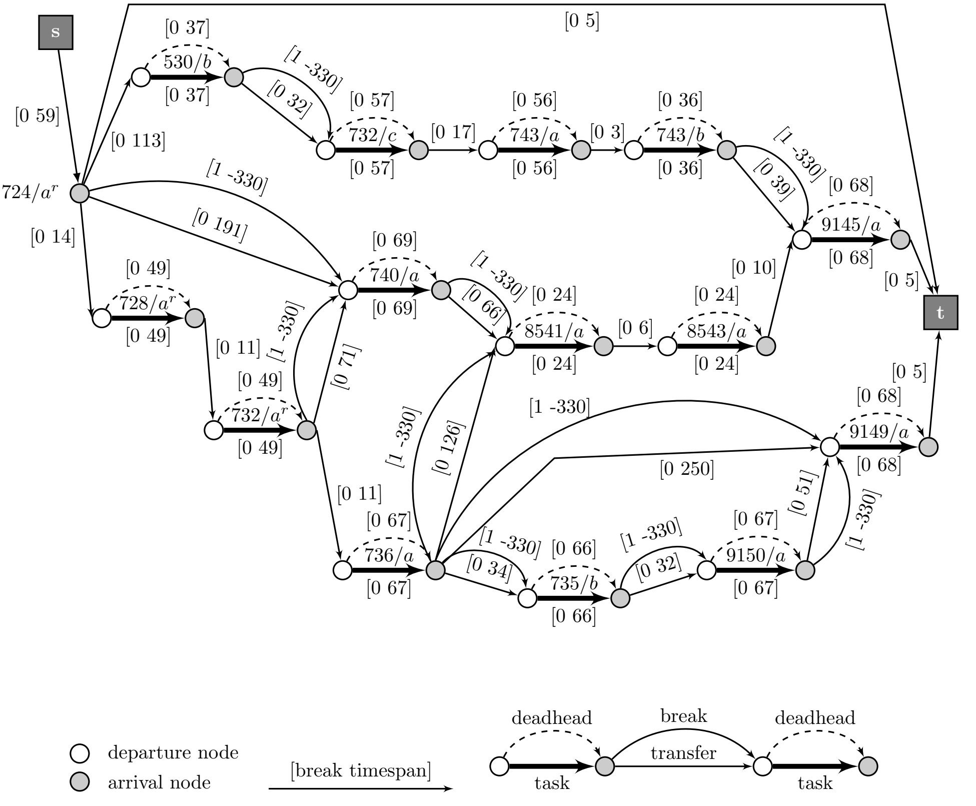


Figure 4.3: A part of the pricing problem graph with resource consumptions for driver "Gn 7".

4.3.5 Computational results

We will compare the DCA heuristic to a branch-and-price heuristic (HBNP) also using version 4.5.1 of the *GENCOL* column generation library and *CPLEX* 10.1 for solving the RMPs and ARMPs. In order to study the performance of the DCA heuristic we are going to create larger core problems for the subset of the set of 10 disruptions where HBNP does not find a solution covering all tasks in the solution to the initial core problems as defined in Section 3.5.1. For the initial core problems the computation times are so small that we did not expect large improvements from using DCA. This was confirmed by computational experiments.

Recall that an initial core problem is given by a subset of original duties $\bar{\Delta}$ and tasks $\bar{N}$. Let $\bar{T} = [t_0, t_1 + 60 \text{ minutes}]$ be the interval based on the rerouted and canceled tasks due to the disruption. Furthermore, let $\vec{T}(\delta)$ be the remaining time duty δ is available after the time of rescheduling. The larger core problems, we will refer to them as *medium* and *large*, are made by adding original duties and tasks to the initial core problems by the following rules.

- Add original duty δ' if its crew base $b_{\delta'}$ is the same as for any $\delta \in \bar{\Delta}$ and $\bar{T}$ is inside $\vec{T}(\delta')$.
- Add original duty δ' if its crew base $b_{\delta'}$ is the same as for any $\delta \in \bar{\Delta}$ and the time $\vec{T}(\delta')$ and $\hat{T}$ overlap $\geq 0.5 \cdot \vec{T}(\delta')$.

Note that by this definition the original duties and tasks in an initial core problem are a subset of the duties and tasks of a medium core problem which are a subset of the duties and tasks of a large core problem. We tested on an *Intel Quad core* machine with 4 GB RAM clocked at 2.83 GHz.

All tables in this section will have the same columns whose meanings are as follows. The name of the instance is given in the first column. The first number in an instance name is the number of original duties, and the second number is the number of set partitioning constraints for the tasks. Column *LP* shows the value of the LP-relaxation. *IP* is the value of the best feasible solution. The percentage gap is given in column *GAP*. *Nodes* is the number of nodes that have been explored in the branch-and-bound tree. *TR* shows the computation time in seconds needed to solve the root node. *TT* shows the total computation time in seconds. *A-B* and *A-A* are the number of tasks of type A-B and A-A respectively that are not covered by any feasible completion.

In Table 4.3 we present the results for the 10 disruptions using HBNP. If tasks could not be covered in the solution we also solved the medium and large core problems. First of all we observe that the LP-relaxation of the initial core problems could be solved in less than 100 seconds for all disruptions. The number of nodes in the branch-and-bound tree that have been solved ranges from 1 for Bl_A, Bl_B, Lls_B, and Ztm_B to 33 for Ac_A. The gap between the LP-Relaxation and the integer solutions is at most 2.5% for all instances and all core problems. Furthermore, we observe a rapid increase in computation time, for the LP-relaxation as well as the total computation time, for the medium and large core problems. For example for Ht_B and Ztm_B the number of set covering constraints in the large core problems is less than twice the number in the medium core problems while the time needed to solve the LP-relaxation increases by a factor of more than 10.

Instance	LP	IP	GAP (%)	Nodes	TR (s)	TT (s)	A-B	A-A
Ac_A_163_602	63107.8	63808	1.1	33	26	121	1	0
Ac_A_233_1074	42426.3	42712	0.7	21	196	730	0	0
Ac_A_318_1567	41441.7	41908	1.1	21	824	2969	0	0
Ac_B_143_734	35814.7	36339	1.5	21	92	328	0	0
Bl_A_80_234	10581.0	10581	0.0	1	2	2	0	0
Bl_B_63_186	9532.0	9532	0.0	1	1	1	0	0
Ht_A_133_614	37916.7	38136	0.6	15	47	124	0	0
Ht_B_119_665	62906.0	62922	0.0	2	96	103	1	1
Ht_B_192_1243	40076.0	40126	0.1	14	815	1035	0	1
Ht_B_289_2060	38699.0	39336	1.6	15	8195	13170	0	1
Lls_A_81_239	17623.0	17651	0.2	3	2	3	0	0
Lls_B_161_456	21796.0	21796	0.0	1	10	10	0	0
Ztm_A_112_508	11846.7	12139	2.5	4	20	28	0	0
Ztm_B_110_423	34736.0	34736	0.0	1	16	16	1	0
Ztm_B_178_853	14485.5	14537	0.4	2	171	188	0	0
Ztm_B_309_1577	11687.5	11739	0.4	2	2137	2249	0	0

Table 4.3: Results for heuristic branch-and-price (HBNP) with 46 reserve duties.

Instance	LP	IP	GAP (%)	Nodes	TR (s)	TT (s)	A-B	A-A
Ac_A_233_1074	42426.3	42859	1.0	14	429	1363	0	0
Ac_A_318_1567	41441.7	42463	2.5	31	1084	8910	0	0
Ht_B_192_1243	40076.0	40178	0.3	4	579	1227	0	1
Ht_B_289_2060	38699.0	38980	0.7	17	2642	10957	0	1
Ztm_B_178_853	14485.5	14536	0.3	2	215	268	0	0
Ztm_B_309_1577	11687.5	11739	0.4	2	1211	1569	0	0

Table 4.4: Results for heuristic dynamic constraint aggregation (DCA) with 46 reserve duties.

The results for the medium and large core problems for Ac_A, Ht_B, and Ztm_B with DCA are shown in Table 4.4. For the medium core problems we observe that the time for solving the LP-relaxation is once shorter (Ht_B) and twice longer (Ac_A and Ztm_B) as with HBNP. For the large core problems DCA solves the LP-relaxation in less time for Ht_B and Ztm_B. Remarkable is the improvement for Ht_B where the LP-relaxation is solved in 2642 seconds with DCA as opposed to 8195 seconds with HBNP. Moreover, we see that the total solution time is only reduced from 13170 seconds without DCA to 10957 seconds with DCA while about the same number of nodes have been explored. A similar observation can be made for Ztm_B.

Instance	LP	IP	GAP (%)	Nodes	TR (s)	TT (s)	A-B	A-A
Ac_A_117_602	71206.2	76152	6.9	41	23	125	1	2
Ac_A_187_1074	48019.1	50493	5.2	53	229	1805	0	1
Ac_A_227_1567	43753.0	45280	3.5	33	936	4652	0	0
Ac_B_111_734	37305.6	38410	3.0	35	100	465	0	0
Bl_A_34_234	11515.5	11590	0.6	2	2	2	0	0
Bl_B_31_186	13114.5	14543	10.9	7	1	3	0	1
Bl_B_120_933	10682.7	10849	1.6	6	590	739	0	0
Bl_B_178_1420	10682.7	10849	1.6	8	2172	3337	0	0
Ht_A_87_614	38934.8	39998	2.7	26	41	156	0	0
Ht_B_83_665	70337.7	70376	0.1	7	84	137	1	3
Ht_B_156_1243	43548.2	43584	0.1	4	601	790	0	2
Ht_B_253_2060	42356.0	43291	2.2	20	7136	14508	0	2
Lls_A_35_239	20574.5	20850	1.3	2	2	3	0	0
Lls_B_115_456	23126.3	23955	3.6	4	10	19	0	0
Ztm_A_66_508	12290.0	12290	0.0	1	17	17	0	0
Ztm_B_64_423	34936.0	34936	0.0	1	16	16	1	0
Ztm_B_132_853	14763.5	15038	1.9	7	229	309	0	0
Ztm_B_263_1577	11687.5	11738	0.4	2	2080	2200	0	0

Table 4.5: Results for heuristic branch-and-price (HBNP) without reserve duties.

We also considered the 10 disruptions in the case that no reserve duties would be available. The results for HBNP are shown in Table 4.5. The solution times for the LP-relaxations are at most 100 seconds for the initial core problems. The percentage gap of the integer solutions is, with the exception of Ztm_A, higher than in the case with 46 reserve duties. Next to Ac_A, Ht_B, and Ztm_B also for Bl_B not all tasks are covered in the solution found for the initial core problem. Comparing the increase in problem size and solution time between the initial, the medium, and the large core problems, we observe that the increase in computation time is more than linear.

In Table 4.6 we report the results for solving the medium and larger core problems with DCA. As for the case with reserve duties, DCA needs more time to solve the LP-relaxation of Ac_A. For the large core problem for Bl_B, HBNP and DCA perform roughly the same in all aspects. For Ht_B and Ztm_B, DCA produces similar or better feasible solutions in less time. While the improvement is only small for the medium core problems it is remarkable for the large core problems. Especially the LP-relaxation is solved much faster with DCA, 2883 and 1098 seconds as opposed to 7136 and 2080 seconds for Ht_B and Ztm_B respectively. However, the reduction in the total computation times for these instances is less impressive.

Instance	LP	IP	GAP (%)	Nodes	TR (s)	TT (s)	A-B	A-A
Ac_A_187_1074	48019.1	48783	1.6	28	350	1857	0	1
Ac_A_227_1567	43753.0	44435	1.6	37	998	10059	0	0
Bl_B_120_933	10682.7	10946	2.5	4	345	461	0	0
Bl_B_178_1420	10682.7	10849	1.6	8	2090	3234	0	0
Ht_B_156_1243	43548.2	43584	0.1	4	433	644	0	2
Ht_B_253_2060	42356.0	42630	0.6	11	2883	7693	0	2
Ztm_B_132_853	14763.5	14837	0.5	6	169	290	0	0
Ztm_B_263_1577	11687.5	11788	0.9	4	1098	1519	0	0

Table 4.6: Results for heuristic dynamic constraint aggregation (DCA) without reserve duties.

General observations

We can state some general observations about our experiments with DCA and we will try to relate our observations to results stated earlier in the literature. During the solution process of the LP-relaxation the initial partition is almost completely disaggregated. Moreover, the number of fractional variables in the solution of the LP-relaxation is about the same compared to HBNP. In Elhallaoui et al. (2008) the authors report that for simultaneous bus and driver scheduling problems the number of fractional variables was reduced significantly. A similar observation was found for bidline scheduling for airlines by Boubaker et al. (2010).

While the LP-Relaxation could be solved much faster witch DCA as compared to HBNP, this improvement was partly lost when exploring the nodes in the branch-and-bound tree. When we look at the average time spent per node except the root node (TT-TR/(Nodes−1)), we see that this time is higher for all but one instance for DCA. The only exception is the large core problem for Bl_B without reserve duties (Bl_B_178_1420). An intuitive explanation is that in a node other than the root node, it is faster to just work with the RMP instead of starting with an aggregated partition in the ARMP which is completely disaggregated in the column generation process anyway. Given the current implementation of DCA we would have probably obtained better results by using DCA only for the root node and using standard column generation without aggregation in all other nodes.

4.4 Comparison

Now we are going to compare the tested methods 2P-RSPPRC, HBNP/DCA, and CGDDS. Therefore we present in Table 4.7 for each instance with 46 reserve duties one result per method from Table 4.1 for 2P-RSPPRC, from Tables 4.3 and 4.4 for HBNP/DCA, and from Tables 3.2 and 3.5 for CGDDS respectively. For the 2P-RSPPRC heuristic and for HBNP/DCA this is the result with the minimum weighted number of uncovered tasks, where the weights are chosen according to the penalty cost associated with not covering a task (20,000 for type A-B and 3,000 for type A-A). If this number is the same for two results, we pick the solution that has been obtained in less computation time. For example, for Ht_B we present the result that has been obtained using the medium size core problem and HBNP. In the same way we present the results for the instances without reserve duties in Table 4.8. The corresponding source tables are Tables 4.1, 4.5, 4.6, 3.4, 3.7, and 3.8.

The columns in Tables 4.7 have the following meaning. The first column *Id* is the code of the instance. Then for each of the three methods we present four columns namely *Obj*, *A-B*, *A-A*, and *T*. *Obj* is the value of the objective function of the solution. *A-B* and *A-A* show the number of tasks of types A-B and A-A that are uncovered in the solution. Finally, column *T* displays the computation time in seconds.

Comparing the results presented in Table 4.7 we observe that HBNP/DCA as well as CGDDS find a solution without uncovered tasks for 9 out of 10 instances with 46 reserve duties. The 2P-RSPPRC heuristic finds such a solution only for 3 out of 10 instances. Even though all tasks are covered, the solutions produced by 2P-RSPPRC have much higher objective values. To a large extent this can be explained by the fact that more reserve duties are used in these solutions. Interesting is that for instance Ht_B HBNP/DCA finds a solution where all tasks of type A-B are covered. This solution was found in 1035 seconds when solving the medium size core problem. CGDDS does not find a solution where all A-B tasks are covered. The 2P-RSPPRC heuristic is not competitive with the column generation based methods HBNP/DCA and CGDDS since it only finds good solutions for the problems with the lowest number of infeasible original duties.

We present the results for the 10 instances without reserve duties in Table 4.8. The columns show the same information as in Table 4.7. Again the solutions of HBNP/DCA cover all tasks in all instances except Ht_B. However, for Ac_A such a solution was only found when solving the large core problem and with HBNP this took 4652 seconds. Using CGDDS we find solutions covering all tasks for 9 out of 10 instances. The 2P-RSPPRC heuristic performs badly, for none of the 10 instances it could find a solution covering all tasks.

4.5 Conclusion

In this chapter we have presented the 2P-RSPPRC heuristic mimicking the manual rescheduling process of crew dispatchers. Moreover, we have investigated the effect of

	2P-RSPPRC				HBNP/DCA				CGDDS			
Id	Obj	A-B	A-A	T (s)	Obj	A-B	A-A	T (s)	Obj	A-B	A-A	T (s)
Ac_A	149651	5	7	975	42712	0	0	730	45412	0	0	236
Ac_B	135824	5	3	374	36339	0	0	328	36134	0	0	255
Bl_A	21070	0	1	23	10581	0	0	2	10581	0	0	26
Bl_B	13852	0	0	3	9532	0	0	1	9532	0	0	18
Ht_A	76704	1	5	1225	38136	0	0	124	38094	0	0	193
Ht_B	68981	0	9	137	40126	0	1	1035	62922	1	1	209
Lls_A	27811	0	0	5	17651	0	0	3	17646	0	0	33
Lls_B	54309	1	2	31	21796	0	0	10	21796	0	0	107
Ztm_A	16777	0	0	33	12139	0	0	28	11935	0	0	87
Ztm_B	37714	1	0	55	14537	0	0	188	15237	0	0	83

Table 4.7: Comparison of results with minimum number of uncovered tasks obtained with 2P-RSPPRC, HBNP/DCA, and CGDDS with 46 reserve duties.

	2P-RSPPRC				HBNP/DCA				CGDDS			
Id	Obj	A-B	A-A	T (s)	Obj	A-B	A-A	T (s)	Obj	A-B	A-A	T (s)
Ac_A	331682	13	18	4830	45280	0	0	4652	53220	0	0	288
Ac_B	324883	13	14	2854	38410	0	0	465	38919	0	0	261
Bl_A	83905	3	6	42	11590	0	0	2	11590	0	0	20
Bl_B	82498	3	5	32	10849	0	0	461	13406	0	0	73
Ht_A	205493	7	13	1924	39998	0	0	156	39799	0	0	206
Ht_B	190018	5	19	1723	43584	0	2	664	64084	1	1	553
Lls_A	63375	2	3	1668	20850	0	0	3	21698	0	0	20
Lls_B	43548	1	2	22	23955	0	0	19	23752	0	0	106
Ztm_A	96880	4	4	159	12290	0	0	17	12290	0	0	40
Ztm_B	26926	0	6	592	14837	0	0	290	15439	0	0	62

Table 4.8: Comparison of results with minimum number of uncovered tasks obtained with 2P-RSPPRC, HBNP/DCA, and CGDDS without reserve duties.

considering more duties and tasks in so called medium and large core problems and compared the outcomes to the outcomes of the CGDDS method. Furthermore, we have compared HBNP and DCA for solving the medium and large core problems.

We can summarize the evaluation of the different solution approaches for railway crew rescheduling as follows. First of all, the 2P-RSPPRC heuristic is not competitive with the other approaches, since for most instances it fails to produce good solutions. Secondly, we have seen that for 5 out of 7 large core problems the computation time of DCA is smaller compared to HBNP while the solutions have approximately the same quality. Finally, for some difficult instances, considering larger core problems enables significantly better solutions compared to the CGDDS method. However, the computation time needed to find these solutions with HBNP and DCA is currently too high for practical purposes.

With additional research into different neighborhood definitions for CGDDS we could try to close the gap between the solutions currently found with CGDDS and the solutions found when considering larger core problems. From the evaluation of the different solution approaches we conclude that CGDDS is the best method available to solve practical rescheduling instances as considered in this thesis. It consistently finds solutions of good quality within a reasonable amount of computation time.

Chapter 5

Railway Crew Rescheduling with Retiming

5.1 Introduction

Passenger railway operators face unforeseen events such as infrastructure malfunctions, accidents or rolling stock breakdowns, that make it impossible to operate the timetable as planned, every day. As described in Chapter 2, the disruption management process consists of the accomplishment of three interconnected tasks: (1) timetable adjustment, (2) rolling stock rescheduling, and (3) crew rescheduling. 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 chapter 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 chapter, which is based on Veelenturf et al. (2009), we look at an extension of the crew rescheduling problem, where some timetabling decisions are integrated into crew rescheduling. More precisely, the departure of trains may be delayed. This gives more flexibility to the third step in the disruption management process and may avoid undesired iterating. Moreover, this new approach is able to provide high quality solutions from a service level point of view.

The first contribution of this chapter is a new formulation for crew rescheduling with retiming, where retiming options are modeled as discrete choices. Moreover, we show how to adapt the solution approach presented in Chapter 3 in order to keep the increase in computation time for the extended model moderate. We evaluate our approach using real life data from NS. Finally, we show that retiming allows us to find better solutions compared to crew rescheduling without retiming.

The remainder of this chapter is organized as follows. A problem description is provided in Section 5.2. The existing literature is reviewed in Section 5.3. In Section 5.4 we present the mathematical formulation. Our solution approach is discussed in Section 5.5. Computational results are presented in Section 5.6. In Section 5.7 we draw some conclusions and give some recommendations for further research.

5.2 Problem description

We first introduce some railway terminology which is necessary to clearly describe the problem.

Recall from Section 2.5.1 that the operational crew rescheduling problem (OCRSP) takes an adjusted timetable and modified rolling stock schedule as input and tries to find a replacement duty for every original duty, such that as many tasks as possible of the adjusted timetable are covered.

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 has been found. The railway operator has to come up with another adjusted timetable, for which it is possible to find a compatible crew schedule.

The idea of allowing retiming is to evaluate not just one fixed timetable but a number of similar timetables at once. By delaying the departure of some tasks more connections for drivers will become possible and hence more feasible completions may exist. Therefore, it may be possible to find a better crew schedule. Compared to classical crew rescheduling, the objective of the extension with retiming also aims at keeping the amount of delay as small as possible.

Example 5.1

In Figure 5.1.a we show the original duty Ah 114 from crew base Arnhem in case the two southbound routes from 's-Hertogenbosch to Breda and Eindhoven are blocked from 15:30 to 18:30. The duty has started with driving task 3043/*e (the fifth task of train 3043) from Arnhem (Ah) to Nijmegen (Nm). At 15:30, when the disruption starts, the driver has completed his next two tasks and is performing task* 3653/*b. The meal break was planned in Roosendaal, thereafter the duty was supposed to end with driving train 3666 from Roosendaal to Arnhem,* 3666/*a*–3666/*d. However, due to the route blockage, task*

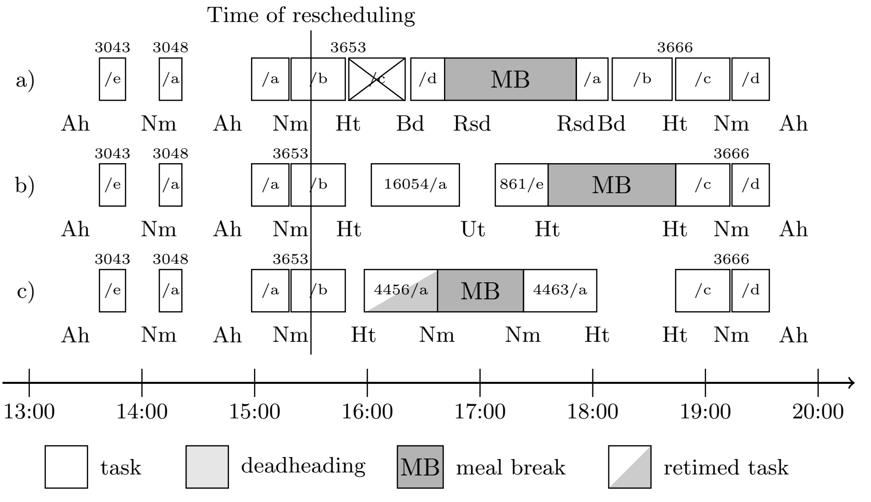


Figure 5.1: The infeasible original duty and two possible replacement duties for duty "Ah 114" in Example 5.1.

