TWAN DOLLEVOET

Delay Management and Dispatching in Railways

Delay Management and Dispatching in Railways

Delay Management and Dispatching in Railways

Het besturen van aansluitingen bij een spoorwegvervoerder

Thesis

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

Prof.dr. H.G. Schmidt

and in accordance with the decision of the Doctorate Board.

The public defense shall be held on

Thursday 10 January 2013 at 13:30 hrs

by

Twan Antonius Bernardus Dollevoet
born in Nuland, the Netherlands.

ERASMUS UNIVERSITEIT ROTTERDAM

Doctoral Committee

Promotor:	Prof.dr. A.P.M. Wagelmans
Other members:	Prof.dr. L.G. Kroon
	Prof.dr. A. Schöbel
	Dr. D. Pacciarelli
Copromotor:	Dr. D. Huisman

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

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

ERIM PhD Series in Research in Management, 272
ERIM reference number: EPS-2013-272-LIS
ISBN 978-90-5892-318-9

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

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

Acknowledgments

This thesis summarizes the research I have been working on during the past five years. Many people have contributed to this thesis and have made my time as a PhD student really enjoyable. I have been looking forward to this opportunity to thank these people.

First and foremost, I want to express my gratitude to my supervisors Dennis Huisman and Albert Wagelmans. Dennis introduced me to the topic of railway scheduling while I was writing my master's thesis at Netherlands Railways in 2007. I really enjoyed our cooperation, so I was very glad with the opportunity to do my PhD research under Dennis' supervision. Dennis, I thank you for your guidance and inspiration over the years. After every meeting we had, I continued to work on our research with renewed enthusiasm. I am also grateful for your clear road map towards the completion of this thesis and for making sure that I started writing our papers whenever that was necessary. Albert, many thanks for being my promotor. Whenever I needed any advice, you were always there to help me.

Part of the research described in this thesis has been carried out during two research visits at the Georg-August-Universität. During these periods in Göttingen, I was a guest of Anita Schöbel. I warmly thank Anita for her supervision and hospitality. I am very pleased that she agreed to be a member of my inner committee. Besides Anita, I want to thank her colleague Stephan Westphal and the PhD students Marie, Thorsten, Marc, Ruth, Robert, and Jonas for the good times in Göttingen.

I enjoyed the joint research with Anita and Marie, which has resulted in several publications and forms the basis for Chapters 2 and 4 of this thesis. I also want to thank Francesco Corman and Andrea D'Ariano for our cooperation, which is presented in Chapter 5.

I am also indebted to the other members of the inner committee, Dario Pacciarelli and Leo Kroon, for their valuable time to evaluate this thesis. In particular, I want to thank Leo for his detailed comments on an earlier draft and for pointing out an error in one of the chapters. I also thank Leo van Dongen and Rommert Dekker for being a member of my doctoral committee.

After my internship, NS provided me the opportunity to be involved in the further development of our crew scheduling algorithm LUCIA with a part-time position at the innovation department. It has been very inspiring to see how this algorithm has been implemented in the planning process and is now used in practice. The atmosphere at our department π has always been motivating. I want to thank all my colleagues at NS for their support and the nice years as an employee of NS. Particularly, I thank Erwin for allowing me to schedule my days at NS with much freedom. Without this flexibility, it would have been much harder to combine both jobs.

Most of the ideas described in this thesis have emerged from everlasting coffee breaks with my fellow PhD students in OR. I have always appreciated these breaks and our joint lunches and dinners. I want to thank Kristiaan, Remy, Ilse, Zahra, Willem, Judith, Mathijn, Lanah and my roommate Joris for always being there if I had a question or needed a two-minute or two-hour break. Willem, thank you for sharing your brilliant idea to switch to econometrics 7 years ago, for the cooperation during the last 11 years, and for your help and advice on all issues I have encountered during that period. I am happy that you agreed to be one of my paranymphs. Furthermore, I want to thank the fellow PhD students from RSM, in particular the other railway PhDs Luuk, Evelien, Paul, and Joris, and Evsen for our valuable discussions during several lunch seminars. Kar Yin, thank you for a template for this thesis and for answering all my questions about the ceremony and the schedule towards the completion of this thesis. Most importantly, I thank all fellow PhDs, in particular Anne, for our weekly drinks in the Smitse and for our adventures in Lunteren, Tromsø, Lisbon, Austin, Melbourne, and Santiago.