3653/*c is canceled. Therefore, original duty Ah 114 has become infeasible. A replacement duty is shown in Figure 5.1.b. Note that because the rescheduling takes place at 15:30, the first four tasks of the duty cannot be changed. After those 4 tasks, the driver arrives in 's-Hertogenbosch at 15:48. If the next task has to be performed on different rolling stock, a minimal transfer time of 10 minutes must be respected. So the replacement duty is allowed to perform task* 16054/*a to Utrecht at 16:02, which is operated with different rolling stock unit than task* 3653/*b. From Utrecht the driver could go back to 's-Hertogenbosch by driving task* 861/*e. After that the duty could end by performing tasks* 3666/*c and* 3666/*d just as in the original duty.*

The motivation for allowing retiming is to make replacement duties possible that are not possible in a fixed timetable. For example, the planned departure time of task 4456/a is 15:56 and the task is operated by a different rolling stock than task 3653/b, which means that due 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. In Figure 5.1.c we show a replacement duty, not feasible without retiming, where tasks 4456/a is delayed by 2 minutes.

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

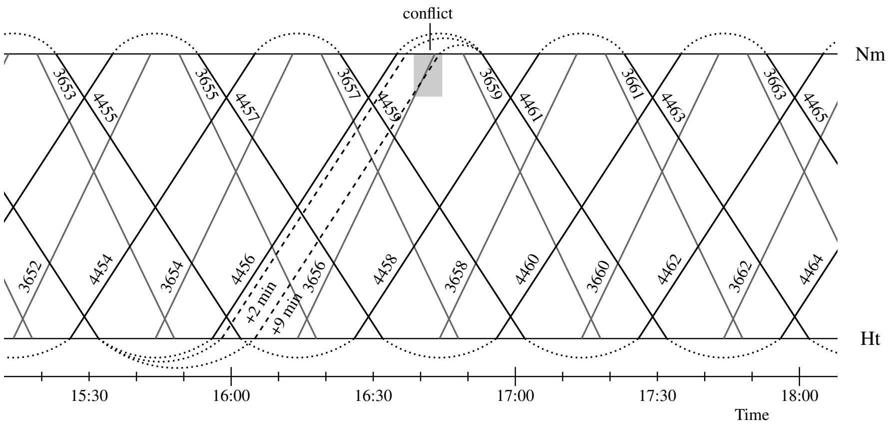


Figure 5.2: An example of a delayed task between 's-Hertogenbosch (Ht) and Nijmegen (Nm).

(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.

The first assumption is not always true in practice. On the one hand, the planned running time for a task may include some buffer time that could be utilized to (partly) absorb delays. On the other hand, a task that is running later than planned could experience an additional delay due to conflicts with other trains. For example, it might be possible that a delayed train has to wait at a signal in a station area. Conversely, a delayed task may affect other trains. A faster train may, for example get stuck behind a slower delayed train. Figure 5.2 shows part of the 2007 timetable for the route between 's-Hertogenbosch and Nijmegen. 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 by e.g. 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, as it is in this situation, the intercity train will be stuck behind the regional train and experience a delay. This example shows that assumption (ii) does not always hold. However, at this point in time it seems reasonable since the objective of this chapter is to analyze the potential retiming in crew rescheduling might offer. In practice there are also cases, especially when the delay is small, where assumptions (i) and (ii) hold, e.g. if train 4456 would be delayed by 2 minutes.

5.3 Literature review

While Walker et al. (2005) was the first paper looking at railway crew rescheduling, in the domain of airlines, crew rescheduling received the first attention much earlier in Johnson et al. (1994). Note that crew rescheduling is also know as crew recovery. For a recent review of literature on airline crew rescheduling we refer the interested reader to Clausen et al. (2010). Stojković and Soumis (2001) and Abdelghany et al. (2004) are the first papers that extend crew rescheduling by the possibility of retiming 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 2.5.1 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 aircraft routing, crew scheduling and flight retiming model.

For a literature review of crew rescheduling without retiming we refer to Section 3.2.

5.4 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:

$$L_e = \{f \in E_{r(i)} \setminus \bigcup_{e' \in \check{E}_e} L_{e'} |\, (t^{\text{dep}}_{r(i)} + d_f) - (t^{\text{arr}}_i + d_e) \geq h_i, \forall e' \in \hat{E}_e \;\; (t^{\text{dep}}_{r(i)} + d_f) - (t^{\text{arr}}_i + d_{e'}) < h_i\} \tag{5.1}$$

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,

$$B_e = \{e' \in E_{i(e)} \mid d_{e'} \leq d_e\} \tag{5.2}$$

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:

$$z_i + \sum_{e' \in B_e} v_{e'} - \sum_{e' \in L_e} v_{e'} \geq 0 \quad \forall i \in N^c : r(i) \neq 0, \forall e \in E_i \tag{5.3}$$

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 5.1 illustrates the definition of L_e (see (5.1)). 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 5.1. Let us assume that $h_l = h_m = 2$ minutes. Then according to (5.1): $L_d = \{e\}$, $L_{d'} = \{e'\}$, $L_e = \emptyset$ 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

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

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 margins in the timetable. Then Constraints (5.3) will become:

$$z_l + v_d - v_e \geq 0 \tag{5.4}$$

$$z_l + v_d + v_{d'} - v_{e'} \geq 0 \tag{5.5}$$

$$z_m + v_e + v_{e'} - v_f \geq 0 \tag{5.6}$$

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

$$\min \quad \sum_{\delta\in\Delta}\sum_{k\in K^\delta} c_k^\delta x_k^\delta + \sum_{i\in N} f_i z_i + \sum_{e\in E} g_e v_e \tag{5.7}$$

$$\text{s.t.} \quad \sum_{\delta\in\Delta}\sum_{k\in K^\delta} a_{ik}^\delta x_k^\delta + z_i \geq 1 \qquad \forall i \in N \tag{5.8}$$

$$\sum_{k\in K^\delta} x_k^\delta = 1 \qquad \forall \delta \in \Delta \tag{5.9}$$

$$|\Delta| v_e - \sum_{\delta\in\Delta}\sum_{k\in K} b_{ek}^\delta x_k^\delta \geq 0 \qquad \forall e \in E \tag{5.10}$$

$$\sum_{e \in E_i} v_e + z_i = 1 \qquad \forall i \in N \tag{5.11}$$

$$z_i + \sum_{e' \in B_e} v_{e'} - \sum_{e' \in L_e} v_{e'} \geq 0 \qquad \forall i \in N^c : r(i) \neq 0, \forall e \in E_i \tag{5.12}$$

$$x_k^\delta \in \{0,1\} \qquad \forall \delta \in \Delta, \forall k \in K^\delta \tag{5.13}$$

$$v_e \in \{0,1\} \qquad \forall e \in E \tag{5.14}$$

$$z_i \in \{0,1\} \qquad \forall i \in N \tag{5.15}$$

We refer to Model (5.7)–(5.15) as OCRSPRT$_1$. In the objective function (5.7) the deviation from the planned crew schedule, the penalties for canceled tasks, and the penalties for delays are minimized. Constraints (5.8) ensure that every task is either assigned to one or more qualified drivers, or is canceled. By Constraints (5.9) exactly one feasible completion must be selected for every original duty. Constraints (5.10) 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 (5.11). Moreover, these constraints guarantee that deadheading is not possible on tasks which have been canceled.

Constraints (5.12) are the same as Constraints (5.3), 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 (5.12) are only required for tasks with multiple copies. Some of the Constraints (5.12) are redundant if $L_e = \emptyset$, but also if $L_e \neq \emptyset$ they can be redundant by Constraints (5.11), (5.14), and (5.15). This is true even in the linear relaxation of OCRSPRT$_1$. Note that in the example discussed above only Equation (5.4) is needed, since Equations (5.5) and (5.6) are redundant.

An alternative formulation OCRSPRT$_2$ can be obtained by replacing Constraints (5.10) in OCRSPRT$_1$ by

$$v_e - \sum_{k \in K^\delta} b_{ek}^\delta x_k^\delta \geq 0 \quad \forall \delta \in \Delta, \forall e \in E \tag{5.16}$$

Proposition 5.2. (5.16) *implies* (5.10).

Proof. If $v_e - \sum_{k \in K^\delta} b_{ek}^\delta x_k^\delta \geq 0 \;\; \forall \delta \in \Delta$, then

$$\sum_{\delta \in \Delta} \big(v_e - \sum_{k \in K^\delta} b_{ek}^\delta x_k^\delta \big) \geq 0 \Rightarrow |\Delta| v_e - \sum_{\delta \in \Delta} \sum_{k \in K^\delta} b_{ek}^\delta x_k^\delta \geq 0$$

□

Proposition 5.3. *The reverse implication of Proposition 5.2 is not true.*

Proof. It suffices to give an example. Consider $\Delta = \{1, 2, 3\}, E = \{1, 2, 3\}$ and $\sum_{k \in K} b^{\delta}_{1k} x^{\delta}_{k} = 0.5$ for $\delta \in \{1, 2\}$ and $\sum_{k \in K} b^{\delta}_{1k} x^{\delta}_{k} = 0.0$ for $\delta = 3$. Then with $v_1 = 1/3$ Constraint (5.10) would hold, but Constraint (5.16) would be violated for $\delta = 1$ and $\delta = 2$. □

Denote by LP_1 the linear relaxation of model $\mathrm{OCRSPRT}_1$ and by LP_2 the linear relaxation of model $\mathrm{OCRSPRT}_2$.

Proposition 5.4. $LP_2 \geq \ LP_1$

Proof. The proof follows directly from Propositions 5.2 and 5.3. □

Proposition 5.4 states that using Constraints (5.16) results in a tighter LP relaxation. However, $|E|$ constraints of type (5.10) are replaced by $|E||\Delta|$ constraints of type (5.16). Thus the number of constraints of type (5.16) is much larger than that of type (5.10).

After several experiments with the solution approach described in Section 5.5, we discovered that the approach of model $\mathrm{OCRSPRT}_2$ resulted in less uncovered tasks and less retimed tasks than the approach of model $\mathrm{OCRSPRT}_1$. In principle the models have the same integer solutions, but since we use an heuristic approach, we do not always find an optimal solution. We also noticed that the problem is solved slower if we use model $\mathrm{OCRSPRT}_2$ instead of model $\mathrm{OCRSPRT}_1$. However, we will accept the increase in computation time to receive better results. So, in the remainder of this chapter we only consider model $\mathrm{OCRSPRT}_2$.

5.5 Solution approach

The crew rescheduling problems arising at NS are of large scale containing 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. Therefore, we will not consider all original duties and all tasks, but we will extract core problems containing only a subset of the duties and tasks. Moreover, we will use a Lagrangian heuristic embedded in a column generation scheme very similar to the one proposed in Section 3.3. In this chapter we will investigate two approaches, which use the same heuristic to explore the core problems, but differ in the way the core problems are defined.

Our first approach is outlined in Figure 5.3. 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 we stop, otherwise we iterate over the uncovered tasks and define one new core problem per uncovered

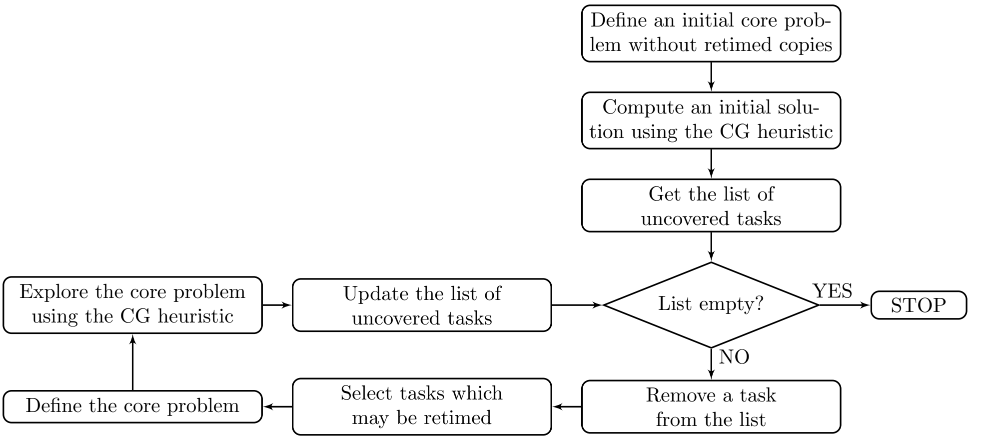


Figure 5.3: Iterative neighborhood exploration with retiming (INER)

task. We use a neighborhood definition to select the tasks for which we allow retiming and for constructing the core problems. The core problems are explored using the *column generation* (CG) *heuristic* and the list of uncovered tasks is updated. We will refer to this approach as *iterative neighborhood exploration with retiming* (INER). The difference to the approach presented in Section 3.3 is that in INER retiming of some tasks is allowed in the neighborhood exploration phase.

Our second method, outlined in Figure 5.4, does not use an iterative neighborhood exploration. If the solution of the initial core problem contains some uncovered tasks, 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. In the remainder of this chapter we refer to this approach as *extended core problem with retiming* (ECPR).

In both approaches INER and ECPR we relax the initial core problems by using only Constraints (5.8), (5.9), (5.13) and (5.15). Note that in this model it can happen that feasible completions are chosen that contain deadheading on tasks which are canceled. However, in the next core problem which is considered in both approaches, these deadheadings are not allowed anymore and a different solution will be computed. The reason we decided to use the relaxed model in the initial core problem is that the computation time for using OCRSPRT_2 is too long, whereas the relaxed model can be solved within acceptable time.

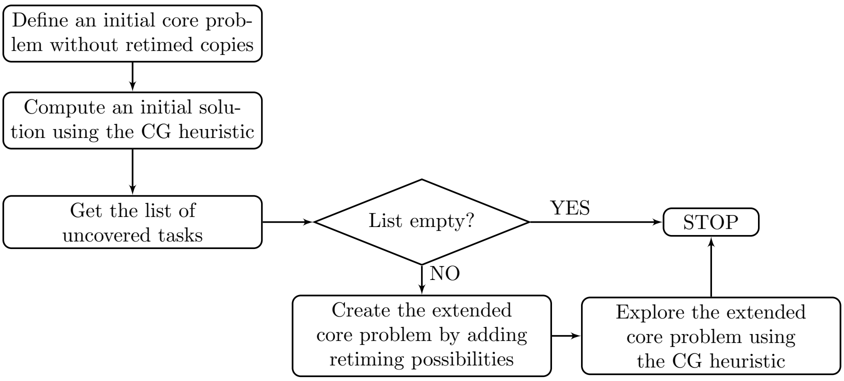


Figure 5.4: Extended core problem with retiming (ECPR)

5.5.1 Initial core problems and neighborhoods for uncovered tasks

The initial core problems in INER and ECPR are constructed in the same way as in Section 3.5.1. The intention is to select the duties that are affected by the timetable adjustments and to add some duties which contain some tasks close in space and time to the modified tasks.

Given an uncovered task, we define a neighborhood which will be extended by retiming possibilities in a subsequent step. We use the neighborhood definition of Section 3.5.2.

5.5.2 Core problems with retiming possibilities

The primary goal of retiming is to enable solutions where less tasks need 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, this indicates that there is a shortage of crew in 's-Hertogenbosch at the start time of the task. By delaying some tasks starting at 's-Hertogenbosch, we can possibly prevent the shortage. Therefore, we propose the following procedure to determine this subset. Let N^u be the uncovered tasks after solving the initial core problem. 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 $N_i^1 = \{j \in N \mid s_j^{\text{dep}} = s_i^{\text{dep}} \text{ and } t_j^{\text{dep}} \in [t_i^{\text{dep}} - p, t_i^{\text{dep}} + p]\}$ and N_i^2 is recursively defined as the set of all tasks which are linked to tasks in N_i^1 or N_i^2.

For INER $N^c = N_i^c$ 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 $\ddot{N}$ contain all tasks covered by an original duty in the neighborhood of the uncovered task under consideration when using 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 $\bar{\Delta} = \{\delta \in \Delta \mid \delta \text{ is using a task } i \in \ddot{N} \cup N^c\}$ and $\hat{N}$ is the set of all tasks used 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 using task j are in $\bar{\Delta}$. By definition of $\bar{\Delta}$ retiming is not allowed for these tasks. Denote by $\bar{N} = \{i \in \hat{N} \mid \delta \in \bar{\Delta}\ \forall \delta \in \Delta \text{ using task } i\}$.

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 use 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 Model (5.7)–(5.15), respectively.

5.5.3 Exploring the core problems

For computing near optimal solutions and lower bounds for the core problems we adapted the column generation based heuristic presented in Section 3.4.3. In the remainder of this section we discuss how we apply Lagrangian relaxation in combination with column generation, how we generate feasible solutions, and how we modify the pricing problems.

Combining column generation and Lagrangian relaxation

A lower bound for a given core problem can be obtained by Lagrangian relaxation. In this section we will present the details for model OCRSPRT$_2$. We relax Constraints (5.8), (5.16) and (5.12) of the core problems in a Lagrangian fashion using non-negative multiplier vectors λ, μ and η, respectively.

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

$$\begin{aligned}\Theta(\lambda, \mu, \eta) \quad = \quad & \min \sum_{i \in \bar{N}} \lambda_i + \sum_{\delta \in \Delta} \sum_{k \in \bar{K}^\delta} (c_k^\delta + \sum_{e \in \bar{E}} \mu_e^\delta b_{ek}^\delta - \sum_{i \in \bar{N}} \lambda_i a_{ik}^\delta) x_k^\delta \\ & + \sum_{i \in \bar{N}} (f_i - \lambda_i - \sum_{e \in \bar{E}_i} \eta_e) z_i + \sum_{i \in \bar{N}} \sum_{e \in \bar{E}_i} (g_e + \gamma_e - \sum_{\delta \in \bar{\Delta}} \mu_e^\delta) v_e \\ & \text{s.t. } (5.9), (5.11), (5.13), (5.14) \text{ and } (5.15)\end{aligned} \tag{5.17}$$

For given vectors λ, η and μ, $\Theta(\lambda, \eta, \mu)$ can be calculated with a simple procedure. First, we determine the values for all x_k^δ variables. To ensure that Constraints (5.9) are not violated, for every duty $\delta \in \bar{\Delta}$ we set x_k^δ equal to 1 for exactly one $k \in \arg\min \left\{ \bar{c}_k^\delta(\lambda, \eta, \mu) \middle| k \in \bar{K}^\delta \right\}$. Here $\bar{c}_k^\delta(\lambda, \eta, \mu) = (c_k^\delta + \sum_{e \in \bar{E}} \mu_e^\delta b_{ek}^\delta - \sum_{i \in \bar{N}} \lambda_i a_{ik}^\delta)$ 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 6 determines for every task $i \in \bar{N}$ the values for the variables z_i and v_e ($\forall e \in \bar{E}_i$) such that Constraints (5.11) are not violated.

```
1 For all e ∈ Ē_i determine ḡ_e = (g_e + γ_e − Σ_{δ∈Δ̄} μ_e^δ);
2 Select e* ∈ arg min { ḡ_e | e ∈ Ē_i };
3 if ḡ_{e*} ≤ f_i − λ_i − Σ_{e∈Ē_i} η_e then
4     Set z_i = 0, v_{e*} = 1 and for all e ∈ Ē_i \ {e*}, set v_e = 0
5 else
6     Set z_i = 1 and for all e ∈ Ē_i, set v_e = 0
```

Algorithm 6: Algorithm to determine z_i and v_e

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

$$\Theta^* = \max \Theta(\lambda, \eta, \mu), \qquad \lambda \geq 0, \eta \geq 0 \text{ and } \mu \geq 0 \tag{5.18}$$

Since the number of feasible completions can be enormous for some original duties, we combine Lagrangian relaxation with column generation. Instead of considering all feasible completions we consider only a subset in a restricted master problem (RMP). Denote by $\bar{K}_n^\delta$ the feasible completions present in the n$^{\text{th}}$ RMP. A lower bound Θ_n^* for the n$^{\text{th}}$ RMP is obtained by subgradient optimization (see e.g. Fisher (1981); Beasley (1993)).

Let λ^n, η^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 lower Lagrangian reduced cost than the feasible completions in the RMP. We will refer to them as *promising* feasible completions. The pricing problems are formulated as shortest path problem with resource constraints (see Section A.1.3). If promising feasible completions exist we add them to the RMP. Let $p_n^\delta = \min\{\bar{c}_k^\delta(\lambda, \eta, \mu) \mid k \in \bar{K}^\delta\}$ be the solution value of the pricing problem for duty δ and let $r_n^\delta = \min\{\bar{c}_k^\delta(\lambda, \eta, \mu) \mid k \in \bar{K}_n^\delta\}$ 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 λ, η and μ we try to generate feasible solutions with a Lagrangian heuristic called *GREEDY* shown in Figure 7.

In procedure *GREEDY*, we select for every duty the best feasible completion. 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 will be ignored. Moreover, we ignore feasible completions which use copies, which would violate the minimum idle time constraints (5.12). Since every set $\bar{K}_n^\delta$ contains the artificial completion without any copies, it is ensured that for every 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 the selected feasible completion does not cover any tasks.

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 18.

1 Order the original duties $\delta \in \bar{\Delta}$ by increasing reduced cost of the x_k^δ variables that were set to 1 in the Lagrangian subproblem solution;
2 Set $z_i = 1$ for all $i \in \bar{N}$ and set $v_e = 0$ for all $e \in \bar{E}$;
3 Set $\hat{\lambda} = \lambda$, $\hat{\eta} = \eta$ and $\hat{\mu} = \mu$;
4 **foreach** $\delta \in \bar{\Delta}$ **do**
5   Choose $k^*(\delta) \in \arg\min\{\bar{c}_k^\delta(\hat{\lambda}, \hat{\eta}, \hat{\mu}) \mid k \in \bar{K}_n^\delta\}$ 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, by using copy e, are not allowed to be 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 **foreach** $i \in \bar{N}$ **do**
13   Set $\hat{\lambda}_i = f_i$, if $z_i = 1$;
14 Construct the set of idle stand-by duties $\bar{\Delta}_I = \{\delta \in \bar{\Delta}_R \mid a_{ik^*(\delta)}^\delta = 0 \ \ \forall i \in \bar{N}\}$;
15 **foreach** $\delta \in \bar{\Delta}_I$ **do**
16   Set $x_{k^*(\delta)}^\delta = 0$;
17   Repeat lines 5 until 11;
18 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 is found.

Algorithm 7: Procedure *GREEDY* to construct feasible solutions

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 5.2: Information about the considered disruptions.

Solving the pricing problems

For every duty in a core problem, we construct a directed acyclic graph that contains 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. The problem of finding the path corresponding to the feasible completion with the smallest Lagrangian reduced cost is models as a shortest path problem with resource constraints (see Section A.1.3). For that purpose, we have adapted the generic dynamic programming algorithm presented in Irnich and Desaulniers (2005).

5.6 Computational results

We will 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. Some information about the scenarios is presented in Table 3.1. Furthermore, we used a crew schedule from NS that was planned for some workday in September 2007. In order to evaluate the benefits of retiming, we compare our new methods with the method proposed in Section 3.3. We will refer to the latter as *Column Generation with Dynamic Duty Selection* (CGDDS). Moreover, we will 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.

5.6.1 Parameter settings

First of all, we have some 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.

5.6.2 Cost parameters for the objective function

We use the following settings to account for the different aspects in the objective function. First, the cost of changing a duty is set to 400. The cost for sending home a stranded driver by taxi is $3,000$. Using 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 new repositioning tasks costs 1,000. The penalty for retiming a task is 200 per minute of delay.

The penalties f_i for canceling task i depend on the characteristic of the task. We say 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, 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.

5.6.3 Numerical results

For the numerical results we use some abbreviations in Tables 5.3 and 5.4: “It” is the iteration number of the general solution approach as given in Figures 5.3 and 5.4. The costs in the columns “LB” and “UB” respectively are the lower bound on the optimal solution and the cost of the best found solution of 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 previous iterations, and that are 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 for 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.

With INER and ECPR we use formulation $\mathrm{OCRSPRT}_2$. We compare the results with the CGDDS 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 CGDDS in Tables 5.3 and 5.4. A remark must be made that 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 5.3 we show the results of the three approaches in case we use the 46 stand-by duties and in Table 5.4 we show the results without using stand-by duties. By using stand-by duties we notice that the ECPR method has in all cases the best solution. However, the computation times of this approach are more than three times longer compared to the other two approaches. In terms 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 in cases *Ht:1* and *Ztm:1* less uncovered tasks than the CGDDS approach. In case *Ac:1*, retiming did not result in better crew schedules. However, a remark must be made that the solution of the method CGDDS for *Ac:1* is a crew schedule which is not compatible with the adjusted timetable since it has one driver deadheading on a canceled task.

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 the knowledge of the new route and is therefore not allowed to drive this train. This task has to be performed exactly at the moment of rescheduling. Therefore, it is not possible to cover the task without retiming it. Because of the minimum transfer time of 10 minutes, the task must be retimed with at least 10 minutes. The retiming approaches INER and ECPR only use a maximum retiming possibility of 5 minutes and were therefore not able to cover the task. In additional tests in which INER and ECPR also constructed retimed copies of 10 minutes delay, it was still not possible to cover all tasks.

If we do not use any stand-by duties (see Table 5.4), 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 compared to CGDDS.

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 the stand-by duties. Moreover, for *Ht:1*, the use of stand-by duties has increased the number of delayed tasks.

	Method	It	$\lvert\bar{\Delta}\rvert$	$\lvert\bar{N}\rvert$	$\lvert\bar{E}\rvert$	LB	UB	Gap (%)	Sol	TT (s)	A-B	A-A	DT	TD (min)
Ac:1	CGDDS	1	176	629	0	58718	59211	0.8	59211	98	1	0	0	0
Ac:1	**CGDDS**	**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	CGDDS	1	126	660	0	61661	61744	0.1	61744	95	1	1	0	0
Ht:1	CGDDS	2	77	391	0	30637	30637	0.0	61694	106	1	1	0	0
Ht:1	**CGDDS**	**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	CGDDS	1	117	432	0	51940	51991	0.1	51991	25	2	0	0	0
Ztm:1	CGDDS	2	99	247	0	43264	43264	0.0	51991	36	2	0	0	0
Ztm:1	**CGDDS**	**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 5.3: Results with stand-by drivers.

	Method	It	$\lvert\bar{\Delta}\rvert$	$\lvert\bar{N}\rvert$	$\lvert\bar{E}\rvert$	LB	UB	Gap (%)	Sol	TT (s)	A-B	A-A	DT	TD (min)
Ac:1	CGDDS	1	130	629	0	61136	62187	1.7	62187	86	1	0	0	0
Ac:1	**CGDDS**	**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	CGDDS	1	90	660	0	65567	65803	0.4	65803	94	1	2	0	0
Ht:1	CGDDS	2	44	407	0	34489	34489	0.0	65803	105	1	2	0	0
Ht:1	CGDDS	3	39	407	0	31080	31080	0.0	65803	115	1	2	0	0
Ht:1	**CGDDS**	**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	CGDDS	1	71	432	0	51991	51992	0.0	51992	16	2	0	0	0
Ztm:1	CGDDS	2	54	249	0	42058	42058	0.0	51992	23	2	0	0	0
Ztm:1	**CGDDS**	**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 5.4: Results without stand-by drivers.

5.7 Conclusions and future research

We presented two approaches to solve railway crew rescheduling with retiming. We have compared our new approaches with an approach that does not allow retiming. In 4 out of the 6 cases (*Ht:1* and *Ztm:1*, both with and without stand-by drivers), the new approaches found solutions with less canceled tasks. Moreover, the observed delay that was introduced into the timetable is very small, which makes it likely that those solutions can be implemented in practice. The computation times of the iterative neighborhood exploration with retiming (INER) approach are within a range that should make it applicable within a decision support system for disruption management.

In this chapter we have limited ourselves to consider only train drivers. However, in a disrupted situation conductors need to be rescheduled as well. This could be done as in Stojković and Soumis (2005) and Abdelghany et al. (2008) by using multiple tasks per trip that represent roles in the optimization model.

In future work conflicts between trains due to retiming decisions should be taken into account as well. We believe that the presented model and solution approaches could be extended into that direction without sacrificing computation time too much.

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 6

Railway Crew Rescheduling under Uncertainty

6.1 Introduction

Effective disruption management is a key to a good operational performance for passenger railway companies. Within the disruption management process (see Chapter 2 for a detailed discussion), the ability to reschedule crew is crucial. In this thesis we proposed a new approach for the operational crew rescheduling problem. However, this approach assumes that an accurate estimate about the duration of the disruption is available at the time the rescheduling is done. The same holds for the approach of Rezanova and Ryan (2010) and models developed for crew rescheduling in the airline industry (see Clausen et al. (2010) for a recent literature review). However, this assumption is not realistic.

Example 6.1
Let us go back to Example 2.1 taking place in the north of the Netherlands. Due to a broken power supply, no train 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 will be updated according to a pattern described by an emergency scenario. In this case, the trains of the train lines 500, 700, and 9100, operated between Zwolle (Zl) and Groningen (Gn), will be turned at intermediate stations. In Figure 6.1 we show how the timetable between Zwolle and Groningen would be updated. Since the repair works will take at least 3 hours the turning pattern will be 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 if the trains will take their normal routes (dashed lines in Figure 6.1) or if they will be turned as well (dotted lines in Figure 6.1).

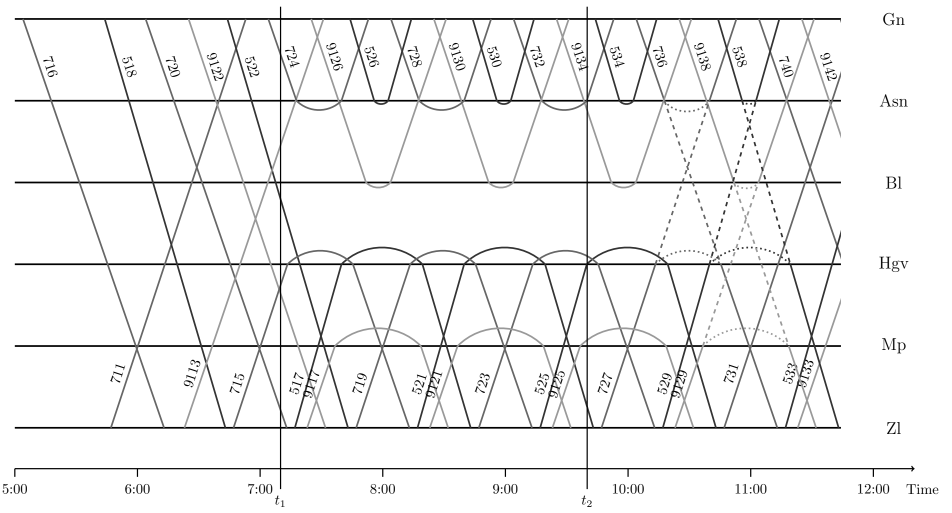


Figure 6.1: Time space diagram showing how the timetable between Groningen (Gn) and Zwolle (Zl) would be updated, if the route between Beilen (Bl) and Hoogeveen (Hgv) is blocked.

Current crew rescheduling approaches would deal with this situation as follows. At time point t_1 the duration of the disruption is estimated and the modified timetable corresponding to this estimate is used as input for the crew rescheduling problem. Actually, in practice this estimate is not an estimate in a probabilistic sense, instead often the most optimistic case is assumed. Given our example this means, that it is estimated that the blockage will be over by 10:10. Therefore, the modified timetable that is given as input to the crew rescheduling assumes that the trains 727, 736, 529, 538, 9129 and 9138 can run between Beilen and Hoogeveen as planned and therefore the corresponding tasks 727/c, 736/a, 529/e, 538/a, 9129/a, 9138/a will be considered in the OCRSP. Recall from Section 2.5.1, that 727/c refers to the third task of train 727. However, it might happen that at time point t_2, 9:40 in the example, new information becomes available saying 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 intermediate stations. At t_2 the crew schedule would be rescheduled again given the new information, meaning the rerouted tasks 727/c^r, 736/a^r, 529/e^r, 538/a^r, 9129/a^r, 9138/a^r would be considered in the OCRSP.

If at t_1 the uncertainty about the duration of the disruption, and therefore the uncertainty about the timetable that will be operated, is not taken into account, we will refer to the above approach as the *expected scenario* approach. In order to take the uncertainty into account at time t_1, we develop in this chapter a quasi robust optimization approach, which uses ideas from robust optimization.

6.2 Problem description

Most symbols used in this chapter will have the same meaning as in Chapters 3–5. However, in order to present a convenient notation, the meaning of some symbols has been changed.

We consider crew rescheduling under uncertainty as a two-stage problem. In Stage 1 at t_1 an estimate of the duration h_1 of the disruption is known. In the case of a malfunctioning switch for example, this estimate could be based on the initial judgment of a repair crew. Based on the estimated duration, the original timetable T_0 will be adjusted according to the unavailable infrastructure. The result is an adjusted timetable T_1. Then the crews are rescheduled according to this adjusted timetable. Later, at time t_2 it becomes clear when the infrastructure can definitely be used again. Often this is later than the expected time $t_1 + h_1$. Usually this means that timetable T_1 cannot be operated and instead another adjusted timetable T_2 will be operated. This could mean that in Stage 2 at time t_2 the drivers need to be rescheduled again according to T_2.

We assume that the timetable that will be operated in the end is one of a small number of possibilities. We refer to these possibilities as scenarios S, indexed by s, where $\bar{s}$ corresponds to the scenario which would be used for the rescheduling in Stage 1 at t_1 in an expected scenario approach. Moreover, let N_s be the tasks in scenario $s \in S$. The crew rescheduling problem under uncertainty can be stated as follows. Given a scenario $\bar{s}$, and a set of alternative scenarios $S \setminus \bar{s}$, find a new crew schedule valid for $\bar{s}$ such that the sum of the cost of this schedule and the expected cost for the additional rescheduling in the second stage at t_2 is minimized. Note that this implies that a probability $p(s)$ is given for all scenarios $s \in S$. In this chapter we will assume that t_2 is the same for all scenarios.

Given that the timetable modifications will follow some structured emergency scenarios, we can assume the following relationship between all scenarios. Given one reference scenario, in our case the optimistic scenario $\bar{s}$, all other possible scenarios can be obtained by removing or rerouting tasks of the reference scenario $\bar{s}$. We will refer to the tasks that are rerouted or canceled in scenarios $s \in S \setminus \bar{s}$ with respect to scenario $\bar{s}$ as *critical* tasks. If a critical task i is rerouted in a scenario $s \neq \bar{s}$ we refer to the associated rerouted alternative as $i(s) \in N_s$. Formally stated, a task $i(s)$ is rerouted with respect to task $i \in N$ if $t^{\text{dep}}_{i(s)} \neq t^{\text{dep}}_i$, or $t^{\text{arr}}_{i(s)} \neq t^{\text{arr}}_i$, or $s^{\text{arr}}_{i(s)} \neq s^{\text{arr}}_i$, or task $i(s)$ takes a different route in the railway network. Let $D(N_s)$ be the set of critical tasks that are rerouted or canceled in

N_s. Furthermore, we assume that the scenarios S are ordered such that $D(N_r) \subset D(N_s)$ if $r < s \; \forall \; r, s \in S$. By definition $D(N_{\bar{s}}) = \emptyset$. Moreover, let $\underline{s}$ be the last $s \in S$.

For application in practice it would be necessary that dispatchers can specify the set of possible scenarios. We think that experienced dispatchers are able to indicate a best and a worst case scenario. However, we are not sure if dispatchers are able to specify all possible scenarios and can estimate the required probabilities in a practical setting. This should be kept in mind when considering mathematical models and solution approaches for railway crew rescheduling under uncertainty.

6.2.1 Optimization under uncertainty

In many optimization problems one has to deal with some kind of uncertainty. Famous examples are e.g. the weather conditions in the farmer's problem, the sales numbers in the newsvendor (newsboy) problem (see e.g. Birge and Louveaux (1997) for both problems), and the return of a capital investment in the portfolio selection problems (see Markowitz (1952) for a classical reference). We will briefly review three concepts that have been developed in order to deal with uncertainty.

Two-stage *stochastic programming* with *recourse* minimizes the sum of the cost for the first stage solution plus the expected cost for the recovery in the second stage. An assumption in stochastic programming is that the probability for each of the considered scenarios is known. For more information on stochastic programming we refer the interested reader to Birge and Louveaux (1997) and Kall and Wallace (1994). Crew rescheduling under uncertainty fits well into the framework of stochastic programming with recourse. However, there are two concerns about the applicability of stochastic programming. First, we are not sure if dispatchers are able to specify the possible scenarios the required probabilities in the disrupted situation. Second, two-stage stochastic programming problems with integer first and second stage decision variables are in general very difficult to solve (see Klein Haneveld and Van der Vlerk (1999)).

Robust optimization tries to find the best solution that, without any modifications or recovery actions, stays feasible under all specified scenarios. For a two-stage problem such as crew rescheduling under uncertainty this would mean that when the realized scenario gets known at time t_2 it is not allowed to reschedule again. Therefore robust optimization would compute the best feasible crew schedule for the timetable corresponding to the longest duration of the disruption. This crew schedule will not be changed at time t_2 which in turn means that the timetable corresponding to the longest duration will be operated in any case. This would be unacceptable from a passenger point of view. Thus it is obvious that robust optimization is of no use for crew rescheduling under uncertainty.

Recoverable robustness was introduced by Liebchen et al. (2007) and Liebchen et al. (2009). The aim of the notion of recoverable robustness is to overcome some shortcomings of “classic ” robust optimization by considering recovery actions. The building blocks in concept of recoverable robustness are: (1) an original optimization problem, (2) a set of scenarios representing the imperfection of the information in the original optimization problem, and (3) (limited) recovery possibilities. The limited recovery possibilities are specified via admissible recovery algorithms. Given an original optimization problem, a set of possible scenarios and admissible recovery algorithms. A solution X to the original optimization problem is called *recovery-robust* if a feasible solution can be recovered from X with one of the admissible algorithms in all specified scenarios. Liebchen et al. (2009) present e.g. how an upper bound on the recovery cost can be integrated within this concept. Moreover, they show that some classes of recoverable robust optimization problems can be solved by linear programming. However, the concept of recoverable robustness has not been applied in a situation comparable to crew rescheduling under uncertainty yet.

6.2.2 Related work

The consideration of uncertainty during resource rescheduling was suggested in Rosenberger et al. (2003) as a subject for future research. However, till today there are hardly any papers on this topic in the scientific literature. Nielsen (2008) presents a rolling horizon approach to rolling stock rescheduling that is designed to take updates of the timetable into account. For rolling stock rescheduling this approach seems to be promising and is practical because the simpler structure of rolling stock duties enables the author to derive desired situations at the end of each horizon. Because the crew duties in the crew schedule of NS cover relatively large distances it seems difficult to apply a rolling horizon approach in our case.

Within crew scheduling, robustness has so far only been considered in the planning phase. *Robust crew scheduling* aims at making the crew schedule more robust against uncertainties in the operation. Two aspects of robustness can and should be considered: The first aspect, sometimes referred to as stability, is the ability of a crew schedule to absorb or to limit the propagation of delays without recovery actions being taken. The second aspect, called recoverability, takes common recovery actions into account when constructing a crew schedule. There have not been notable publications on robust crew scheduling for passenger railways yet. For research on the similar airline problem we refer to Ehrgott and Ryan (2002), Schaefer et al. (2005), Yen and Birge (2006), Shebalov and Klabjan (2006), and Weide et al. (2010).

6.3 Quasi robust optimization approaches

In Section 6.2.1 we have reviewed three concepts for optimization under uncertainty: Robust optimization, two-stage stochastic programming, and recoverable robustness. The first two do not seem to be appropriate for crew crew rescheduling under uncertainty. Robust optimization is too conservative, two-stage stochastic programming needs a lot of information and is computationally very difficult. This motivates investigating a quasi robust optimization approach which offers two advantages. First of all, less information needs to be provided by the dispatchers, since only an optimistic (best case) scenario $\bar{s}$ and a worst case scenario $\underline{s}$ need to be specified. The second advantage is that we believe that the quasi robust optimization approach can be solved with less computational effort, which is very important given the practical disruption management setting.

The idea behind quasi robust optimization approaches is to use feasible completions that are, in some sense, robust against all scenarios and in this way minimize the cost for rescheduling in the second stage at time t_2 if a scenario other than $\bar{s}$ occurs. Another argument in favor of this approach is that, as we will show, it requires only small modifications in the solution method presented in Chapter 3.

6.3.1 Mathematical model

Let us first give an informal definition of quasi robust feasible completions. A feasible completion is called quasi robust if all tasks that are used by this feasible completion in scenario $\bar{s}$ can also be used in every other scenario. A task is used by a feasible completion if the driver is driving this task or if the driver is deadheading on this task. If a feasible completion k, valid for scenario $s \in S$ is using task $i \in N_s$, then $b^{\delta}_{ik} = 1$, otherwise $b^{\delta}_{ik} = 0$. Moreover, $a^{\delta}_{ik} = 1$ if the driver associated with a feasible completion is allowed to drive task i, otherwise $a^{\delta}_{ik} = 0$.

Definiton 6.2. *A feasible completion γ in the second stage OCRSP, in the case scenario s occurs, is a **recovery alternative** for a feasible completion k from the first stage OCRSP if $b^{\delta}_{i\gamma} = 1$ for all $i \in N_{\bar{s}}$ with $i(s) \in N_s$ and $b^{\delta}_{ik} = 1$.*

Definiton 6.3. *A feasible completion k is called **quasi robust** if there exists a recovery alternative γ for all $s \in S$.*

Note that by Definition (6.3) every feasible completion that does not use any critical task is quasi robust. Another observation concerns critical tasks i that are canceled in scenario $\underline{s}$ and are of type A-A, meaning that the departure station s^{dep}_i is the same as the arrival station s^{arr}_i. A feasible completion containing only critical tasks of type A-A is quasi robust, since we can just leave out these tasks in the recovery alternative. Note that this holds because we have no limitations on the length of a transfer. Given the route blockage between Hoogeveen and Beilen and the possible scenarios as presented in Example 6.1 let us give an example of the concept of quasi robust feasible completions.

Example 6.4
Figure 6.2.a shows the planned duty from crew base Groningen (Gn). Due to the route

blockage the task 724/a from Groningen to Zwolle (Zl) is rerouted and returns to Groningen. Therefore, the driver cannot follow his planned duty. A feasible completion of the duty under the optimistic scenario $\bar{s}$, is shown in Figure 6.2.b. The optimistic scenario $\bar{s}$ assumes that the route blockage lasts until 10:10. Since this completion does not cover any of the critical tasks it is a quasi robust completion. The completion in Figure 6.2.c is a non-robust one. It covers the critical task 736/a from Groningen to Zwolle. If scenario $\underline{s}$, meaning the route will be blocked until 11:10, occurs, this task is rerouted (736/a^r) and ends in Groningen and the driver will not be able to get to Zwolle in time to deadhead on task 538/b from Zl to Amersfoort (Amf). Figure 6.2.d shows a quasi robust feasible completion covering this critical task. Its recovery alternative that is valid in $\underline{s}$ is shown in Figure 6.2.e.

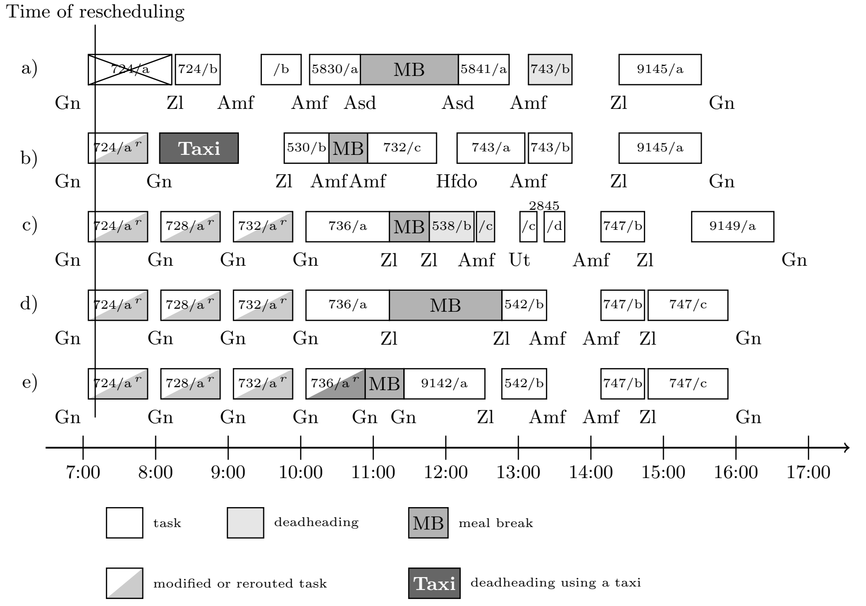


Figure 6.2: Examples of feasible completions for an affected original duty from crew base Groningen (Gn).

Based on the definition of quasi robust feasible completions we derive the quasi robust operational crew scheduling problem (QROCRSP). Denoting the set of quasi robust feasible completions for original duty δ by $R^\delta \subseteq K^\delta$ we can state the *strong* QROCRSP as:

$$\min \quad \sum_{\delta\in\Delta}\sum_{k\in R^\delta} c_k^\delta x_k^\delta + \sum_{i\in N_{\bar{s}}} f_i z_i \tag{6.1}$$

$$\text{s.t.} \quad \sum_{\delta\in\Delta}\sum_{k\in R^\delta} a_{ik}^\delta x_k^\delta + z_i \geq 1 \qquad \forall i \in N_{\bar{s}} \tag{6.2}$$

$$\sum_{k\in R^\delta} x_k^\delta = 1 \qquad \forall \delta \in \Delta \tag{6.3}$$

$$x_k^\delta, z_i \in \{0,1\} \qquad \forall \delta\in\Delta, \forall k \in R^\delta, \forall i \in N_{\bar{s}} \tag{6.4}$$

The quasi robust optimization problem given by Formulation (6.1)–(6.4) is almost the same as the classical operational crew rescheduling problem OCRSP given by Formulation (3.1)–(3.4). The only difference is that we only consider quasi robust feasible completions R^δ.

We can derive a variant by requiring the use of quasi robust feasible completions only for the infeasible original duties Δ_C. The resulting *weak* QROCRSP can be derived from Formulation (6.1)–(6.4) by replacing the sets R^δ with $\mathcal{K}^\delta$, where

$$\mathcal{K}^\delta = \begin{cases} R^\delta & \text{if } \delta \in \bar{\Delta}_C \\ K^\delta & \text{otherwise.} \end{cases}$$

Solving the strong QROCRSP in the first stage at time t_1, gives the guarantee that a recovery alternative exists for each original duty. If at time t_2 it becomes certain that the timetable will be operated according to scenario $s \in S$ we know that a crew schedule constructed from the recovery alternatives covers all tasks $i \in N_s$. Therefore, every solution X to the strong QROCRSP is recovery-robust (see e.g. Liebchen et al. (2007)) against the scenarios in S given a recovery algorithm that finds the recovery alternative for the feasible completions in X in the second stage OCRSP at time t_2.

6.3.2 Solution approach

We will solve the quasi robust crew rescheduling problem with an adapted version of the column generation based heuristic proposed in Chapter 3. Because for some or all original duties we consider only quasi robust feasible completions, we are interested in modifying the pricing problems such that only quasi robust feasible completions are generated.