My final word of thanks goes to my friends and family. I truly believe that our everyday work benefits from having some distraction at times. I want to thank Jeroen and Lonneke, Willem and Eva, Rudi and Sylvia, Ruud and Roel, Joran and Lydia, Otto and Lonneke, Wilbert and Hanneke, Thijs and Willeke, Paul and Esther, Luc, Arthur and Marieke, Kyra, Annemieke, Janneke, Imke, and Marieke for making sure my life was not only about trains and scheduling algorithms. My deepest word of thanks is to my parents, sister, Ron, and my niece Fenne for their unconditional support and interest. Renske, I am pleased that you are also willing to stand by my side as one of my paranymphs.

Twan Dollevoet
's-Hertogenbosch, November 11, 2012

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Passenger railway transportation plays a crucial role for the mobility in the Netherlands. Especially around the main cities in the Randstad, which is the Western, most populated part of the Netherlands, many commuters make use of the railway system to get to their jobs and schools. During the peak hours, heavy congestion on the highways increases the travel times for cars, while the railway system provides a fast and sustainable mode of transportation.

The main performance indicator for most European railway operators is the punctuality. The punctuality measures the percentage of trains that arrive at a station with a delay below some threshold value. In most European countries, including the Netherlands, this threshold value is 5 minutes. The punctuality discards an important aspect: It considers only the delays of the trains; not those of the passengers. If a train is delayed by 4 minutes, it is possible that passengers cannot transfer to a connecting train. In that case, these passengers will get a delay that is much larger than the delay of their first train. To take the passenger delays into consideration, Netherlands Railways has recently introduced the *passenger punctuality* that measures the percentage of passengers who arrive within 5 minutes after their planned arrival time.

Delay management is a field in Operations Research that tries to minimize the nuisance for the passengers in case of small delays. By delaying connecting trains slightly, the transfer from one train to the next can be maintained. The delay for transferring passengers will be reduced by maintaining the connection, but the passengers who are already in the connecting train will be delayed. We illustrate this by the following real-world example.

One of the busiest long distance train series in the Netherlands is the 800 series. Trains from this series start in Maastricht in the South and continue via 's-Hertogenbosch,

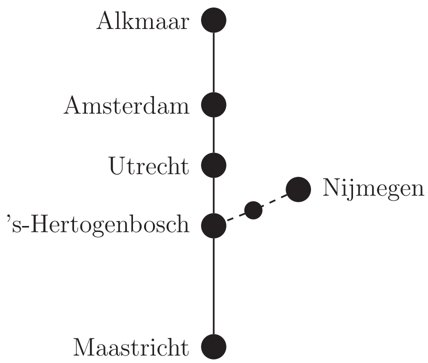

Figure 1.1: Part of the railway network in the Netherlands. The solid line represents the long distance train; the dashed line depicts a regional train. In ’s-Hertogenbosch, many commuters want to transfer from the regional train to the long distance train and vice versa.

Utrecht, and Amsterdam towards Alkmaar in the North-West of the country. Many commuters that use this series and depart from ’s-Hertogenbosch in Northern direction arrive at the station in ’s-Hertogenbosch by a regional train from Nijmegen. In Figure 1.1, the corresponding railway network is depicted. The long distance train departs five minutes after the regional train has arrived and departure and arrival take place on opposite sides of the same platform. If the regional train arrives on time, passengers can thus easily transfer to the long distance train. However, if the regional train has a small delay and the long distance train departs on time, this may not be possible. It would then be better for the transferring passengers if the long distance train would depart a few minutes later. If the long distance train departs on time, transferring passengers have to wait for the next train towards Utrecht, which departs 15 minutes later.
Things are worse in the opposite direction. Many commuters travel back from the North to ’s-Hertogenbosch and then transfer to the regional train. In this situation, the transfer time is again five minutes, but now the regional train departs from a different platform than where the long distance train arrives. Even if the delay of the long distance train is only two minutes, it is impossible to reach the regional train on time. If the regional train then departs on time, the transfer cannot be made by the passengers. As regional trains run less frequently, the passengers then have to wait for half an hour for the next regional train. This can be very frustrating. If the regional train would have waited for the passengers from the long distance line, their delay would be reduced by half an hour.

However, for passengers already in the train, waiting would have introduced delays that could be avoided.

The example illustrates the trade off between a large delay for few transferring passengers on the one hand and a small delay for many passengers in the connecting train on the other hand. The delay for the passengers who want to transfer will be reduced if the connecting train is delayed, but the passengers who are already in the train will get unnecessarily delayed. Furthermore, delaying a train can have consequences for other trains in the network, especially in highly utilized railway networks such as the one in the Netherlands. This suggests that the decision to delay a train is a very complex decision, where the consequences for all other trains should be considered carefully. Deciding which trains should wait for passengers from a delayed feeder train is the topic of this thesis.