The column generation pricing problem

Recall from Section 3.4.2 that we have modeled the pricing problems for every original duty δ as a shortest path problem with resource constraints (SPPRC) on a directed acyclic graph. While computing a feasible completion we need to be able to check if it is quasi robust. In other words we must be sure that a recovery alternative for k exists for all scenarios $s \in S$.

By construction of the pricing problem graphs, any feasible completion corresponds to a path in a pricing problem graph. Let n_i^{arr} (n_i^{dep}) be the arrival (departure) node of task i. A task j is said to be used directly after task i in a feasible completion if n_i^{arr} is directly followed by n_j^{dep} in the corresponding path. We denote all predecessor (successor) nodes of node n by $pred(n)$ ($suc(n)$).

We have to modify the pricing problems in order to guarantee the existence of a recovery alternative. This will be done by (i) modifying the graphs and (ii) considering additional resources in the SPPRC. The latter is necessary in order to account for the meal break rule. Therefore we use five additional arc properties as shown in Table 6.1 next to the cost. These arc properties are used to define the resource extension functions (REFs) for the two resources tcp and $tcpa$ which measure for every subpath the time spent in the current part (before or after the meal break) of the associated duty. In Table 6.2 we present the corresponding REFs and resource windows. Before the preprocessing we set $tca = tc$, $mba = mb$, and $taba = 0$ on all arcs.

Property	Description
tc	The time consumption in minutes.
tca	The time consumption in minutes in the recovery alternative.
mb	Indicates if the arc corresponds to a meal break.
mba	Indicates if the arc corresponds to a meal break in the recovery alternative.
$taba$	Time in minutes after the meal break in the recovery alternative.

Table 6.1: Information about the arc properties used in the SPPRC.

Resource	Resource extension function (REF)	Resource window
tcp	$f_{ij}^{tcp}(T_i) = \begin{cases} 0 & \text{, if } t_{ij}^{mb} = 1 \\ T_i^{tcp} + t_{ij}^{tc} & \text{, otherwise} \end{cases}$	$[0, 330]$
$tcpa$	$f_{ij}^{tcpa}(T_i) = \begin{cases} t_{ij}^{taba} & \text{, if } t_{ij}^{mba} = 1 \\ T_i^{tcpa} + t_{ij}^{tca} & \text{, otherwise} \end{cases}$	$[0,\ 330]$

Table 6.2: Information about the resources and the corresponding REFs used in the SPPRC.

For every critical task, except those of type A-A mentioned above, starting with the one with the earliest departure time, we solve an auxiliary shortest path problem and then modify the pricing problem graphs according to the outcomes. For the auxiliary problem, we start for every original duty with an auxiliary graph that is identical to the graph that would be used in the pricing problem in the OCRSP. Then we remove the arcs corresponding to using a critical task i that is canceled in $\underline{s}$. For a critical task i that will be rerouted in some scenarios $s \in S$ we replace the arcs representing using this task by the arcs representing using the rerouted alternative. Moreover, we apply this also to critical tasks which start before the end time of critical task i.

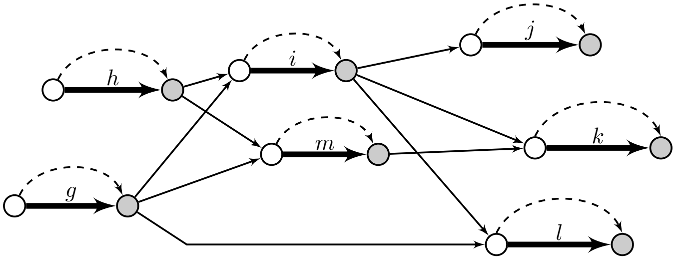
After these graph modifications we can easily determine for every predecessor node h of n_i^{dep} all successor nodes of n_i^{arr} that can be reached from h. However, we are not interested in reachability alone, but we are also interested in information about how a successor node can be reached. To be more precise, we would like to find the path from h to a successor node $j \in \text{suc}(n_i^{\text{arr}})$ with the smallest time after a possible meal break. This information is represented by the *tcp* resource. Hence, we solve an auxiliary shortest path problem where we only use the resource *tcp* and the corresponding REF. In the label setting algorithm in every node we only keep one label with the minimum value of *tcp*.

Then we modify the pricing problem graph as follows. We remove the arcs corresponding to the use of task i and replace them for every predecessor h by a copy for every successor of n_i^{arr} that can be reached from h. We copy the original transfer arcs and set the resource consumptions for $tca = 0$ and the property $mba = 0$. On the arcs corresponding to task i we set *tca*, *mba*, and *taba* according to the resource consumption of the path found when solving the auxiliary problem.

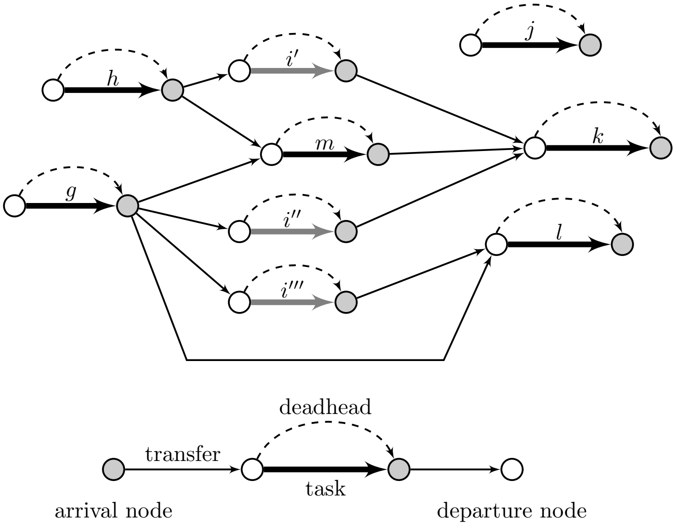
Example 6.5

In Figure 6.3 we show an example of a part of a pricing problem graph before (Figure 6.3a) and after (Figure 6.3b) the preprocessing, where i is a critical task that will be canceled in scenario $\underline{s}$. The departure node n_i^{dep} of task i has two predecessors, $\text{pred}(n_i^{dep}) = \{n_h^{arr}, n_g^{arr}\}$ and n_i^{arr} has three successors, $\text{suc}(n_i^{arr}) = \{n_j^{dep}, n_k^{dep}, n_l^{dep}\}$. From n_h^{arr} only n_k^{dep} can be reached via task m in the auxiliary problem when the tasks corresponding to task i have been removed. For this relation we introduce new nodes $n_{i'}^{dep}$ and $n_{i'}^{arr}$ and the necessary arcs. From n_g^{arr}, n_k^{dep} and n_l^{dep} are reachable in the auxiliary problem. Note that n_j^{dep} cannot be reached from any predecessor of n_i^{dep}.

It becomes clear from above example that the number of nodes and arcs in the pricing problem graphs can 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 quasi robustness on instances of practical relevance. This will be shown in Section 6.4.



(a) Before preprocessing



(b) After preprocessing

Figure 6.3: A part of a pricing problem graph before and after the preprocessing for critical task i that will be canceled in scenario $\underline{s}$.

6.4 Numerical results

In this section we will test two variants of a quasi robust optimization approach for crew rescheduling under uncertainty on instances of practical relevance using crew schedules from NS. The two variants are the weak (WQR) and the strong (SQR) quasi robust optimization approach based on the weak QROCRSP and the strong QROCRSP models presented in Section 6.3. We will compare these new approaches with the expected scenario (ES) approach.

The quasi robust optimization approaches have been implemented in C++ and compiled with the Visual C++ 9.0 compiler. We used a *Intel Pentium D* processor with 2 GB RAM clocked at 3.4 GHz for the test runs. The parameters for the objective function is the first stage as well as in the second stage problem are the same as specified in Section 3.6.

Based on instances for the OCRSP we have constructed five instances for crew rescheduling under uncertainty. Each of the five instances has two scenarios, namely $\bar{s}$ and $\underline{s}$. Note that for the quasi robust optimization approach the number of scenarios does not matter. What matters, however, is the number of critical tasks. In Tabel 6.3 we present some information about the five instances. For every instance we present the expected (optimistic) length of the disruption and the time the disruption lasts longer in scenario $\underline{s}$. Moreover, we show the resulting number of critical tasks.

Instance	expected duration	considered extension	Critical tasks: Canceled in $\underline{s}$	Rerouted in $\underline{s}$	Total
Ac_C	2:00	0:30	4	4	8
Bl_A	3:00	1:00	0	6	6
Bl_B	3:00	1:00	1	4	5
Ht_A	3:00	0:30	8	2	10
Ztm_A	3:00	1:00	16	6	22

Table 6.3: Information about the disruptions and the considered uncertainty.

In the first stage problem we considered initial core problems as described in Section 3.3. In order to account for the uncertainty we constructed the initial core problems based on the expected duration plus the possible extension. In Table 6.4 we present the computation times and the number of arcs in the pricing problem graphs for the ES, WQR, and SQR approach. All tasks are covered in the solutions of the first stage problem by all approaches for all instances. The number of arcs that lie on a path from the source to the sink node in the pricing problem graphs are shown in column *Arcs*. All other arcs are removed in a preprocessing step before we start the column generation procedure. As expected the number of arcs is much higher for the quasi robust optimization approaches.

For the instances Ac_C, Ht_A, and Ztm_A the number of arcs in the SQR approach is more than 10 times the number of arcs in the ES approach. For the instances Bl_A and Bl_B the increase in the number of arcs is less but still above a factor of five.

	ES		WQR		SQR	
Instance	Time (s)	Arcs	Time (s)	Arcs	Time (s)	Arcs
Ac_C	264.6	625,259	433.9	2,238,209	727.5	6,444,493
Bl_A	38.6	39,727	36.3	125,364	34.7	300,843
Bl_B	25.8	41,796	24.6	57,841	24.8	233,296
Ht_A	397.8	526,928	1,097.1	4,401,100	1,911.2	9,266,084
Ztm_A	155.7	519,163	703.1	5,605,934	1,364.1	18,752,666

Table 6.4: Computation time and number of arcs in the pricing problems for the expected scenario (ES), weak quasi robust (WQR), and strong quasi robust (SQR) solution approach in the first stage.

Interestingly, the computation time is about the same for the three approaches for the instances Bl_A and Bl_B. For the other instances the computation time for the WQR and SQR approach is considerably longer than for the ES approach. For the ES approach we observe the longest computation time of around 400 seconds for instance Ht_A. For the WQR approach the longest computation time is approx. 1,100 seconds for instance Ht_A. For the same instance we also observed the longest computation time for the SQR approach, of just less than 2,000 seconds. The computation time for the SQR approach is at most 5 times longer compared to the ES approach.

We also solved the second stage problems using the results of the first stage problems as input. All second stage problems have been solved with the algorithm presented in Chapter 3. Note that we do not require the use of the recovery alternatives. We show the solution values of the first and second stage problems in Table 6.5. For the three solution approaches we report three numbers. Columns $C(\bar{s})$ display the solution value of the first stage problem obtained by the approach. The solution value of the first stage problem plus the solution value of the second stage problem if scenario $\underline{s}$ occurred is shown in columns $C(\underline{s})$. The column *Diff* states $C(\underline{s}) - C(\bar{s})$ which is the solution value of the second stage problem under the realization of scenario $\underline{s}$. Note that by definition of the first stage problem, the solution value of the second stage problem under scenario $\bar{s}$ is 0. First of all, we notice that the effort for the rescheduling in the second stage if scenario $\underline{s}$ occurs is the smallest for the SQR approach except for Ztm_A. This illustrates that the concept of quasi robustness succeeds in keeping the cost for the rescheduling in the second stage small. On the other hand, the cost of the first stage solution is the highest for all instances. Moreover, the sum of the two, which is equal to $E(\underline{s})$ is also higher compared to the ES and WQR approach, with the exception of Ac_C and the ES approach. We therefore conclude that under the considered objective function the SQR approach is inferior to the ES and WQR approach.

	ES			WQR			SQR		
Instance	$C(\bar{s})$	$C(\underline{s})$	Diff	$C(\bar{s})$	$C(\underline{s})$	Diff	$C(\bar{s})$	$C(\underline{s})$	Diff
Ac_C	27,760	37,837	10,077	29,064	35,788	6,724	32,436	36,451	4,015
Bl_A	10,586	14,248	3,662	10,989	14,599	3,610	14,058	16,917	2,859
Bl_B	9,439	15,159	5,720	9,789	13,748	3,959	12,899	16,458	3,559
Ht_A	38,050	49,243	11,193	38,300	47,483	9,183	41,758	49,587	7,829
Ztm_A	11,596	17,618	6,022	11,842	17,311	5,469	16,917	22,436	5,519

Table 6.5: Comparison of the expected scenario (ES), weak quasi robust (WQR), and strong quasi robust (SQR) solution approach.

We will now focus on the comparison of the ES and the WQR approach. Comparing the columns *Diff* we see that the WQR approach performs better in the second stage if scenario $\underline{s}$ occurs. The absolute difference is the largest for Ac_C. Furthermore, the values $C(\underline{s})$ are smaller for WQR for all instances except Bl_A. This means that for the WQR approach the improvements in the second stage problem under scenario $\underline{s}$ are lager than the difference in the solution values of the first stage problem in 4 out of 5 instances.

Next we are going to extend our comparison of the ES and WQR approach by using probabilities for the possible scenarios. Let $p(\bar{s})$ ($p(\underline{s})$) be the probability for scenario $\bar{s}$ ($\underline{s}$). In this analysis we will assume that $\bar{s}$ and $\underline{s}$ are the only possible scenarios and hence $p(\bar{s}) = 1 - p(\underline{s})$. Now we can compute the expected values for an approach as $EV = p(\bar{s})C(\bar{s}) + p(\underline{s})C(\underline{s})$. We will refer to these expected values as EV(ES) for the ES approach and EV(WQR) for the WQR approach. For Bl_A EV(ES) < EV(WQR) for $0 \leq p(\underline{s}) \leq 1$. For the other four instances we have that EV(WQR) < EV(ES) for $p(\underline{s}) = 1$. Moreover, we can compute a threshold value $q(\underline{s})$ such that EV(ES) $\geq$ EV(WQR) if $p(\underline{s}) \geq q(\underline{s})$ (see Table 6.6). The threshold of the probability for scenario $\underline{s}$ such that using the WQR approach pays off ranges from 19.9% for Bl_A to 44.5% for Ztm_A.

6.5 Concluding remarks and future work

In this chapter we discussed how the uncertainty about the length of a disruption can be considered as a two stage optimization problem. We explained why it is difficult to apply known concepts for optimization under uncertainty. In order to overcome the shortcomings of classical robust optimization, we presented the novel concept of quasi robustness for crew rescheduling under uncertainty. Based on the concept of quasi robustness we

Instance	$q(\underline{s})$
Ac_C	0.240
Bl_A	–
Bl_B	0.199
Ht_A	0.416
Ztm_A	0.445

Table 6.6: Thresholds $q(\underline{s})$ such that the quasi robust approach outperforms the expected scenario approach for $p(\underline{s}) \geq q(\underline{s})$.

developed two new optimization models, namely the strong quasi robust and the weak quasi robust operational crew rescheduling problem. These models differ from "classic" models for crew rescheduling in the set of feasible solutions. Furthermore, we have shown that near optimal solutions to these models can be obtained by column generation based algorithms that have been developed for crew rescheduling, when the column generation pricing problem is modified. This is one advantage of the quasi robust approaches, they do not require new special purpose algorithms but only small modifications to existing algorithms. Another advantage is that no probabilities for the possible scenarios are used within the quasi robust approaches. This is an interesting feature to apply this concept in practice since it is doubtful if these probabilities would be available in a real-time setting.

In the numerical study we have compared the two quasi robust approaches with an expected scenario (ES) approach that ignores the uncertainty in the length of the disruption. We have shown that the strong quasi robust (SQR) approach is inferior in terms of solution quality and computation time. The weak quasi robust approach (WQR) produces better results for 4 out of the 5 considered instances. However, the computation time was up to 4.5 times longer compared to the ES approach. This bottleneck should be addressed in future research. One should exploit that the restricted master problem as well as the pricing problem are separable per original and parallelize the implementation. We think that this will reduce the computation times in way the WQR approach will be applicable in practice. Another possibility for further research is to design an iterative algorithm that starts computing the solution of the ES approach and then produces different, hopefully more robust, solutions by using the concept of quasi robustness.

Chapter 7

Summary and Concluding Remarks

In this thesis we discussed several facets of the operational (railway) crew rescheduling problem. Nowadays, this problem is still one of the bottlenecks in the disruption management process of passenger railway operators. Although models and algorithms for the counterpart in the airline world have been available for more than ten years, dispatchers of European passenger railway operators do not have automated decision support tools for this problem yet.

In Chapter 2 we considered the role of crew rescheduling in the disruption management process of passenger railway operators. We illustrated the interaction with the other two main steps in disruption management, namely timetable adjustment and rolling stock rescheduling. Moreover, we discussed the information flow between the different actors in the disruption management process. Furthermore, we reviewed the scientific literature on railway disruption management and concluded that there is a need for optimization models and fast algorithms especially for rescheduling the two main resources rolling stock and crew.

We presented a novel algorithm for crew rescheduling in Chapter 3. As many approaches that have been proposed in the literature for airline crew rescheduling, also our algorithm considers only a part of the given crew schedule, referred to as a core problem, in order to be able to compute solutions within a couple of minutes. The selection of the initial core problems has proved to lead to very good results. In some cases however, some tasks remain uncovered in the solution of the initial core problem. In this case the algorithm defines new core problems representing a neighborhood of the uncovered tasks. Such a neighborhood exploration scheme has not yet been used in the context of crew rescheduling. Because the core problems are selected based on duties, we refer to this algorithm as column generation with dynamic duty selection (CGDDS). We show that for some instances we can improve the solution of the initial core problem via the neighborhood exploration within little additional computation time. Near optimal solutions for the core problems are computed with an algorithm that combines column generation and Lagrangian relaxation techniques. In addition, we reported on a real-life application. In March 2009, our algorithm was used to compute new schedules for the train drivers when some important route of the Dutch railway network was available only at limited capacity after the derailment of a cargo train.

A computational comparison of three methods for crew rescheduling has been conducted in Chapter 4. We compared our CGDDS algorithm with a new 2 Phase Repeated Shortest Path with Resource Constraints heuristic (2P-RSPPRC), a heuristic that mimics manual dispatching, and a heuristic based on dynamic constraint aggregation (DCA). DCA is an advanced column generation method where set partitioning constraints of the restricted master problem are dynamically grouped and regrouped into clusters. In the numerical study we showed that the 2P-RSPPRC heuristic is inferior since it fails to find good solutions for most of the instances. Moreover, it turned out that for some instances DCA outperforms a classic column generation method in terms of computation time for some instances, while it is the other way around for other instances. We concluded that DCA in its current form is not yet understood well enough. More research is necessary to refine the method in order to consistently perform better than classic column generation. Finally, we have shown that for some instances significantly better solutions than the one found by our CGDDS method exist. This motivates further research into the neighborhood exploration scheme of our CGDDS method. However, the main conclusion of Chapter 4 is that our CGDDS method is best suited for railway crew rescheduling among the compared methods because it finds good solutions within a short amount of computation time.

In Chapter 5 we extended the crew rescheduling problems by the possibility to delay the departure of some trains. This extension can be seen as a partial integration of timetabling and crew rescheduling. We presented a mathematical model for railway crew rescheduling with retiming and showed how the CGDDS method can be adapted in order to solve this new model. In the computational results we showed that by allowing retiming we can reduce the number of uncovered tasks in the crew rescheduling solutions. This is a very interesting improvement from the point of view of the surrounding disruption management process, because it could save some iterations of the timetable adjustment, rolling stock rescheduling and crew rescheduling loop.

Finally, in Chapter 6 we evaluated what happens if a disruption lasts longer than expected. Therefore, we described a new problem namely crew rescheduling under uncertainty. This is a two stage problem where in the first stage a number of possible scenarios is known and a crew schedule for the most optimistic scenario must be computed. In the second stage when it has become clear which scenario occurred, additional actions have to be taken. These additional actions are, in fact, another rescheduling based on the earlier computed crew schedule. We presented the notion of quasi robust feasible completions and based on that notion we presented two optimization models that take the uncertainty about the length of the disruption into account. Furthermore, we showed that by modifying the column generation pricing problems we can use the CGDDS algorithm presented in Chapter 3 to obtain near optimal solutions for these quasi robust optimization models. In the numerical study we showed that it is often advantageous to use one the quasi robust optimization approaches instead of not considering the uncertainty in the first stage optimization problem.

Within the next year, the presented research should lead to the availability of algorithmic decision support tools for the dispatchers at Netherlands Railways (NS). We suggest to integrate the CGDDS algorithm we presented in Chapter 3 into the current computer systems for dispatching in the control centers. Already the available implementation is powerful enough to deal with practical instances under the constraint of the available computation time. That there is a lot of room for improvement e.g. by exploring a parallel implementation is a plus. Moreover, it is possible to extend the CGDDS algorithm in subsequent steps to consider the uncertainty in a more sophisticated way. Anyway, we recommend NS to make the basic version available in a decision support tool as soon as possible because this will already result in a better operational performance with less delays and less canceled trains.

Appendix A

Selected Topics in Combinatorial Optimization

A.1 Selected combinatorial optimization problems

In this section we will discuss some well-known combinatorial optimization problems which play an important role in this thesis. In combinatorial optimization problems one aims to minimize or maximize an objective function over a countable set of feasible solutions. The set of feasible solutions can usually be described mathematically using constraints and decision variables of which some or all have to take discrete values.

If the objective function as well as the constraints are linear, a combinatorial optimization problem can be modeled as a *mixed integer program* (MIP), or *integer program* (IP) in the case that all decision variables have to take discrete values. For the theory behind and general solution approaches for MIPs we refer the interested reader to the excellent books of Schrijver (1986) and Nemhauser and Wolsey (1988).

A.1.1 Set partitioning/covering problem

The minimization version of the *set partitioning problem* can be stated as follows: Given a finite set S and a finite family F of subsets of S, with cost c_f associated with each $f \in F$, find a minimum cost subset F' such that F' is a partition of S.

This problem can be formulated as an *integer program* (IP). To this aim, we associate a binary decision variable x_f with every subset in the family F. This variables is set to 1 if f is part of F' and 0 otherwise. Moreover, let $a_{if} = 1$ if subset $f \in F$ contains element

$i \in S$ and $a_{if} = 0$ otherwise. Now the set partitioning problem reads:

$$\min \quad \sum_{f \in F} c_f x_f \tag{A.1}$$

$$\text{s.t.} \quad \sum_{f \in F} a_{if} x_f = 1 \qquad \forall i \in S \tag{A.2}$$

$$x_f \in \{0,1\} \qquad \forall f \in F \tag{A.3}$$

It can be shown that the decision version of the set partitioning problem is NP-complete if there are at least three elements in each subset (see Garey and Johnson (1979)).

The *set covering problem* is a relaxation of the set partitioning problem, its IP formulation can be obtained by replacing the "=" sign in Constraint (A.2) with a "≥" sign. It is obvious that every feasible solution to a set partitioning problem is also a feasible solution to the corresponding set covering problem. The decision version of the set covering problem has been shown to be NP-complete by Karp (1972).

Crew scheduling problems that arise in airline, bus, and railway companies are often modeled as set partitioning or set covering problems. The flight legs, respectively the trips that need to be assigned to pilots or drivers form the set S. The family F consists of all duties that are legal with respect to union and company regulations. Typically, the number of duties is not polynomially in the size of flight legs/trips that need to be covered. The huge number of variables motivates the use of column generation (see Section A.2.1) based solution techniques for these problems.

A.1.2 Multicommodity flow problem

In many applications one likes to route different commodities $k \in K$ through a shared network in the most efficient way. These problems can be formulated as multicommodity flow problems. We will state a basic version of the multicommodity flow problem (see Ahuja et al. (1993)). Let $G = (N, A)$ be a directed network with N as the set of nodes and A as the set of arcs. For every arc $(i, j) \in A$ we have cost c_{ij} per unit of flow on that arc. Moreover, we have an upper bound u_{ij} on the total flow on each arc $(i, j) \in A$. With every node $i \in N$ we associate an integer b_i. The value of b_i characterizes the nodes. If $b_i > 0$, node i is a supply node, if $b_i < 0$, node i is a demand node. Nodes with $b_i = 0$ are called transshipment nodes. Finally, let the decision variables x_{ij}^k specify the amount of flow of commodity k on arc (i,j). An linear programming (LP) formulation is as follows:

$$\min \quad \sum_{k\in K}\sum_{(i,j)\in A} c_{ij}x_{ij}^k \tag{A.4}$$

$$\text{s.t.} \quad \sum_{\{j:(i,j)\}\in A} x_{ij} - \sum_{\{j:(j,i)\}\in A} x_{ji} = b_i^k \qquad \forall i \in N, \forall k \in K \tag{A.5}$$

$$\sum_{k\in K} x_{ij} \leq u_{ij} \qquad \forall (i,j) \in A \tag{A.6}$$

$$x_{ij} \geq 0 \qquad \forall (i,j) \in A \tag{A.7}$$

Constraints (A.5) are referred to as flow or mass conservation constraints. That the joint flow of all commodities on an arc may not exceed its capacity is ensured by Constraints (A.6). Many practical applications require integer flows which can be obtained by modifying Constraints (A.7) accordingly. It is important to note that the decision version of the multicommodity integral flow problem is NP-complete (see Garey and Johnson (1979)) if there are two or more commodities.

A.1.3 Shortest path problem with resource constraints

In the classical shortest path problem one is interested in finding the shortest path from a source node s to a sink node t in an underlying directed graph $G = (N, A)$. The shortest path problem with resource constraints is an extension of this classical problem where one requires every feasible path to satisfy additional constraints. We will stick closely to the notation used in Irnich and Desaulniers (2005). These additional constraints are defined in terms of resource windows $[a_i^r, b_i^r]$ for all nodes $i \in N$ and all resources $r \in R$. Next to the cost c_{ij}, with every arc $(i,j) \in N$ we associate a (minimal) resource consumption t_{ij}^r for every resource $r \in R$. The change of the accumulated resource consumption for a given resource along an arc is specified by a so-called resource extension function $f_{ij}^r(T_i)$ that depends on the resource vector T_i which corresponds to the resource consumption accumulated along a path from s to i. Simple variants of resource extension functions are of the form $f_{ij}^r(T_i) = T_i^r + t_{ij}^r$. However, more general definitions of resource extension functions allow for example interdependencies of resources.

In general, shortest path problems with resource constraints are NP-hard. Garey and Johnson (1979) have shown that the decision version of the weight-constrained shortest path problem is NP-complete. This problem is a special case of a shortest path problem with resource constraints.

Shortest path problems with resource constraints can be solved by dynamic programming algorithms. If the underlying graph is acyclic, as it is in all problems considered in this thesis, we can order the nodes by reachability and use this order to obtain a label setting type of dynamic programming algorithm.

A.2 Selected solution techniques

A.2.1 Column Generation

The idea of column generation is to solve linear programs with a huge number of variables by iteratively considering two problems, a *restricted master problem* (RMP) and a *pricing problem*. The column generation algorithm stops if it can establish that the solution to RMP is also an optimal solution if all variables would be considered. We will illustrate this concept using an example. Let P be a linear program with a huge number of variables J.

$$\min \quad \sum_{j \in J} c_j x_j \tag{A.8}$$

$$\text{s.t.} \quad \sum_{j \in J} a_{ij} x_j = b_i \qquad \forall i \in M \tag{A.9}$$

$$x_j \geq 0 \qquad \forall j \in J \tag{A.10}$$

In order to solve P by column generation, we start with an initial RMP that contains only a subset of the variables $J' \subseteq J$ and the corresponding columns in the constraint matrix A given by its entries a_{ij} $\forall i \in M, j \in J$. We solve the RMP to obtain a (primal and) dual solution. Denote by u_i the value of the dual variables associated with Constraints (A.9) in the dual solution. It follows from the well known rules for the simplex algorithm (see e.g. Chvátal (1983)) that the solution to the RMP is also an optimal solution for P if all variables $j \in J$ have non-negative reduced cost. The reduced cost of a variable j is defined as $c_j - \sum_{i \in M} a_{ij} u_i$. The pricing problem

$$\min_{j \in J} \{c_j - \sum_{i \in M} a_{ij} u_i\}$$

checks if there exist variables $j \in J$ with negative reduced cost. If so, one or more of these variables and the corresponding columns of the constraint matrix are added to the RMP.

The idea of column generation (also known as Dantzig-Wolfe decomposition algorithm) was first presented in Ford and Fulkerson (1958) to solve a multicommodity flow problem by solving its *extensive formulation* via column generation. The idea was generalized by Dantzig and Wolfe (1960). Extensive formulations can be derived from *compact* formulations by using the Minkowski-Weyl theorem (see Nemhauser and Wolsey (1988)) that states that every non-empty convex polyhedron can be represented by a convex combination of its extreme points and a weighted combination of its extreme rays. The term extensive refers to the fact that the number of variables is huge compared to the amount of input data. In contrast to that, the number variables is polynomial in the *compact* formulation as presented in Section A.1.2. In the context of network flow problems, compact formulations are also known as arc based formulations whereas one refers to the extensive formulations as path based formulations.

Column generation has proven to lead to successful exact algorithms for integer programming problems if it is embedded in a branch-and-bound tree. The resulting algorithms are called branch-and-price algorithms (see Barnhart et al. (1998)). Desrochers and Soumis (1989) present the first branch-and-price algorithm for a crew scheduling problem. The theory behind column generation is discussed in Lübbecke and Desrosiers (2005). Many practical aspects can be found in Desaulniers et al. (2005). Desaulniers et al. (1998) provides many hints on implementation issues based on hands-on experience.

A.2.2 Lagrangian relaxation

Lagrangian relaxation is a technique to obtain bounds on the optimal solution value of constraint optimization problems. Without loss of generality we may assume that the optimization problem at hand is a minimization problem. In this case, Lagrangian relaxation can be used to compute lower bounds. The idea behind Lagrangian relaxation is to remove some constraints from an optimization problem and penalize their violation in the objective function. Let us give an example considering a general integer programming problem (P) with two sets of constraints (M_1 and M_2) given by Formulation (A.11)–(A.14).

$$\min \quad \sum_{j \in J} c_j x_j \tag{A.11}$$

$$\text{s.t.} \quad \sum_{j \in J} a_{ij} x_j = b_i \qquad \forall i \in M_1 \tag{A.12}$$

$$\sum_{j \in J} d_{kj} x_j = e_k \qquad \forall k \in M_2 \tag{A.13}$$

$$x_j \geq 0 \text{ and integer} \qquad \forall j \in J \tag{A.14}$$

A Lagrangian relaxation of problem P could for example be obtained by relaxing constraints M_1. Then, we introduce a *Lagrangian multiplier* λ_i for every constraint i in M_1. Let us denote by λ the vector of these multipliers. Given any choice for λ, the *Lagrangian subproblem* reads as follows:

$$\Phi(\lambda) = \quad \min \quad \sum_{j \in J} c_j x_j + \sum_{i \in M_1} \lambda_i (b_i - \sum_{j \in J} a_{ij} x_j) \tag{A.15}$$

$$\text{s.t.} \quad \sum_{j \in J} d_{kj} x_j = e_k \qquad \forall k \in M_2 \tag{A.16}$$

$$x_j \geq 0 \text{ and integer} \qquad \forall j \in J \tag{A.17}$$

The value of $\Phi(\lambda)$ is a lower bound on the optimal solution value $Z(P)$ of problem P for every λ. This can be argued as follows. Any feasible solution to the original problem P

is also a feasible solution to the Lagrangian subproblem. This holds also for any optimal solution to P. Moreover, the objective value of a feasible solution for problem P in the Lagrangian subproblem is $\sum_{j \in J} c_j x_j$, since $b_i - \sum_{j \in J} a_{ij} x_j = 0 \ \forall \ i \in M_1$. Therefore, $\Phi(\lambda)$ must be smaller or equal than $Z(P)$ for every λ. In the case of inequalities in Constraints (A.12), the Lagrangian multipliers must be restricted in sign in the Lagrangian subproblem. That is $\lambda_i \geq 0 \ \forall \ i \in M_1$ for "$\geq$" inequalities and $\lambda_i \leq 0 \ \forall \ i \in M_1$ for "$\leq$" inequalities.

As we have just argued, $\Phi(\lambda)$ gives a lower bound for every choice of λ. Naturally we are interested in the "best" bound. This can be found by maximizing the *Lagrangian function* $\Phi(\lambda)$. This optimization problem $Z(LR) = \max_\lambda \Phi(\lambda)$ is the so-called *Lagrangian dual problem*. The Lagrangian function $\Phi(\lambda)$ is a piecewise linear concave function. On its breakpoints, where the Lagrangian subproblem has multiple optimal solutions, it is only subdifferentiable. A very popular approach to solve $\max_\lambda \Phi(\lambda)$ is subgradient optimization. In the context of Lagrangian relaxation it was introduced by Held and Karp (1971). Several modifications to the original algorithm have been suggested. In our implementation we use a modified search direction as proposed by Camerini et al. (1975) and some of the modifications suggested in Beasley (1993). *Bundle methods* (Hiriart-Urruty and Lemaréchal (1993)) offer an alternative to subgradient optimization that have stronger convergence properties, but their iterations are computationally more expensive.

Let us now investigate the relation between the bound from the linear programming relaxation (LP-relaxation) $Z(LP)$ of P and the bound from Lagrangian relaxation $Z(LR)$ which was first stated by Geoffrion (1974). For a compact representation we use matrices and column vectors.

$$
\begin{aligned}
Z(LR) = & \max_\lambda \{ \min_x \ c^T x + (b - Ax)^T \lambda \\
& \quad \text{s.t. } Dx = e, \ x \geq 0 \text{ and integral} \} && \text{(A.18)} \\
\geq & \max_\lambda \{ \min_x \ c^T x + (b - Ax)^T \lambda \\
& \quad \text{s.t. } Dx = e, \ x \geq 0 \} && \text{(A.19)} \\
= & \max_\lambda \{ \min_x \ (c - A^T \lambda)^T x + b^T \lambda \\
& \quad \text{s.t. } Dx = e, \ x \geq 0 \} && \text{(A.20)} \\
= & \max_\lambda \ \max_\mu \{ \ b^T \lambda + e^T \mu \\
& \quad \text{s.t. } D^T \mu \leq c - A^T \lambda \} && \text{(A.21)} \\
= & \max_{\lambda, \mu} \ b^T \lambda + e^T \mu \\
& \quad \text{s.t. } A^T \lambda + D^T \mu \leq c && \text{(A.22)}
\end{aligned}
$$

$$
\begin{aligned}
&= \min \; c^T x \\
&\quad \text{s.t. } Ax = b, \; Dx = e, \; x \geq 0
\end{aligned} \tag{A.23}
$$

$$
= Z(LP) \tag{A.24}
$$

In (A.18) we state the Lagrangian dual problem. We drop the requirement on the x variables to be integer, hence the "$\geq$" relation, and obtain the optimization problem in (A.19). Note that the minimization problem in (A.19) is a linear programming problem. Rewriting its objective function leads to (A.20). Constructing the dual of the minimization problem gives (A.21). (A.22) is obtained by rearranging some terms. Applying linear programming duality leads to (A.23) which we recognize as the LP-relaxation of our original problem P. We therefore conclude the well known result $Z(LR) \geq Z(LP)$. Moreover, this inequality is an equality if the solution values of the minimization problems in (A.18) and (A.19) are the same. In this case the Lagrangian subproblem is said to have the *integrality property* since one can solve it by solving its LP-relaxation. We suggest the surveys of Fisher (1981) and Beasley (1993) as sources of additional information.

Lagrangian heuristics

Another reason for the popularity of Lagrangian relaxation is the fact that for most types of problems it is relatively easy to derive a heuristic that produces "good" feasible solutions for the original problem P based on Lagrangian multipliers. Most *Lagrangian heuristics* are based on one of two general ideas. Either a Lagrangian heuristic tries to construct a feasible solution based on an optimal solution to a Lagrangian subproblem, or they build feasible solutions from scratch guided by the Lagrangian multipliers.

Combining column generation and Lagrangian relaxation

In this section we will discuss the relation between Lagrangian relaxation and column generation. Moreover, we show how both techniques can be combined.

There exists a strong relation between Lagrangian relaxation and Dantzig-Wolfe decomposition. Assume that we have obtained an extensive formulation E as result of a Dantzig-Wolfe reformulation applied to a compact formulation C of a problem P. Furthermore, denote by $LP(E)$ the value of the LP-relaxation of E. One possible Lagrangian relaxation of C can be obtained by relaxing the set of constraints which are the linking constraints in the extensive formulation E. Let $LR(C)$ be the value of the corresponding Lagrangian dual problem $LDPC$. It is well known that the Lagrangian dual problem $LDPC$ and the LP-relaxation of the extensive formulation are dual to one another (see e.g. Geoffrion (1974) or Fisher (1981)). Moreover, the Lagrangian subproblem and the pricing problem of the column generation procedure to solve the LP-relaxation of E have the same constraints and differ only by a constant term in the objective function. Hence, solutions of the Lagrangian subproblem can also be added to the restricted master problem of the column generation procedure. Barahona and Jensen (1998) and Degraeve and Jans (2007) present hybrid solution methods that dynamically switch between solving the Lagrangian dual problem of the Lagrangian relaxation applied to the compact formulation and solving the LP-relaxation of the extended formulation by column generation. The outline of such a hybrid method as well as a general discussion can be found in Huisman et al. (2005a).

Another possibility to combine Lagrangian relaxation and column generation is to apply Lagrangian relaxation to the restricted master problem in order to obtain an approximate dual solution. Note that for the column generation algorithm only a dual solution to the restricted master problem is necessary. Algorithms based on this combination of Lagrangian relaxation and column generation have been presented for several crew scheduling applications. We refer to Borndörfer et al. (2003), Huisman et al. (2005c), and Steinzen et al. (2010) for crew scheduling at bus companies, Abbink et al. (2005) for crew scheduling at railways, and Barahona and Anbil (2000) and Subramanian and Sherali (2008) for airline crew scheduling.

Bibliography

E. Abbink, D. Mobach, P.-J. Fioole, L.G. Kroon, E. van der Heijden, and N. Wijngaards. Actor-Agent Application for Train Driver Rescheduling. In *Proceedings of the Eighth International Conference on Autonomous Agents and Mulitagent Systems*, 2009.

E.J.W. Abbink, M. Fischetti, L.G. Kroon, G. Timmer, and M.J.C.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 E. Goutham. An integrated decision support tool for airline schedule recovery during irregular operations. *European Journal of Operational Research*, 185:825–848, 2008.

R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, 1993.

A.A. Assad. Models for rail transportation. *Transportation Research A*, 14:205–220, 1980.

F. Barahona and R. Anbil. The volume algorithm: Producing primal solutions with a subgradient method. *Mathematical Programming*, 87:385–399, 2000.

F. Barahona and D. Jensen. Plant location with minimum inventory. *Mathematical Programming*, 83:101–111, 1998.

C. Barnhart, A.M. Cohn, E.L. Johnson, D. Klabjan, G.L. Nemhauser, and P.H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1998.

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

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

R. Borndörfer, M. Grötschel, and A. Löbel. Duty Scheduling in Public Transit. In W. Jäger and H.-J. Krebs, editors, *Mathematics - Key Technology for the Future*, pages 653–674. Springer, Berlin, 2003.

K. Boubaker, G. Desaulniers, and I. Elhallaoui. Bidline scheduling with equity by heuristic dynamic constraint aggregation. *Transportation Research B*, 44:50–61, 2010.

G. Budai, G. Maróti, R. Dekker, D. Huisman, and L.G. Kroon. Rescheduling in passenger railways: The rolling stock rebalancing problem. *Journal of Scheduling*, 2009. DOI 10.1007/s10951-009-0133-91.

P. Camerini, L Fratta, and F. Maffioli. On improving relaxation methods by modified gradient techniques. *Mathematical Programming Study*, 3:26–34, 1975.

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

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

V. Chvátal. *Linear Programming.* W.H. Freeman and Company, New York, 1983.

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.

J.-F. Cordeau, P. Toth, and D. Vigo. A survey of optimization models for train routing and scheduling. *Transportation Science*, 32:380–404, 1998.

G.B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

A. D'Ariano. *Improving Real-Time Train Dispatching: Models, Algorithms and Applications.* PhD thesis, Technical University Delft, 2008.

A. D'Ariano, M. Pranzo, and I.A. Hansen. Conflict Resolution and Train Speed Coordination for Solving Real-Time Timetable Perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 8:208–222, 2007.

Z. Degraeve and R. Jans. A new Dantzig-Wolfe Reformulation and Branch-and-Price Algorithm for the Capacitated Lot-Sizing Problem with Setup Times. *Operations Research*, 55:909–920, 2007.

G. Desaulniers, J. Desrosiers, I. Ioachim, M.M. Solomon, F. Soumis, and D. Villeneuve. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In T. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 57–93. Kluwer, Boston, 1998.

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

G. Desaulniers, J. Desrosiers, and M.M. Solomon, editors. *Column Generation*. Springer, New York, 2005.

M. Desrochers and F. Soumis. A Column Generation Approach to the Urban Transit Crew Scheduling Problem. *Transportation Science*, 23:1–13, 1989.

M. Ehrgott and D.M. Ryan. Constructing Robust Crew Schedules with Bicriteria Optimization. *Journal of Multi-Criteria Decision Analysis*, 11:139–150, 2002.

I. Elhallaoui, D. Villeneuve, F. Soumis, and G. Desaulniers. Dynamic Aggregation of Set-Partitioning Constraints in Column Generation. *Operations Research*, 53:632–645, 2005.

I. Elhallaoui, M. Abdelmoutalib, F. Soumis, and G. Desaulniers. Multi-phase dynamic constraint aggrgation for set partitioning type problems. *Mathematical Programming*, 44:50–61, 2008. 10.1007/s10107-008-0254-5.

P.-J. Fioole, L.G. 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.

L.R. Ford and D.R. Fulkerson. A Suggested Computation for Maximal Multi-Commodity Network Flows. *Management Science*, 5:97–101, 1958.

M.R. Garey and D.S. Johnson. *Computer and Intractability: A Guide to NP-Completeness*. Freeman, San Francisco, 1979.

A.M. Geoffrion. Lagrangean Relaxation for Integer Programming. *Mathematical Programming Studies*, 2:82–114, 1974.

F. Geraets, L.G. Kroon, A. Schöbel, D. Wagner, and C. Zaroliagis. *Algorithmic Methods for Railway Optimization*, volume 4359 of *Lecture Notes in Computer Science*. Springer, Berlin, 2007.

M. Held and R.M. Karp. The travalling salesman problem and minimum spanning trees: Part II. *Mathematical Programming*, 1:6–25, 1971.

J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and Minimization Algorithms, Part 2: Advanced Theory and Bundle Methods*. Springer, Berlin, 1993.

K. Holmberg and D. Yuan. A Lagrangian Heuristic Based Branch-and-Bound Approach for the Capacitated Network Design Problem. *Operations Research*, 48:461–481, 2000.

D. Huisman. A column generation approach for the rail crew re-scheduling problem. *European Journal of Operational Research*, 180:163–173, 2007.

D. Huisman, R. Jans, M. Peters, and A.P.M. Wagelmans. Combining Column Generation and Lagrangian Relaxation. In G. Desaulniers, J. Desrosiers, and M.M. Solomon, editors, *Column Generation*, GERAD 25th Anniversary Series, pages 247–270. Springer, New York, 2005a.

D. Huisman, L.G. Kroon, R.M. Lentink, and M.J.C.M. Vromans. Operations Research in passenger railway transportation. *Statistica Neerlandica*, 59:467–497, 2005b.

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

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. *Decision Support for the Rolling Stock Dispatcher*. PhD thesis, Technical University of Denmark, Lyngby, Denmark, 2009.

J. Jespersen-Groth, J. Clausen, and J. Larsen. Optimal reinsertion of cancelled train lines. Technical Report IMM-TR-2001-15, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2006.

J. Jespersen-Groth, J. Clausen, and J. Larsen. Rolling Stock Recovery Problem. Technical Report IMM-TR-2008-18, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2008.

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, New York, 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.W. Wallace. *Stochastic Programming*. John Wiley & Sons, Chichester, 1994.

R.M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

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.

W.K. Klein Haneveld and M.H. Van der Vlerk. Stochastic integer programming: General models and algorithms. *Annals of Operations Research*, 85:39–57, 1999.

N. Kohl, A. Larsen, J. Larsen, A. Ross, and S. Tiourine. Airline Disruption Management - Perspectives, Experiences and Outlook. *Journal of Air Transport Management*, 13: 147–162, 2007.

L.G. Kroon and D. Huisman. Algorithmic Support for Railway Disruption Management. Technical Report EI 2009-50, Econometric Intsitute, Erasmus University Rotterdam, 2009.

L.G. 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. Lettovský, E.L. Johnson, and G.L. Nemhauser. Airline Crew Recovery. *Transportation Science*, 34:337–348, 2000.

C. Liebchen, M.E. Lübbecke, R.H. Möhring, and S. Stiller. Recoverable Robusness. Technical Report ARRIVAL-TR-0066, Algorithms for Robust and online Railway optimization: Improving the Validity and reliAbility of Large scale systems (ARRIVAL), 2007.

C. Liebchen, M.E. Lübbecke, R.H. 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.

M.E. Lübbecke and J. Desrosiers. Selected Topics in Column Generation. *Operations Research*, 53:1007–1023, 2005.

H.M. Markowitz. Portfolio Selection. *Journal of Finance*, 7:77–91, 1952.

C.P. Medard and N. Sawhney. Airline crew scheduling from planning to operations. *European Journal of Operational Research*, 183:1013–1027, 2007.

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

D. Mobach, E. Abbink, P.-J. Fioole, R. Lentink, L.G. Kroon, E. van der Heijden, and N. Wijngaards. Train Driver Rescheduling using Task-Exchange Teams. In *Proceedings of the Second International Workshop on Optimisation in Multi-Agent Systems (OPTMAS) at AAMAS 2009*, 2009.

E. Morgado and J. Martins. CREWS_NS - scheduling train crews in the Netherlands. *AI Magazine*, 19:25–38, 1998.

G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, Inc., New York, 1988.

Netherlands Railways. Nederlandse spoorwegen - jaarverslag 2009, 2010. URL `http://www.ns.nl/cs/Satellite/jaarverslag/duurzaam/resultatenvervoer`.

L.K. Nielsen. A Decision Support Framework for Rolling Stock Rescheduling. Technical Report ARRIVAL-TR-0158, Algorithms for Robust and online Railway optimization: Improving the Validity and reliAbility of Large scale systems (ARRIVAL), 2008.

R. Nissen and K. Haase. Duty-period-based network model for crew rescheduling in European airlines. *Journal of Scheduling*, 9:255–278, 2006.

A.-S. Pepin, G. Desaulniers, A. Hertz, and D. Huisman. Comparison of heuristic approaches for the multiple depot vehicle scheduling problem. *Journal of Scheduling*, 12: 17–30, 2009.

D. Potthoff, D. Huisman, and G. Desaulniers. Column Generation with Dynamic Duty Selection for Railway Crew Rescheduling. *Transportation Science*, 2010. DOI 10.1287/trsc.1100.0322.

E. Prescott-Gagnon, G. Desaulniers, and L.-M. Rousseau. A Branch-and-Price-Based Large Neighborhood Search Algorithm for the Vehicle Routing Problem with Time Windows. *Networks*, 54:190–204, 2009.

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 method. *Computers & Operations Research*, 37:845–856, 2010.

S. Ropke and D. Pisinger. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, 40:455–472, 2006.

J.M. Rosenberger, E.L. Johnson, and G.L. Nemhauser. Rerouting Aircraft for Airline Recovery. *Transportation Science*, 37:408–421, 2003.

RTL. ProRail veroorzaakte problemen spoor, 2010. URL `http://www.rtl.nl/(/actueel/rtlnieuws/binnenland/)/components/actueel/rtlnieuws/2010/01_januari/14/binnenland/prorail_veroorzaakte_problemen_spoor.xml`. accessed 2010-01-25.

A.J. Schaefer, E.L. Johnson, A.J. Kleywegt, and G.L. Nemhauser. Airline Crew Scheduling Under Uncertainty. *Transportation Science*, 39:340–348, 2005.

A. Schöbel. *Optimization in Public Transportation*, volume 3 of *Springer Optimization and Its Applications*. Springer, Berlin, 2006.

A. Schöbel. Integer programming approaches for solving the delay management problem. In F. Geraets, L.G. Kroon, A. Schöbel, D. Wagner, and C. Zaroliagis, editors, *Algorithmic Methods for Railway Optimization*, volume 4359 of *Lecture Notes in Computer Science*, pages 145–170. Springer, Berlin, 2007.

A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., New York, 1986.

S. Shebalov and D. Klabjan. Robust Airline Crew Pairing: Move-up Crews. *Transportation Science*, 40:300–312, 2006.

I. Steinzen, V. Gintner, L. Suhl, and N. Kliewer. A Time-Space Network Approach for the Integrated Vehicle- an Crew-Scheduling Problem with Muliple Depots. *Transportation Science*, 2010. doi 10.1287/trsc.1090.0304.

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.

M. Stojković, F. Soumis, and J. Desrosiers. The Operational Airline Crew Scheduling Problem. *Transportation Science*, 32:232–245, 1998.

S. Subramanian and H.D. Sherali. An Effective Deflected Subgradient Optimization Scheme for Implementing Column Generation for Large-Scale Airline Crew Scheduling Problems. *INFORMS Journal on Computing*, 20:565–578, 2008.

J. Törnquist. *Railway traffic disturbance management*. PhD thesis, Blekinge Institute of Technology, Karlskrona, 2006.

L.P. Veelenturf, D. Potthoff, D. Huisman, and L.G. Kroon. Railway Crew Rescheduling with Retiming. Technical Report EI 2009-24, Econometric Institute, Erasmus University Rotterdam, 2009.

D. Villeneuve. *Logiciel de génération de Colonnes*. PhD thesis, Université de Montréal, Montreal, Quebec, Canada, 1999.

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.

O. Weide, D.M. Ryan, and M. Ehrgott. An iterative approach to robust and integrated aircraft routing and crew scheduling. *Computers & Operations Research*, 37:833–844, 2010.

J.W. Yen and J.R. Birge. A Stochastic Programming Approach to the Airline Crew Scheduling Problem. *Transportation Science*, 40(6):3–14, 2006.

G. Yu and X. Qi. *Disruption Management: Framework, Models and Applications*. World Scientific Publishing Company, 2004.

G. Yu, M. Argüello, Song G., S.M. McCowan, and A. White. A New Era for Crew Recovery at Continental Airlines. *Interfaces*, 33:5–22, 2003.

Nederlandse samenvatting (Summary in Dutch)

Op een gemiddelde werkdag vervoert de Nederlandse Spoorwegen (NS) ongeveer 1,2 miljoen reizigers. Om de gepubliceerde dienstregeling uit te voeren maakt NS een gedetailleerde materieel- en personeelsplanning. Iedere dag kunnen er echter onverwachte gebeurtenissen plaatsvinden die de dienstregeling verstoren. Hierbij kan men bijvoorbeeld denken aan storingen aan de infrastructuur, defect materieel of extreme weersomstandigheden. In deze situaties is vaak een deel van de infrastructuur voor een bepaalde tijd niet of slechts beperkt beschikbaar. Dit resulteert in grote aanpassingen aan de dienstregeling waaronder het opheffen van treinen. Als deze veranderingen in de dienstregeling leiden tot conflicten in de materieel- en personeelsplanning moet men deze plannen herstellen oftewel bijsturen. Het is belangrijk voor een spoorvervoerder om dit te doen zodanig dat er niet nog extra treinen uitvallen. Op dit moment gebeurt deze bijsturing nog steeds handmatig door bijstuurders. Het belangrijkste doel van dit proefschrift is om de fundering te leggen voor het ontwikkelen van een beslissingsondersteunend systeem voor de bijsturing van rijdend personeel bij verstoringen.

In hoofdstuk 2 van dit proefschrift beschrijven we het bijsturingsproces. Vanwege de complexiteit en de verschillende verantwoordelijkheden tussen infrastructuurbeheerder en vervoerders wordt dit proces in drie stappen verdeeld. In de eerste stap wordt de dienstregeling aangepast zodanig dat de nieuwe dienstregeling kan worden uitgevoerd onder de gewijzigde omstandigheden. Deze dienstregelingswijzigingen volgen meestal uit van te voren vastgestelde versperringsmaatregelen. Het aanpassen van de dienstregeling is de verantwoordelijkheid van de infrastructuurbeheerder (ProRail in Nederland). De tweede stap in het bijsturingsproces is het bijsturen van het materieelplan. Alhoewel we graag opmerken dat een betere materieelbijsturing mogelijk moet zijn met behulp van wiskundige modellen, valt dat buiten de scope van dit proefschrift. De derde stap is de bijsturing van het rijdend personeel (machinisten en conducteurs). Voor dit probleem, formeel in het Engels aangeduid als het “operational crew rescheduling problem” ofwel OCRSP, wordt als input gebruik gemaakt van de gewijzigde dienstregeling en materieelomlopen. Het doel van het OCRSP is om alle conflicten in de personeelsdiensten op te lossen en de diensten zodanig te wijzigen dat zo veel mogelijk treinen van personeel worden voorzien. Hierbij dient rekening te worden gehouden met een groot aantal arbeidsregels.

In hoofdstuk 3 wordt een nieuwe oplossingsmethode gepresenteerd om het OCRSP op te lossen. Gegeven het operationele karakter van het probleem, moet zo'n oplossingsmethode binnen enkele minuten rekentijd een goede oplossing vinden. Onze oplossingsmethode, genaamd CGDDS, start met het beschouwen van een initieel kernprobleem dat alleen een deel van de originele diensten bevat. We hebben voor deze benadering gekozen, omdat veel diensten niet zullen veranderen als gevolg van de verstoring. Bijvoorbeeld omdat ze geografisch verder van de verstoring vandaan zitten. Het initiële kernprobleem wordt vervolgens opgelost met een methode gebaseerd op kolomgeneratie. Kolomgeneratie is een geavanceerde wiskundige techniek die ontwikkeld is om wiskundige problemen met enorme aantallen beslissingsvariabelen efficiënt op te lossen. Personeelsplannings- en bijsturingsproblemen hebben een enorm aantal beslissingsvariabelen, omdat iedere mogelijke dienst gerepresenteerd wordt door een beslissingsvariabele, en het aantal mogelijke diensten loopt bij grote bedrijven als NS al snel in de vele miljoenen. Het oplossen van het initiële kernprobleem geeft vaak al een goede oplossing maar soms zijn enkele treinen nog niet van een personeelslid voorzien. Daarom controleren we of alle treinen in het bijgestuurde personeelsplan zitten. Voor iedere trein waarbij dit niet het geval is, maken we een nieuwe kernprobleem met de bedoeling om deze specifieke trein wel in te plannen. We doen dit door voor dit nieuwe kernprobleem diensten te selecteren die in de "buurt" liggen van de trein die we beschouwen. In dit hoofdstuk tonen we ook aan dat we met de CGDDS methode snel goede oplossingen vinden voor verschillende data instanties van NS. Hiervoor hebben we gebruik gemaakt van 10 instanties die allemaal gebaseerd zijn op verstoringen die in het verleden hebben plaatsgevonden. In deze instanties waren er tussen de 15 en 59 diensten met conflicten. De grootste rekentijd lag op ongeveer 4 minuten. Deze rekentijd was nodig voor een versperring tussen Utrecht en Amsterdam met een duur van 3 uur. Na enkele iteraties konden in 9 instanties alle treinen ingepland worden als er gebruik werd gemaakt van de reservediensten in het plan. Zonder reservediensten lukte dat in 7 van de 10 instanties. In 2 andere gevallen was er 1, in het slechtste geval waren er 3 treinen die niet konden worden ingepland.

Om de kwaliteit van de methode te evalueren hebben we in hoofdstuk 4 de CGDDS methode vergeleken met twee andere methoden. De eerste methode is een 2-fase heuristische benadering (2P-RSPPRC) die het handmatige bijsturingsproces probeert na te bootsen. De tweede methode is een heuristiek die gebruik maakt van dynamische aggregatie van restricties (DCA). Dit is een geavanceerde methode die gebruik maakt van kolomgeneratie, maar waarbij de restricties in het wiskundig model geclusterd worden met als doel het reduceren van de rekentijd. Een vergelijking tussen de drie oplossingsmethoden laat zien dat de 2P-RSPPRC heuristiek alleen in staat is goede oplossingen te genereren voor heel makkelijke instanties. Bovendien bleek dat de DCA methode niet altijd beter presteert dan de klassieke kolomgeneratie methode. Concluderend kunnen we stellen dat onze CGDDS methode het best presteert kijkend naar zowel de kwaliteit van de oplossing als de rekentijd.

In hoofdstuk 5 bekijken we een uitbreiding van het OCRSP waarbij we kleine vertragingen van enkele minuten bij het vertrek van enkele treinen toestaan. Deze uitbreiding is gebaseerd op de observatie dat kleine wijzigingen in de dienstregelingen soms leiden tot betere oplossingen voor het bijsturen van rijdend personeel. Dit kan o.a. afgeleid worden uit het volgende voorbeeld met twee treinen. De eerste trein heeft als eindpunt het beginpunt van de tweede trein. Gegeven de geplande aankomst- en vertrektijden van deze treinen is het een machinist niet toegestaan om beide treinen te rijden, omdat anders de regel m.b.t. de minimale overgangstijd wordt geschonden. Als de tweede trein echter enkele - zeg 3 - minuten later vertrekt, wordt deze regel niet meer geschonden en kan één machinist beide treinen rijden. Dit betekent dat we een dienst kunnen construeren die niet toegestaan zou zijn als we de dienstregeling niet zouden wijzigingen. Door de verzameling van toegelaten diensten te vergroten, zijn er meer toegelaten oplossingen mogelijk. Hierdoor is het mogelijk om betere oplossingen te vinden vergeleken met de situatie waarbij de dienstregeling niet mag worden gewijzigd. Natuurlijk heeft het vertragen van treinen ook een ongewenst effect voor de passagiers. Daarom beboeten we in de doelstellingsfunctie het vertragen van treinen zodanig dat we dit alleen doen als we significant betere oplossingen vinden voor het personeelsbijsturingsprobleem, d.w.z. als we door het vertragen van enkele treinen andere treinen niet hoeven uit te vallen. Het uitgebreide OCRSP model waarbij het toegestaan is om te schuiven met de vertrektijden van enkele treinen heeft veel meer restricties dan het basismodel. Deze extra restricties zijn nodig om de propagatie van de vertragingen door te berekenen en om te garanderen dat alle diensten die in de oplossing worden gekozen ook passen bij de gekozen vertragingen. In dit hoofdstuk laten we zien hoe de CGDDS methode kan worden aangepast zodanig dat we ook dit uitgebreide model kunnen oplossen. De rekenexperimenten laten zien dat we door het toestaan van kleine vertragingen in 4 van de 6 cases betere oplossingen kunnen krijgen, d.w.z. dat er minder treinen uitvallen. Bovendien blijkt de toename in de rekentijd beperkt te zijn wat het mogelijk maakt om deze uitbreiding ook in de praktijk toe te passen.

In de eerste hoofdstukken van het proefschrift is nog verondersteld dat de duur van de verstoringen bekend is. In hoofdstuk 6 laten we deze veronderstelling los. De onzekerheid in de duur van de verstoring kan in het OCRSP worden gezien als onzekerheid hoe de gewijzigde dienstregeling er uit gaat zien. Er is echter maar een beperkt aantal mogelijke dienstregelingen. Het personeelsbijsturingsprobleem onder onzekerheid kan daarom worden gezien als een 2-fase optimalisatieprobleem. In de eerste fase wordt het OCRSP opgelost zodanig dat een personeelsplan wordt berekend voor het meest optimistische scenario voor de duur van de verstoring. In de tweede fase is het bekend welke dienstregeling uiteindelijk wordt gereden en is het eventueel nodig om het OCRSP opnieuw op te lossen. In deze tweede fase is de oplossing van de eerste fase input. Dit hoofdstuk heeft twee doelen. Ten eerste analyseren we wat er gebeurt als we in de eerste fase geen rekening houden met de onzekerheid in de duur van de verstoring. Vanzelfsprekend zijn er dan wel correcties nodig indien de daadwerkelijke duur van de verstoring afwijkt van de geschatte duur. Deze aanpak zouden we een naïeve aanpak kunnen noemen. Ten tweede beschouwen we twee *quasi robuuste* oplossingsmethoden die al rekening houden met de onzekerheid in de eerste fase. Het concept van quasi robuustheid is een nieuw concept dat probeert de tekortkomingen van bekende concepten voor het optimaliseren onder onzekerheid zoals robuuste optimalisatie en stochastisch programmeren teniet te doen. Het hoofdidee achter onze quasi robuuste optimaliseringsaanpak is om de mogelijke beslissingen te beperken tot diegene die garanderen dat alle treinen onafhankelijk van welke dienstregeling uiteindelijk gereden wordt, van personeel zijn voorzien. We tonen in dit hoofdstuk aan dat een quasi robuuste model benadert kan worden met een op kolomgeneratie gebaseerde heuristiek. Bovendien laten we zien dat een van de twee quasi robuuste oplossingsmethoden in 4 van de 5 gevallen beter presteert dan de naïeve aanpak. We concluderen dan ook dat het nuttig is om rekening te houden met de onzekerheid in de verstoring bij het oplossen van het OCRSP. Echter de quasi robuuste oplossingsmethoden vragen veel meer rekentijd en zijn nu nog niet snel genoeg voor de praktijk.

Tenslotte eindigen we in hoofdstuk 7 met het opsommen van de belangrijkste resultaten uit dit proefschrift. Bovendien geven we als advies aan de Nederlandse Spoorwegen om zo snel mogelijk een beslissingsondersteunend systeem voor de bijsturing van rijdend personeel te introduceren gebaseerd op de methoden uit dit proefschrift. Naar onze overtuiging zal dit leiden tot betere operationele prestaties van NS: namelijk minder vertraagde en minder uitgevallen treinen!

Curriculum Vitae

Daniel Potthoff (1976) obtained his master's degree in Business Information Systems from the University of Paderborn, Germany in 2005. In the same year he started with his PhD research at Erasmus University Rotterdam. His main research interest is in Operations Research, and in particular in algorithms for railway crew rescheduling. Daniel conducted his research in cooperation with the Logistics department of Netherlands Railways. He presented his research on conferences on three continents. In addition, parts of his research lead to a book chapter and to a paper published in *Transportation Science*. Moreover, he was awarded the second prize in the 2008 Informs Railroad Applications Section Student Paper Award. The prize was presented at the 2008 Informs Annual Meeting in Washington, DC. After the completion of his thesis, Daniel continued working on railway crew rescheduling at Netherlands Railways. Since September 2010, he works as a consultant for Ab Ovo Germany.

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT (ERIM)

ERIM PH.D. SERIES

RESEARCH IN MANAGEMENT

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

Acciaro, M., *Bundling Strategies in Global Supply Chains,* Promotor: Prof.dr. H.E. Haralambides, EPS-2010-197-LIS, ISBN: 978-90-5892-240-3, http://hdl.handle.net/1765/1

Agatz, N.A.H., *Demand Management in E-Fulfillment*, Promotor: Prof.dr.ir. J.A.E.E. van Nunen, EPS-2009-163-LIS, ISBN: 978-90-5892-200-7, http://hdl.handle.net/1765/15425

Althuizen, N.A.P., *Analogical Reasoning as a Decision Support Principle for Weakly Structured Marketing Problems*, Promotor: Prof.dr.ir. B. Wierenga, EPS-2006-095-MKT, ISBN: 90-5892-129-8, http://hdl.handle.net/1765/8190

Alvarez, H.L., *Distributed Collaborative Learning Communities Enabled by Information Communication Technology*, Promotor: Prof.dr. K. Kumar, EPS-2006-080-LIS, ISBN: 90-5892-112-3, http://hdl.handle.net/1765/7830

Appelman, J.H., *Governance of Global Interorganizational Tourism Networks: Changing Forms of Co-ordination between the Travel Agency and Aviation Sector*, Promotors: Prof.dr. F.M. Go & Prof.dr. B. Nooteboom, EPS-2004-036-MKT, ISBN: 90-5892-060-7, http://hdl.handle.net/1765/1199

Asperen, E. van, *Essays on Port, Container, and Bulk Chemical Logistics Optimization*, Promotor: Prof.dr.ir. R. Dekker, EPS-2009-181-LIS, ISBN: 978-90-5892-222-9, http://hdl.handle.net/1765/1

Assem, M.J. van den, *Deal or No Deal? Decision Making under Risk in a Large-Stake TV Game Show and Related Experiments*, Promotor: Prof.dr. J. Spronk, EPS-2008-138-F&A, ISBN: 978-90-5892-173-4, http://hdl.handle.net/1765/13566

Baquero, G, *On Hedge Fund Performance, Capital Flows and Investor Psychology*, Promotor: Prof.dr. M.J.C.M. Verbeek, EPS-2006-094-F&A, ISBN: 90-5892-131-X, http://hdl.handle.net/1765/8192

Berens, G., *Corporate Branding: The Development of Corporate Associations and their Influence on Stakeholder Reactions*, Promotor: Prof.dr. C.B.M. van Riel, EPS-2004-039-ORG, ISBN: 90-5892-065-8, http://hdl.handle.net/1765/1273

Berghe, D.A.F. van den, *Working Across Borders: Multinational Enterprises and the Internationalization of Employment*, Promotors: Prof.dr. R.J.M. van Tulder & Prof.dr. E.J.J. Schenk, EPS-2003-029-ORG, ISBN: 90-5892-05-34, http://hdl.handle.net/1765/1041

Berghman, L.A., *Strategic Innovation Capacity: A Mixed Method Study on Deliberate Strategic Learning Mechanisms*, Promotor: Prof.dr. P. Mattyssens, EPS-2006-087-MKT, ISBN: 90-5892-120-4, http://hdl.handle.net/1765/7991

Bezemer, P.J., *Diffusion of Corporate Governance Beliefs: Board Independence and the Emergence of a Shareholder Value Orientation in the Netherlands,* Promotors: Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2009-192-STR, ISBN: 978-90-5892-232-8, http://hdl.handle.net/1765/1

Bijman, W.J.J., *Essays on Agricultural Co-operatives: Governance Structure in Fruit and Vegetable Chains*, Promotor: Prof.dr. G.W.J. Hendrikse, EPS-2002-015-ORG, ISBN: 90-5892-024-0, http://hdl.handle.net/1765/867

Bispo, A., *Labour Market Segmentation: An investigation into the Dutch hospitality industry*, Promotors: Prof.dr. G.H.M. Evers & Prof.dr. A.R. Thurik, EPS-2007-108-ORG, ISBN: 90-5892-136-9, http://hdl.handle.net/1765/10283

Blindenbach-Driessen, F., *Innovation Management in Project-Based Firms*, Promotor: Prof.dr. S.L. van de Velde, EPS-2006-082-LIS, ISBN: 90-5892-110-7, http://hdl.handle.net/1765/7828

Boer, C.A., *Distributed Simulation in Industry,* Promotors: Prof.dr. A. de Bruin & Prof.dr.ir. A. Verbraeck, EPS-2005-065-LIS, ISBN: 90-5892-093-3, http://hdl.handle.net/1765/6925

Boer, N.I., *Knowledge Sharing within Organizations: A situated and Relational Perspective,* Promotor: Prof.dr. K. Kumar, EPS-2005-060-LIS, ISBN: 90-5892-086-0, http://hdl.handle.net/1765/6770

Boer-Sorbán, K., *Agent-Based Simulation of Financial Markets: A modular, Continuous-Time Approach,* Promotor: Prof.dr. A. de Bruin, EPS-2008-119-LIS, ISBN: 90-5892-155-0, http://hdl.handle.net/1765/10870

Boon, C.T., *HRM and Fit: Survival of the Fittest!?,* Promotors: Prof.dr. J. Paauwe & Prof.dr. D.N. den Hartog, EPS-2008-129-ORG, ISBN: 978-90-5892-162-8, http://hdl.handle.net/1765/12606

Braun, E., *City Marketing: Towards an Integrated Approach,* Promotor: Prof.dr. L. van den Berg, EPS-2008-142-MKT, ISBN: 978-90-5892-180-2, http://hdl.handle.net/1765/13694

Brito, M.P. de, *Managing Reverse Logistics or Reversing Logistics Management?* Promotors: Prof.dr.ir. R. Dekker & Prof.dr. M. B. M. de Koster, EPS-2004-035-LIS, ISBN: 90-5892-058-5, http://hdl.handle.net/1765/1132

Brohm, R., *Polycentric Order in Organizations: A Dialogue between Michael Polanyi and IT-Consultants on Knowledge, Morality, and Organization,* Promotors: Prof.dr. G. W. J. Hendrikse & Prof.dr. H. K. Letiche, EPS-2005-063-ORG, ISBN: 90-5892-095-X, http://hdl.handle.net/1765/6911

Brumme, W.-H., *Manufacturing Capability Switching in the High-Tech Electronics Technology Life Cycle,* Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. L.N. Van Wassenhove, EPS-2008-126-LIS, ISBN: 978-90-5892-150-5, http://hdl.handle.net/1765/12103

Budiono, D.P., The Analysis of Mutual Fund Performance: Evidence from U.S. Equity Mutual Funds, Promotor: Prof.dr.ir. M.J.C.M. Verbeek, EPS-2010-185-F&A, ISBN: 978-90-5892-224-3, http://hdl.handle.net/1765/1

Burgers, J.H., *Managing Corporate Venturing: Multilevel Studies on Project Autonomy, Integration, Knowledge Relatedness, and Phases in the New Business Development Process,* Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2008-136-STR, ISBN: 978-90-5892-174-1, http://hdl.handle.net/1765/13484

Campbell, R.A.J., *Rethinking Risk in International Financial Markets*, Promotor: Prof.dr. C.G. Koedijk, EPS-2001-005-F&A, ISBN: 90-5892-008-9, http://hdl.handle.net/1765/306

Carvalho de Mesquita Ferreira, L., *Attention Mosaics: Studies of Organizational Attention*, Promotors: Prof.dr. P.M.A.R. Heugens & Prof.dr. J. van Oosterhout, EPS-2010-205-ORG, ISBN: 978-90-5892-242-7, http://hdl.handle.net/1765/1

Chen, C.-M., *Evaluation and Design of Supply Chain Operations Using DEA*, Promotor: Prof.dr. J.A.E.E. van Nunen, EPS-2009-172-LIS, ISBN: 978-90-5892-209-0, http://hdl.handle.net/1765/1

Chen, H., *Individual Mobile Communication Services and Tariffs*, Promotor: Prof.dr. L.F.J.M. Pau, EPS-2008-123-LIS, ISBN: 90-5892-158-1, http://hdl.handle.net/1765/11141

Chen, Y., *Labour Flexibility in China's Companies: An Empirical Study*, Promotors: Prof.dr. A. Buitendam & Prof.dr. B. Krug, EPS-2001-006-ORG, ISBN: 90-5892-012-7, http://hdl.handle.net/1765/307

Damen, F.J.A., *Taking the Lead: The Role of Affect in Leadership Effectiveness*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2007-107-ORG, http://hdl.handle.net/1765/10282

Daniševská, P., *Empirical Studies on Financial Intermediation and Corporate Policies*, Promotor: Prof.dr. C.G. Koedijk, EPS-2004-044-F&A, ISBN: 90-5892-070-4, http://hdl.handle.net/1765/1518

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, ISBN: 978-90-5892-245-8, http://hdl.handle.net/1765/1

Delporte-Vermeiren, D.J.E., *Improving the Flexibility and Profitability of ICT-enabled Business Networks: An Assessment Method and Tool*, Promotors: Prof. mr. dr. P.H.M. Vervest & Prof.dr.ir. H.W.G.M. van Heck, EPS-2003-020-LIS, ISBN: 90-5892-040-2, http://hdl.handle.net/1765/359

Derwall, J.M.M., *The Economic Virtues of SRI and CSR*, Promotor: Prof.dr. C.G. Koedijk, EPS-2007-101-F&A, ISBN: 90-5892-132-8, http://hdl.handle.net/1765/8986

Diepen, M. van, *Dynamics and Competition in Charitable Giving*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2009-159-MKT, ISBN: 978-90-5892-188-8, http://hdl.handle.net/1765/14526

Dietz, H.M.S., *Managing (Sales)People towards Perfomance: HR Strategy, Leadership & Teamwork*, Promotor: Prof.dr. G.W.J. Hendrikse, EPS-2009-168-ORG, ISBN: 978-90-5892-210-6, http://hdl.handle.net/1765/1

Dijksterhuis, M., *Organizational Dynamics of Cognition and Action in the Changing Dutch and US Banking Industries*, Promotors: Prof.dr.ir. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2003-026-STR, ISBN: 90-5892-048-8, http://hdl.handle.net/1765/1037

Eijk, A.R. van der, *Behind Networks: Knowledge Transfer, Favor Exchange and Performance*, Promotors: Prof.dr. S.L. van de Velde & Prof.dr.drs. W.A. Dolfsma, EPS-2009-161-LIS, ISBN: 978-90-5892-190-1, http://hdl.handle.net/1765/14613

Elstak, M.N., *Flipping the Identity Coin: The Comparative Effect of Perceived, Projected and Desired Organizational Identity on Organizational Identification and Desired Behavior*, Promotor: Prof.dr. C.B.M. van Riel, EPS-2008-117-ORG, ISBN: 90-5892-148-2, http://hdl.handle.net/1765/10723

Erken, H.P.G., *Productivity, R&D and Entrepreneurship*, Promotor: Prof.dr. A.R. Thurik, EPS-2008-147-ORG, ISBN: 978-90-5892-179-6, http://hdl.handle.net/1765/14004

Fenema, P.C. van, *Coordination and Control of Globally Distributed Software Projects*, Promotor: Prof.dr. K. Kumar, EPS-2002-019-LIS, ISBN: 90-5892-030-5, http://hdl.handle.net/1765/360

Fleischmann, M., *Quantitative Models for Reverse Logistics*, Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. R. Dekker, EPS-2000-002-LIS, ISBN: 35-4041-711-7, http://hdl.handle.net/1765/1044

Flier, B., *Strategic Renewal of European Financial Incumbents: Coevolution of Environmental Selection, Institutional Effects, and Managerial Intentionality*, Promotors: Prof.dr.ir. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2003-033-STR, ISBN: 90-5892-055-0, http://hdl.handle.net/1765/1071

Fok, D., *Advanced Econometric Marketing Models*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2003-027-MKT, ISBN: 90-5892-049-6, http://hdl.handle.net/1765/1035

Ganzaroli, A., *Creating Trust between Local and Global Systems*, Promotors: Prof.dr. K. Kumar & Prof.dr. R.M. Lee, EPS-2002-018-LIS, ISBN: 90-5892-031-3, http://hdl.handle.net/1765/361

Gertsen, H.F.M., *Riding a Tiger without Being Eaten: How Companies and Analysts Tame Financial Restatements and Influence Corporate Reputation*, Promotor: Prof.dr. C.B.M. van Riel, EPS-2009-171-ORG, ISBN: 90-5892-214-4, http://hdl.handle.net/1765/1

Gilsing, V.A., *Exploration, Exploitation and Co-evolution in Innovation Networks*, Promotors: Prof.dr. B. Nooteboom & Prof.dr. J.P.M. Groenewegen, EPS-2003-032-ORG, ISBN: 90-5892-054-2, http://hdl.handle.net/1765/1040

Gijsbers, G.W., *Agricultural Innovation in Asia: Drivers, Paradigms and Performance,* Promotor: Prof.dr. R.J.M. van Tulder, EPS-2009-156-ORG, ISBN: 978-90-5892-191-8, http://hdl.handle.net/1765/14524

Gong, Y., *Stochastic Modelling and Analysis of Warehouse Operations*, Promotors: Prof.dr. M.B.M. de Koster & Prof.dr. S.L. van de Velde, EPS-2009-180-LIS, ISBN: 978-90-5892-219-9, http://hdl.handle.net/1765/1

Govers, R., *Virtual Tourism Destination Image: Glocal Identities Constructed, Perceived and Experienced*, Promotors: Prof.dr. F.M. Go & Prof.dr. K. Kumar, EPS-2005-069-MKT, ISBN: 90-5892-107-7, http://hdl.handle.net/1765/6981

Graaf, G. de, *Tractable Morality: Customer Discourses of Bankers, Veterinarians and Charity Workers*, Promotors: Prof.dr. F. Leijnse & Prof.dr. T. van Willigenburg, EPS-2003-031-ORG, ISBN: 90-5892-051-8, http://hdl.handle.net/1765/1038

Greeven, M.J., *Innovation in an Uncertain Institutional Environment: Private Software Entrepreneurs in Hangzhou, China*, Promotor: Prof.dr. B. Krug, EPS-2009-164-ORG, ISBN: 978-90-5892-202-1, http://hdl.handle.net/1765/15426

Groot, E.A. de, *Essays on Economic Cycles*, Promotors: Prof.dr. Ph.H.B.F. Franses & Prof.dr. H.R. Commandeur, EPS-2006-091-MKT, ISBN: 90-5892-123-9, http://hdl.handle.net/1765/8216

Guenster, N.K., *Investment Strategies Based on Social Responsibility and Bubbles*, Promotor: Prof.dr. C.G. Koedijk, EPS-2008-175-F&A, ISBN: 978-90-5892-206-9, http://hdl.handle.net/1765/1

Gutkowska, A.B., *Essays on the Dynamic Portfolio Choice*, Promotor: Prof.dr. A.C.F. Vorst, EPS-2006-085-F&A, ISBN: 90-5892-118-2, http://hdl.handle.net/1765/7994

Hagemeijer, R.E., *The Unmasking of the Other*, Promotors: Prof.dr. S.J. Magala & Prof.dr. H.K. Letiche, EPS-2005-068-ORG, ISBN: 90-5892-097-6, http://hdl.handle.net/1765/6963

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/1765/1

Halderen, M.D. van, *Organizational Identity Expressiveness and Perception Management: Principles for Expressing the Organizational Identity in Order to Manage the Perceptions and Behavioral Reactions of External Stakeholders*, Promotor: Prof.dr. S.B.M. van Riel, EPS-2008-122-ORG, ISBN: 90-5892-153-6, http://hdl.handle.net/1765/10872

Hartigh, E. den, *Increasing Returns and Firm Performance: An Empirical Study*, Promotor: Prof.dr. H.R. Commandeur, EPS-2005-067-STR, ISBN: 90-5892-098-4, http://hdl.handle.net/1765/6939

Hensmans, M., *A Republican Settlement Theory of the Firm: Applied to Retail Banks in England and the Netherlands (1830-2007)*, Promotors: Prof.dr. A. Jolink& Prof.dr. S.J. Magala, EPS-2010-193-ORG, ISBN 90-5892-235-9, http://hdl.handle.net/1765/1

Hermans. J.M., *ICT in Information Services; Use and Deployment of the Dutch Securities Trade, 1860-1970,* Promotor: Prof.dr. drs. F.H.A. Janszen, EPS-2004-046-ORG, ISBN 90-5892-072-0, http://hdl.handle.net/1765/1793

Hernandez Mireles, C., *Marketing Modeling for New Products,* Promotor: Prof.dr. P.H. Franses, EPS-2010-202-MKT, ISBN 90-5892-237-3, http://hdl.handle.net/1765/1

Hessels, S.J.A., *International Entrepreneurship: Value Creation Across National Borders*, Promotor: Prof.dr. A.R. Thurik, EPS-2008-144-ORG, ISBN: 978-90-5892-181-9, http://hdl.handle.net/1765/13942

Heugens, P.P.M.A.R., *Strategic Issues Management: Implications for Corporate Performance*, Promotors: Prof.dr.ir. F.A.J. van den Bosch & Prof.dr. C.B.M. van Riel, EPS-2001-007-STR, ISBN: 90-5892-009-9, http://hdl.handle.net/1765/358

Heuvel, W. van den, *The Economic Lot-Sizing Problem: New Results and Extensions*, Promotor: Prof.dr. A.P.L. Wagelmans, EPS-2006-093-LIS, ISBN: 90-5892-124-7, http://hdl.handle.net/1765/1805

Hoedemaekers, C.M.W., *Performance, Pinned down: A Lacanian Analysis of Subjectivity at Work*, Promotors: Prof.dr. S. Magala & Prof.dr. D.H. den Hartog, EPS-2008-121-ORG, ISBN: 90-5892-156-7, http://hdl.handle.net/1765/10871

Hooghiemstra, R., *The Construction of Reality: Cultural Differences in Self-serving Behaviour in Accounting Narratives*, Promotors: Prof.dr. L.G. van der Tas RA & Prof.dr. A.Th.H. Pruyn, EPS-2003-025-F&A, ISBN: 90-5892-047-X, http://hdl.handle.net/1765/871

Hu, Y., *Essays on the Governance of Agricultural Products: Cooperatives and Contract Farming*, Promotors: Prof.dr. G.W.J. Hendrkse & Prof.dr. B. Krug, EPS-2007-113-ORG, ISBN: 90-5892-145-1, http://hdl.handle.net/1765/10535

Huang, X., *An Analysis of Occupational Pension Provision: From Evaluation to Redesigh,* Promotors: Prof.dr. M.J.C.M. Verbeek & Prof.dr. R.J. Mahieu, EPS-2010-196-F&A, ISBN: 90-5892-239-7, http://hdl.handle.net/1765/1

Huij, J.J., *New Insights into Mutual Funds: Performance and Family Strategies*, Promotor: Prof.dr. M.C.J.M. Verbeek, EPS-2007-099-F&A, ISBN: 90-5892-134-4, http://hdl.handle.net/1765/9398

Huurman, C.I., *Dealing with Electricity Prices*, Promotor: Prof.dr. C.D. Koedijk, EPS-2007-098-F&A, ISBN: 90-5892-130-1, http://hdl.handle.net/1765/9399

Iastrebova, K, *Manager's Information Overload: The Impact of Coping Strategies on Decision-Making Performance*, Promotor: Prof.dr. H.G. van Dissel, EPS-2006-077-LIS, ISBN: 90-5892-111-5, http://hdl.handle.net/1765/7329

Iwaarden, J.D. van, *Changing Quality Controls: The Effects of Increasing Product Variety and Shortening Product Life Cycles,* Promotors: Prof.dr. B.G. Dale & Prof.dr. A.R.T. Williams, EPS-2006-084-ORG, ISBN: 90-5892-117-4, http://hdl.handle.net/1765/7992

Jansen, J.J.P., *Ambidextrous Organizations*, Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2005-055-STR, ISBN: 90-5892-081-X, http://hdl.handle.net/1765/6774

Jaspers, F.P.H., *Organizing Systemic Innovation*, Promotor: Prof.dr.ir. J.C.M. van den Ende, EPS-2009-160-ORG, ISBN: 978-90-5892-197-), http://hdl.handle.net/1765/14974

Jennen, M.G.J., *Empirical Essays on Office Market Dynamics*, Promotors: Prof.dr. C.G. Koedijk & Prof.dr. D. Brounen, EPS-2008-140-F&A, ISBN: 978-90-5892-176-5, http://hdl.handle.net/1765/13692

Jiang, T., *Capital Structure Determinants and Governance Structure Variety in Franchising*, Promotors: Prof.dr. G. Hendrikse & Prof.dr. A. de Jong, EPS-2009-158-F&A, ISBN: 978-90-5892-199-4, http://hdl.handle.net/1765/14975

Jiao, T., *Essays in Financial Accounting*, Promotor: Prof.dr. G.M.H. Mertens, EPS-2009-176-F&A, ISBN: 978-90-5892-211-3, http://hdl.handle.net/1765/1

Jong, C. de, *Dealing with Derivatives: Studies on the Role, Informational Content and Pricing of Financial Derivatives*, Promotor: Prof.dr. C.G. Koedijk, EPS-2003-023-F&A, ISBN: 90-5892-043-7, http://hdl.handle.net/1765/1043

Kaa, G. van, *Standard Battles for Complex Systems: Empirical Research on the Home Network,* Promotors: Prof.dr.ir. J. van den Ende & Prof.dr.ir. H.W.G.M. van Heck, EPS-2009-166-ORG, ISBN: 978-90-5892-205-2, http://hdl.handle.net/1765/1

Kagie, M., *Advances in Online Shopping Interfaces: Product Catalog Maps and Recommender Systems,* Promotor: Prof.dr. P.J.F. Groenen, EPS-2010-195-MKT, ISBN: 978-90-5892-233-5, http://hdl.handle.net/1765/1

Keizer, A.B., *The Changing Logic of Japanese Employment Practices: A Firm-Level Analysis of Four Industries,* Promotors: Prof.dr. J.A. Stam & Prof.dr. J.P.M. Groenewegen, EPS-2005-057-ORG, ISBN: 90-5892-087-9, http://hdl.handle.net/1765/6667

Kijkuit, R.C., *Social Networks in the Front End: The Organizational Life of an Idea,* Promotor: Prof.dr. B. Nooteboom, EPS-2007-104-ORG, ISBN: 90-5892-137-6, http://hdl.handle.net/1765/10074

Kippers, J., *Empirical Studies on Cash Payments,* Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2004-043-F&A, ISBN: 90-5892-069-0, http://hdl.handle.net/1765/1520

Klein, M.H., *Poverty Alleviation through Sustainable Strategic Business Models: Essays on Poverty Alleviation as a Business Strategy,* Promotor: Prof.dr. H.R. Commandeur, EPS-2008-135-STR, ISBN: 978-90-5892-168-0, http://hdl.handle.net/1765/13482

Knapp, S., *The Econometrics of Maritime Safety: Recommendations to Enhance Safety at Sea*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2007-096-ORG, ISBN: 90-5892-127-1, http://hdl.handle.net/1765/7913

Kole, E., *On Crises, Crashes and Comovements*, Promotors: Prof.dr. C.G. Koedijk & Prof.dr. M.J.C.M. Verbeek, EPS-2006-083-F&A, ISBN: 90-5892-114-X, http://hdl.handle.net/1765/7829

Kooij-de Bode, J.M., *Distributed Information and Group Decision-Making: Effects of Diversity and Affect*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2007-115-ORG, http://hdl.handle.net/1765/10722

Koppius, O.R., *Information Architecture and Electronic Market Performance*, Promotors: Prof.dr. P.H.M. Vervest & Prof.dr.ir. H.W.G.M. van Heck, EPS-2002-013-LIS, ISBN: 90-5892-023-2, http://hdl.handle.net/1765/921

Kotlarsky, J., *Management of Globally Distributed Component-Based Software Development Projects,* Promotor: Prof.dr. K. Kumar, EPS-2005-059-LIS, ISBN: 90-5892-088-7, http://hdl.handle.net/1765/6772

Krauth, E.I., *Real-Time Planning Support: A Task-Technology Fit Perspective,* Promotors: Prof.dr. S.L. van de Velde & Prof.dr. J. van Hillegersberg, EPS-2008-155-LIS, ISBN: 978-90-5892-193-2, http://hdl.handle.net/1765/14521

Kuilman, J., *The Re-Emergence of Foreign Banks in Shanghai: An Ecological Analysis,* Promotor: Prof.dr. B. Krug, EPS-2005-066-ORG, ISBN: 90-5892-096-8, http://hdl.handle.net/1765/6926

Kwee, Z., *Investigating Three Key Principles of Sustained Strategic Renewal: A Longitudinal Study of Long-Lived Firms,* Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2009-174-STR, ISBN: 90-5892-212-0, http://hdl.handle.net/1765/1

Langen, P.W. de, *The Performance of Seaport Clusters: A Framework to Analyze Cluster Performance and an Application to the Seaport Clusters of Durban, Rotterdam and the Lower Mississippi*, Promotors: Prof.dr. B. Nooteboom & Prof. drs. H.W.H. Welters, EPS-2004-034-LIS, ISBN: 90-5892-056-9, http://hdl.handle.net/1765/1133

Le Anh, T., *Intelligent Control of Vehicle-Based Internal Transport Systems*, Promotors: Prof.dr. M.B.M. de Koster & Prof.dr.ir. R. Dekker, EPS-2005-051-LIS, ISBN: 90-5892-079-8, http://hdl.handle.net/1765/6554

Le-Duc, T., *Design and Control of Efficient Order Picking Processes*, Promotor: Prof.dr. M.B.M. de Koster, EPS-2005-064-LIS, ISBN: 90-5892-094-1, http://hdl.handle.net/1765/6910

Leeuwen, E.P. van, *Recovered-Resource Dependent Industries and the Strategic Renewal of Incumbent Firm: A Multi-Level Study of Recovered Resource Dependence Management and Strategic Renewal in the European Paper and Board Industry*, Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2007-109-STR, ISBN: 90-5892-140-6, http://hdl.handle.net/1765/10183

Lentink, R.M., *Algorithmic Decision Support for Shunt Planning*, Promotors: Prof.dr. L.G. Kroon & Prof.dr.ir. J.A.E.E. van Nunen, EPS-2006-073-LIS, ISBN: 90-5892-104-2, http://hdl.handle.net/1765/7328

Li, T., *Informedness and Customer-Centric Revenue Management,* Promotors: Prof.dr. P.H.M. Vervest & Prof.dr.ir. H.W.G.M. van Heck, EPS-2009-146-LIS, ISBN: 978-90-5892-195-6, http://hdl.handle.net/1765/14525

Liang, G., *New Competition: Foreign Direct Investment and Industrial Development in China,* Promotor: Prof.dr. R.J.M. van Tulder, EPS-2004-047-ORG, ISBN: 90-5892-073-9, http://hdl.handle.net/1765/1795

Liere, D.W. van, *Network Horizon and the Dynamics of Network Positions: A Multi-Method Multi-Level Longitudinal Study of Interfirm Networks,* Promotor: Prof.dr. P.H.M. Vervest, EPS-2007-105-LIS, ISBN: 90-5892-139-0, http://hdl.handle.net/1765/10181

Loef, J., *Incongruity between Ads and Consumer Expectations of Advertising*, Promotors: Prof.dr. W.F. van Raaij & Prof.dr. G. Antonides, EPS-2002-017-MKT, ISBN: 90-5892-028-3, http://hdl.handle.net/1765/869

Maeseneire, W., de, *Essays on Firm Valuation and Value Appropriation,* Promotor: Prof.dr. J.T.J. Smit, EPS-2005-053-F&A, ISBN: 90-5892-082-8, http://hdl.handle.net/1765/6768

Londoño, M. del Pilar, *Institutional Arrangements that Affect Free Trade Agreements: Economic Rationality Versus Interest Groups*, Promotors: Prof.dr. H.E. Haralambides & Prof.dr. J.F. Francois, EPS-2006-078-LIS, ISBN: 90-5892-108-5, http://hdl.handle.net/1765/7578

Maas, A.A., van der, *Strategy Implementation in a Small Island Context: An Integrative Framework,* Promotor: Prof.dr. H.G. van Dissel, EPS-2008-127-LIS, ISBN: 978-90-5892-160-4, http://hdl.handle.net/1765/12278

Maas, K.E.G., *Corporate Social Performance: From Output Measurement to Impact Measurement,* Promotor: Prof.dr. H.R. Commandeur, EPS-2009-182-STR, ISBN: 978-90-5892-225-0, http://hdl.handle.net/1765/1

Maeseneire, W., de, *Essays on Firm Valuation and Value Appropriation,* Promotor: Prof.dr. J.T.J. Smit, EPS-2005-053-F&A, ISBN: 90-5892-082-8, http://hdl.handle.net/1765/6768

Mandele, L.M., van der, *Leadership and the Inflection Point: A Longitudinal Perspective*, Promotors: Prof.dr. H.W. Volberda & Prof.dr. H.R. Commandeur, EPS-2004-042-STR, ISBN: 90-5892-067-4, http://hdl.handle.net/1765/1302

Meer, J.R. van der, *Operational Control of Internal Transport*, Promotors: Prof.dr. M.B.M. de Koster & Prof.dr.ir. R. Dekker, EPS-2000-001-LIS, ISBN: 90-5892-004-6, http://hdl.handle.net/1765/859

Mentink, A., *Essays on Corporate Bonds*, Promotor: Prof.dr. A.C.F. Vorst, EPS-2005-070-F&A, ISBN: 90-5892-100-X, http://hdl.handle.net/1765/7121

Meyer, R.J.H., *Mapping the Mind of the Strategist: A Quantitative Methodology for Measuring the Strategic Beliefs of Executives*, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2007-106-ORG, ISBN: 978-90-5892-141-3, http://hdl.handle.net/1765/10182

Miltenburg, P.R., *Effects of Modular Sourcing on Manufacturing Flexibility in the Automotive Industry: A Study among German OEMs*, Promotors: Prof.dr. J. Paauwe & Prof.dr. H.R. Commandeur, EPS-2003-030-ORG, ISBN: 90-5892-052-6, http://hdl.handle.net/1765/1039

Moerman, G.A., *Empirical Studies on Asset Pricing and Banking in the Euro Area,* Promotor: Prof.dr. C.G. Koedijk, EPS-2005-058-F&A, ISBN: 90-5892-090-9, http://hdl.handle.net/1765/6666

Moitra, D., *Globalization of R&D: Leveraging Offshoring for Innovative Capability and Organizational Flexibility,* Promotor: Prof.dr. K. Kumar, EPS-2008-150-LIS, ISBN: 978-90-5892-184-0, http://hdl.handle.net/1765/14081

Mol, M.M., *Outsourcing, Supplier-relations and Internationalisation: Global Source Strategy as a Chinese Puzzle*, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2001-010-ORG, ISBN: 90-5892-014-3, http://hdl.handle.net/1765/355

Mom, T.J.M., *Managers' Exploration and Exploitation Activities: The Influence of Organizational Factors and Knowledge Inflows*, Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2006-079-STR, ISBN: 90-5892-116-6, http://hdl.handle.net/1765

Moonen, J.M., *Multi-Agent Systems for Transportation Planning and Coordination*, Promotors: Prof.dr. J. van Hillegersberg & Prof.dr. S.L. van de Velde, EPS-2009-177-LIS, ISBN: 978-90-5892-216-8, http://hdl.handle.net/1

Mulder, A., *Government Dilemmas in the Private Provision of Public Goods*, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2004-045-ORG, ISBN: 90-5892-071-2, http://hdl.handle.net/1765/1790

Muller, A.R., *The Rise of Regionalism: Core Company Strategies Under The Second Wave of Integration*, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2004-038-ORG, ISBN: 90-5892-062-3, http://hdl.handle.net/1765/1272

Nalbantov G.I., *Essays on Some Recent Penalization Methods with Applications in Finance and Marketing*, Promotor: Prof. dr P.J.F. Groenen, EPS-2008-132-F&A, ISBN: 978-90-5892-166-6, http://hdl.handle.net/1765/13319

Nederveen Pieterse, A., *Goal Orientation in Teams: The Role of Diversity*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2009-162-ORG, http://hdl.handle.net/1765/15240

Nguyen, T.T., *Capital Structure, Strategic Competition, and Governance*, Promotor: Prof.dr. A. de Jong, EPS-2008-148-F&A, ISBN: 90-5892-178-9, http://hdl.handle.net/1765/14005

Niesten, E.M.M.I., *Regulation, Governance and Adaptation: Governance Transformations in the Dutch and French Liberalizing Electricity Industries*, Promotors: Prof.dr. A. Jolink & Prof.dr. J.P.M. Groenewegen, EPS-2009-170-ORG, ISBN: 978-90-5892-208-3, http://hdl.handle.net/1765/1

Nieuwenboer, N.A. den, *Seeing the Shadow of the Self*, Promotor: Prof.dr. S.P. Kaptein, EPS-2008-151-ORG, ISBN: 978-90-5892-182-6, http://hdl.handle.net/1765/14223

Ning, H., *Hierarchical Portfolio Management: Theory and Applications*, Promotor: Prof.dr. J. Spronk, EPS-2007-118-F&A, ISBN: 90-5892-152-9, http://hdl.handle.net/1765/10868

Noeverman, J., *Management Control Systems, Evaluative Style, and Behaviour: Exploring the Concept and Behavioural Consequences of Evaluative Style*, Promotors: Prof.dr. E.G.J. Vosselman & Prof.dr. A.R.T. Williams, EPS-2007-120-F&A, ISBN: 90-5892-151-2, http://hdl.handle.net/1765/10869

Nuijten, I., *Servant Leadership: Paradox or Diamond in the Rough? A Multidimensional Measure and Empirical Evidence*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2009-183-ORG, http://hdl.handle.net/1765/1

Oosterhout, J., van, *The Quest for Legitimacy: On Authority and Responsibility in Governance*, Promotors: Prof.dr. T. van Willigenburg & Prof.mr. H.R. van Gunsteren, EPS-2002-012-ORG, ISBN: 90-5892-022-4, http://hdl.handle.net/1765/362

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, ISBN: 90-5092-236-6, http://hdl.handle.net/1765/1

Oostrum, J.M., van, *Applying Mathematical Models to Surgical Patient Planning*, Promotor: Prof.dr. A.P.M. Wagelmans, EPS-2009-179-LIS, ISBN: 978-90-5892-217-5, http://hdl.handle.net/1765/1

Paape, L., *Corporate Governance: The Impact on the Role, Position, and Scope of Services of the Internal Audit Function*, Promotors: Prof.dr. G.J. van der Pijl & Prof.dr. H. Commandeur, EPS-2007-111-MKT, ISBN: 90-5892-143-7, http://hdl.handle.net/1765/10417

Pak, K., *Revenue Management: New Features and Models*, Promotor: Prof.dr.ir. R. Dekker, EPS-2005-061-LIS, ISBN: 90-5892-092-5, http://hdl.handle.net/1765/362/6771

Pattikawa, L.H, *Innovation in the Pharmaceutical Industry: Evidence from Drug Introduction in the U.S.*, Promotors: Prof.dr. H.R.Commandeur, EPS-2007-102-MKT, ISBN: 90-5892-135-2, http://hdl.handle.net/1765/9626

Peeters, L.W.P., *Cyclic Railway Timetable Optimization*, Promotors: Prof.dr. L.G. Kroon & Prof.dr.ir. J.A.E.E. van Nunen, EPS-2003-022-LIS, ISBN: 90-5892-042-9, http://hdl.handle.net/1765/429

Pietersz, R., *Pricing Models for Bermudan-style Interest Rate Derivatives*, Promotors: Prof.dr. A.A.J. Pelsser & Prof.dr. A.C.F. Vorst, EPS-2005-071-F&A, ISBN: 90-5892-099-2, http://hdl.handle.net/1765/7122

Pince, C., *Advances in Inventory Management: Dynamic Models,* Promotor: Prof.dr.ir. R. Dekker, EPS-2010-199-LIS, ISBN: 978-90-5892-243-4, http://hdl.handle.net/1765/1

Poel, A.M. van der, *Empirical Essays in Corporate Finance and Financial Reporting*, Promotors: Prof.dr. A. de Jong & Prof.dr. G.M.H. Mertens, EPS-2007-133-F&A, ISBN: 978-90-5892-165-9, http://hdl.handle.net/1765/13320

Popova, V., *Knowledge Discovery and Monotonicity*, Promotor: Prof.dr. A. de Bruin, EPS-2004-037-LIS, ISBN: 90-5892-061-5, http://hdl.handle.net/1765/1201

Pouchkarev, I., *Performance Evaluation of Constrained Portfolios*, Promotors: Prof.dr. J. Spronk & Dr. W.G.P.M. Hallerbach, EPS-2005-052-F&A, ISBN: 90-5892-083-6, http://hdl.handle.net/1765/6731

Prins, R., *Modeling Consumer Adoption and Usage of Value-Added Mobile Services*, Promotors: Prof.dr. Ph.H.B.F. Franses & Prof.dr. P.C. Verhoef, EPS-2008-128-MKT, ISBN: 978/90-5892-161-1, http://hdl.handle.net/1765/12461

Puvanasvari Ratnasingam, P., *Interorganizational Trust in Business to Business E-Commerce*, Promotors: Prof.dr. K. Kumar & Prof.dr. H.G. van Dissel, EPS-2001-009-LIS, ISBN: 90-5892-017-8, http://hdl.handle.net/1765/356

Quak, H.J., *Sustainability of Urban Freight Transport: Retail Distribution and Local Regulation in Cities*, Promotor: Prof.dr. M.B.M. de Koster, EPS-2008-124-LIS, ISBN: 978-90-5892-154-3, http://hdl.handle.net/1765/11990

Quariguasi Frota Neto, J., *Eco-efficient Supply Chains for Electrical and Electronic Products*, Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. H.W.G.M. van Heck, EPS-2008-152-LIS, ISBN: 978-90-5892-192-5, http://hdl.handle.net/1765/14785

Radkevitch, U.L, *Online Reverse Auction for Procurement of Services*, Promotor: Prof.dr.ir. H.W.G.M. van Heck, EPS-2008-137-LIS, ISBN: 978-90-5892-171-0, http://hdl.handle.net/1765/13497

Rinsum, M. van, *Performance Measurement and Managerial Time Orientation*, Promotor: Prof.dr. F.G.H. Hartmann, EPS-2006-088-F&A, ISBN: 90-5892-121-2, http://hdl.handle.net/1765/7993

Roelofsen, E.M., *The Role of Analyst Conference Calls in Capital Markets*, Promotors: Prof.dr. G.M.H. Mertens & Prof.dr. L.G. van der Tas RA, EPS-2010-190-F&A, ISBN: 978-90-5892-228-1, http://hdl.handle.net/1765/1

Romero Morales, D., *Optimization Problems in Supply Chain Management*, Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Dr. H.E. Romeijn, EPS-2000-003-LIS, ISBN: 90-9014078-6, http://hdl.handle.net/1765/865

Roodbergen, K.J., *Layout and Routing Methods for Warehouses*, Promotors: Prof.dr. M.B.M. de Koster & Prof.dr.ir. J.A.E.E. van Nunen, EPS-2001-004-LIS, ISBN: 90-5892-005-4, http://hdl.handle.net/1765/861

Rook, L., *Imitation in Creative Task Performance*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2008-125-ORG, http://hdl.handle.net/1765/11555

Rosmalen, J. van, *Segmentation and Dimension Reduction: Exploratory and Model-Based Approaches*, Promotor: Prof.dr. P.J.F. Groenen, EPS-2009-165-MKT, ISBN: 978-90-5892-201-4, http://hdl.handle.net/1765/15536

Rus, D., *The Dark Side of Leadership: Exploring the Psychology of Leader Self-serving Behavior*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2009-178-ORG, http://hdl.handle.net/1765/1

Samii, R., *Leveraging Logistics Partnerships: Lessons from Humanitarian Organizations,* Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. L.N. Van Wassenhove, EPS-2008-153-LIS, ISBN: 978-90-5892-186-4, http://hdl.handle.net/1765/14519

Schaik, D. van, *M&A in Japan: An Analysis of Merger Waves and Hostile Takeovers,* Promotors: Prof.dr. J. Spronk & Prof.dr. J.P.M. Groenewegen, EPS-2008-141-F&A, ISBN: 978-90-5892-169-7, http://hdl.handle.net/1765/13693

Schauten, M.B.J., *Valuation, Capital Structure Decisions and the Cost of Capital,* Promotors: Prof.dr. J. Spronk & Prof.dr. D. van Dijk, EPS-2008-134-F&A, ISBN: 978-90-5892-172-7, http://hdl.handle.net/1765/13480

Schramade, W.L.J., *Corporate Bonds Issuers,* Promotor: Prof.dr. A. De Jong, EPS-2006-092-F&A, ISBN: 90-5892-125-5, http://hdl.handle.net/1765/8191

Schweizer, T.S., *An Individual Psychology of Novelty-Seeking, Creativity and Innovation,* Promotor: Prof.dr. R.J.M. van Tulder, EPS-2004-048-ORG, ISBN: 90-5892-077-1, http://hdl.handle.net/1765/1818

Six, F.E., *Trust and Trouble: Building Interpersonal Trust Within Organizations*, Promotors: Prof.dr. B. Nooteboom & Prof.dr. A.M. Sorge, EPS-2004-040-ORG, ISBN: 90-5892-064-X, http://hdl.handle.net/1765/1271

Slager, A.M.H., *Banking across Borders*, Promotors: Prof.dr. R.J.M. van Tulder & Prof.dr. D.M.N. van Wensveen, EPS-2004-041-ORG, ISBN: 90-5892-066–6, http://hdl.handle.net/1765/1301

Sloot, L., *Understanding Consumer Reactions to Assortment Unavailability*, Promotors: Prof.dr. H.R. Commandeur, Prof.dr. E. Peelen & Prof.dr. P.C. Verhoef, EPS-2006-074-MKT, ISBN: 90-5892-102-6, http://hdl.handle.net/1765/7438

Smit, W., *Market Information Sharing in Channel Relationships: Its Nature, Antecedents and Consequences*, Promotors: Prof.dr.ir. G.H. van Bruggen & Prof.dr.ir. B. Wierenga, EPS-2006-076-MKT, ISBN: 90-5892-106-9, http://hdl.handle.net/1765/7327

Sonnenberg, M., *The Signalling Effect of HRM on Psychological Contracts of Employees: A Multi-level Perspective*, Promotor: Prof.dr. J. Paauwe, EPS-2006-086-ORG, ISBN: 90-5892-119-0, http://hdl.handle.net/1765/7995

Sotgiu, F., *Not All Promotions are Made Equal: From the Effects of a Price War to Cross-chain Cannibalization,* Promotors: Prof.dr. M.G. Dekimpe & Prof.dr.ir. B. Wierenga, EPS-2010-203-MKT, ISBN: 978-90-5892-238-0, http://hdl.handle.net/1765/1

Speklé, R.F., *Beyond Generics: A closer Look at Hybrid and Hierarchical Governance*, Promotor: Prof.dr. M.A. van Hoepen RA, EPS-2001-008-F&A, ISBN: 90-5892-011-9, http://hdl.handle.net/1765/357

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/1765/1

Stam, D.A., *Managing Dreams and Ambitions: A Psychological Analysis of Vision Communication*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2008-149-ORG, http://hdl.handle.net/1765/14080

Stienstra, M., *Strategic Renewal in Regulatory Environments: How Inter- and Intra-organisational Institutional Forces Influence European Energy Incumbent Firms,* Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2008-145-STR, ISBN: 978-90-5892-184-0, http://hdl.handle.net/1765/13943

Sweldens, S.T.L.R., *Evaluative Conditioning 2.0: Direct versus Associative Transfer of Affect to Brands,* Promotor: Prof.dr. S.M.J. van Osselaer, EPS-2009-167-MKT, ISBN: 978-90-5892-204-5, http://hdl.handle.net/1765/1

Szkudlarek, B.A.*, Spinning the Web of Reentry: [Re]connecting reentry training theory and practice*, Promotor: Prof.dr. S.J. Magala, EPS-2008-143-ORG, ISBN: 978-90-5892-177-2, http://hdl.handle.net/1765/13695

Tempelaar, M.P., *Organizing for Ambidexterity: Studies on the Pursuit of Exploration and Exploitation through Differentiation, Integration, Contextual and Individual Attributes,* Promotors: Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-191-STR, ISBN: 978-90-5892-231-1, http://hdl.handle.net/1765/1

Teunter, L.H., *Analysis of Sales Promotion Effects on Household Purchase Behavior*, Promotors: Prof.dr.ir. B. Wierenga & Prof.dr. T. Kloek, EPS-2002-016-MKT, ISBN: 90-5892-029-1, http://hdl.handle.net/1765/868

Tims, B., *Empirical Studies on Exchange Rate Puzzles and Volatility*, Promotor: Prof.dr. C.G. Koedijk, EPS-2006-089-F&A, ISBN: 90-5892-113-1, http://hdl.handle.net/1765/8066

Tiwari, V., *Transition Process and Performance in IT Outsourcing: Evidence from a Field Study and Laboratory Experiments,* Promotors: Prof.dr.ir. H.W.G.M. van Heck & Prof.dr. P.H.M. Vervest, EPS-2010-201-LIS, ISBN: 978-90-5892-241-0, http://hdl.handle.net/1765/1

Tuk, M.A., *Is Friendship Silent When Money Talks? How Consumers Respond to Word-of-Mouth Marketing*, Promotors: Prof.dr.ir. A. Smidts & Prof.dr. D.H.J. Wigboldus, EPS-2008-130-MKT, ISBN: 978-90-5892-164-2, http://hdl.handle.net/1765/12702

Valck, K. de, *Virtual Communities of Consumption: Networks of Consumer Knowledge and Companionship*, Promotors: Prof.dr.ir. G.H. van Bruggen & Prof.dr.ir. B. Wierenga, EPS-2005-050-MKT, ISBN: 90-5892-078-X, http://hdl.handle.net/1765/6663

Valk, W. van der, *Buyer-Seller Interaction Patterns During Ongoing Service Exchange*, Promotors: Prof.dr. J.Y.F. Wynstra & Prof.dr.ir. B. Axelsson, EPS-2007-116-MKT, ISBN: 90-5892-146-8, http://hdl.handle.net/1765/10856

Verheijen, H.J.J., *Vendor-Buyer Coordination in Supply Chains,* Promotor: Prof.dr.ir. J.A.E.E. van Nunen, EPS-2010-194-LIS, ISBN: 90-5892-234-2, http://hdl.handle.net/1765/1

Verheul, I., *Is There a (Fe)male Approach? Understanding Gender Differences in Entrepreneurship*, Prof.dr. A.R. Thurik, EPS-2005-054-ORG, ISBN: 90-5892-080-1, http://hdl.handle.net/1765/2005

Verwijmeren, P., *Empirical Essays on Debt, Equity, and Convertible Securities*, Promotors: Prof.dr. A. de Jong & Prof.dr. M.J.C.M. Verbeek, EPS-2009-154-F&A, ISBN: 978-90-5892-187-1, http://hdl.handle.net/1765/14312

Vis, I.F.A., *Planning and Control Concepts for Material Handling Systems*, Promotors: Prof.dr. M.B.M. de Koster & Prof.dr.ir. R. Dekker, EPS-2002-014-LIS, ISBN: 90-5892-021-6, http://hdl.handle.net/1765/866

Vlaar, P.W.L., *Making Sense of Formalization in Interorganizational Relationships: Beyond Coordination and Control,* Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2006-075-STR, ISBN 90-5892-103-4, http://hdl.handle.net/1765/7326

Vliet, P. van, *Downside Risk and Empirical Asset Pricing*, Promotor: Prof.dr. G.T. Post, EPS-2004-049-F&A, ISBN: 90-5892-07-55, http://hdl.handle.net/1765/1819

Vlist, P. van der, *Synchronizing the Retail Supply Chain*, Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr. A.G. de Kok, EPS-2007-110-LIS, ISBN: 90-5892-142-0, http://hdl.handle.net/1765/10418

Vries-van Ketel E. de, *How Assortment Variety Affects Assortment Attractiveness: A Consumer Perspective*, Promotors: Prof.dr. G.H. van Bruggen & Prof.dr.ir. A. Smidts, EPS-2006-072-MKT, ISBN: 90-5892-101-8, http://hdl.handle.net/1765/7193

Vromans, M.J.C.M., *Reliability of Railway Systems*, Promotors: Prof.dr. L.G. Kroon, Prof.dr.ir. R. Dekker & Prof.dr.ir. J.A.E.E. van Nunen, EPS-2005-062-LIS, ISBN: 90-5892-089-5, http://hdl.handle.net/1765/6773

Vroomen, B.L.K., *The Effects of the Internet, Recommendation Quality and Decision Strategies on Consumer Choice*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2006-090-MKT, ISBN: 90-5892-122-0, http://hdl.handle.net/1765/8067

Waal, T. de, *Processing of Erroneous and Unsafe Data*, Promotor: Prof.dr.ir. R. Dekker, EPS-2003-024-LIS, ISBN: 90-5892-045-3, http://hdl.handle.net/1765/870

Waard, E.J. de, *Engaging Environmental Turbulence: Organizational Determinants for Repetitive Quick and Adequate Responses*, Promotors: Prof.dr. H.W. Volberda & Prof.dr. J. Soeters, EPS-2010-189-STR, ISBN: 978-90-5892-229-8, http://hdl.handle.net/1765/1

Wall, R.S., *Netscape: Cities and Global Corporate Networks,* Promotor: Prof.dr. G.A. van der Knaap, EPS-2009-169-ORG, ISBN: 978-90-5892-207-6, http://hdl.handle.net/1765/1

Watkins Fassler, K., *Macroeconomic Crisis and Firm Performance,* Promotors: Prof.dr. J. Spronk & Prof.dr. D.J. van Dijk, EPS-2007-103-F&A, ISBN: 90-5892-138-3, http://hdl.handle.net/1765/10065

Weerdt, N.P. van der, *Organizational Flexibility for Hypercompetitive Markets: Empirical Evidence of the Composition and Context Specificity of Dynamic Capabilities and Organization Design Parameters,* Promotor: Prof.dr. H.W. Volberda, EPS-2009-173-STR, ISBN: 978-90-5892-215-1, http://hdl.handle.net/1765/1

Wennekers, A.R.M., *Entrepreneurship at Country Level: Economic and Non-Economic Determinants,* Promotor: Prof.dr. A.R. Thurik, EPS-2006-81-ORG, ISBN: 90-5892-115-8, http://hdl.handle.net/1765/7982

Wielemaker, M.W., *Managing Initiatives: A Synthesis of the Conditioning and Knowledge-Creating View*, Promotors: Prof.dr. H.W. Volberda & Prof.dr. C.W.F. Baden-Fuller, EPS-2003-28-STR, ISBN: 90-5892-050-X, http://hdl.handle.net/1765/1042

Wijk, R.A.J.L. van, *Organizing Knowledge in Internal Networks: A Multilevel Study*, Promotor: Prof.dr.ir. F.A.J. van den Bosch, EPS-2003-021-STR, ISBN: 90-5892-039-9, http://hdl.handle.net/1765/347

Wolters, M.J.J., *The Business of Modularity and the Modularity of Business*, Promotors: Prof. mr. dr. P.H.M. Vervest & Prof.dr.ir. H.W.G.M. van Heck, EPS-2002-011-LIS, ISBN: 90-5892-020-8, http://hdl.handle.net/1765/920

Wubben, M.J.J., *Social Functions of Emotions in Social Dilemmas*, Promotors: Prof.dr. D. De Cremer & Prof.dr. E. van Dijk, EPS-2009-187-ORG, http://hdl.handle.net/1765/1

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/1765/1

Yang, J., *Towards the Restructuring and Co-ordination Mechanisms for the Architecture of Chinese Transport Logistics*, Promotor: Prof.dr. H.E. Harlambides, EPS-2009-157-LIS, ISBN: 978-90-5892-198-7, http://hdl.handle.net/1765/14527

Yu, M., *Enhancing Warehouse Perfromance by Efficient Order Picking*, Promotor: Prof.dr. M.B.M. de Koster, EPS-2008-139-LIS, ISBN: 978-90-5892-167-3, http://hdl.handle.net/1765/13691

Zhang, X., *Strategizing of Foreign Firms in China: An Institution-based Perspective*, Promotor: Prof.dr. B. Krug, EPS-2007-114-ORG, ISBN: 90-5892-147-5, http://hdl.handle.net/1765/10721

Zhang, X., *Scheduling with Time Lags,* Promotor: Prof.dr. S.L. van de Velde, EPS-2010-206-LIS, ISBN: 978-90-5892-244-1, http://hdl.handle.net/1765/1

Zhu, Z., *Essays on China's Tax System*, Promotors: Prof.dr. B. Krug & Prof.dr. G.W.J. Hendrikse, EPS-2007-112-ORG, ISBN: 90-5892-144-4, http://hdl.handle.net/1765/10502

Zwart, G.J. de, *Empirical Studies on Financial Markets: Private Equity, Corporate Bonds and Emerging Markets*, Promotors: Prof.dr. M.J.C.M. Verbeek & Prof.dr. D.J.C. van Dijk, EPS-2008-131-F&A, ISBN: 978-90-5892-163-5, http://hdl.handle.net/1765/12703

Erasmus Research Institute of Management - ERIM

RAILWAY CREW RESCHEDULING
NOVEL APPROACHES AND EXTENSIONS

Passenger railway operators meticulously plan how to use the rolling stock and the crew in order to operate the published timetable. However, unexpected events such as infrastructure malfunctions, or weather conditions disturb the operation every day. As a consequence, significant changes, such as cancellation of trains, to the timetable must be made. If these timetable changes make the planned rolling stock and crew schedule infeasible, one speaks of a disruption. It is very important that these schedules are fixed such that no additional cancellations of trains are necessary. Nowadays this rescheduling is still done manually by the dispatchers in the control centers.

In this thesis we use Operations Research techniques to develop solution approaches for crew rescheduling during disruptions. This enables us to solve the basic operational crew rescheduling problem in a short amount of computation time. Moreover, we studied an extension to the basic problem where the departure times of some trains may be delayed by some minutes. We show that this can lead to significantly better solutions for some real-life instances. Furthermore, we presented two new quasi robust optimization approaches that deal with the uncertainty in the length of the disruption. The computational study reveals that one of these approaches outperforms a naive approach in many cases. We believe that the methods developed in this thesis provided the foundation for a decision support system for railway crew rescheduling.

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 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)
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

Design & layout: B&T Ontwerp en advies (www.b-en-t.nl) Print: Haveka (www.haveka.nl)