In the first half of 2012, the percentage of dropped connections in the Netherlands equals over 8 percent. These missed connections are an important point of criticism from passengers. Decreasing the percentage of missed connections, and thereby lowering the average travel time, can improve customer satisfaction significantly.

The main focus in this thesis will be on the railway system in the Netherlands. Although our models and solution approaches are applicable to other railway operators as well, we have evaluated them on cases from the Dutch railway system only. Since the late nineties, railway activities in the Netherlands are assigned to an infrastructure manager (ProRail) on the one hand and several railway operators on the other hand. The infrastructure manager is responsible for building and maintaining the infrastructure and for the safe execution of the timetable. The railway operators are responsible for the product that is delivered to the passengers and manage the rolling stock and the crew. Netherlands Railways (Nederlandse Spoorwegen in Dutch, or NS) is the largest passenger railway operator. It transports about 1.2 million passengers with over 5000 scheduled trains on a normal weekday.

The remainder of this chapter is organized as follows. We first discuss the planning process and traffic control at Netherlands Railways and ProRail in Section 1.2. In Section 1.3, we will present some applications of Operations Research to scheduling problems arising at the railway system in the Netherlands. Finally, in Section 1.4, we will give an outline of this thesis.

1.2 Resource management at a railway operator

In this section, we first describe the planning process. Then, we will introduce delay management and dispatching, which controls the railway operations and resolves small deviations from the schedules. When the railway system faces larger disturbances, the timetable and crew and rolling stock schedules should be adjusted as well. This process is called disruption management and it is the last topic in this section.

1.2.1 Planning process

Here, we briefly discuss the planning process at Netherlands Railways. The resources that are discussed in this section are an important aspect of delay management as well. Four processes are distinguished: line planning, timetabling, rolling stock planning and crew planning. Most European railway operators apply a similar planning approach. For more details on the planning process at Netherlands Railways, we refer to Huisman et al. (2005).

The line plan describes the routes and frequencies of the train lines. It is determined at the strategic level, and generally remains unchanged for several decades. Netherlands Railways introduced a new line plan in 2007 that was completely developed from scratch. Since then, the line plan distinguishes between long-distance trains that only stop at larger stations and regional trains that stop at every station that is passed. Some of the long-distance trains run across the entire country. Regional train lines are generally much shorter and cover only 2 or 3 larger stations. The line planning process aims at giving as many passengers as possible a direct connection from their origin to their destination. However, it is impossible to offer a direct connection between each pair of stations. If many passengers want to travel between two stations that are not directly connected, the line plan also prescribes a connection from one line to another that makes sure that passengers can conveniently travel between these two stations. Political wishes and the available infrastructure also play an important role when the line plan is determined. The line plan of 2013 has again been developed completely from scratch.

When the line plan has been constructed, a timetable must be developed. The timetable gives the departure and arrival times of the trains at every station in the railway network. Moreover, it precisely describes the routes of the trains through a station and prescribes the platforms at which trains will stop. Netherlands Railways generates a new timetable every year. Doing so, it can adapt the timetable to new infrastructure becoming available, to new stations being opened and to changes in the passenger demand. As most railway companies, Netherlands Railways operates a cyclic, or periodic, timetable, which means that the timetable is repeated every hour. A clear advantage of cyclic timetables is that they are much easier to remember for the passengers. A drawback is their limited flexibility: As the timetable is equal every hour, fewer direct connections between pairs of stations can be offered. Furthermore, cyclic timetables are more expensive to operate. The timetable is constructed in two stages (see Kroon et al. (2008a)). The first stage is performed globally for the complete railway network and determines the arrival and departure times of the trains. Detailed information about the infrastructure within the stations is neglected in the first stage. However, the capacity of the tracks between the stations is taken into account. For example, when a long-distance train and a regional train use the same track, the long-distance train is not allowed to overtake the slower regional train. Besides this limited track capacity, the minimal travel times between stations should be respected. These travel times include some buffer times in order to deal with small delays.

In the second stage, which is performed locally at the larger stations, the trains are assigned to the platforms. When the arrival and departure times are determined, the routes of the trains through the stations should be determined. The objective for the local timetabling process is robustness. When more time is available between two trains using the same platform or track, a delayed train will not directly influence the next one.

After the timetable has been generated, rolling stock and crew schedules are constructed. For both processes, a similar division in a global stage and a local stage is observed. In the rolling stock planning, units of rolling stock are assigned to trains. In the global stage, first the type of rolling stock and the number of carriages for each train that runs at 8:00 A.M. is determined. The required capacity is highest at 8:00 A.M., so if an assignment can be found at that time, there will be a feasible assignment during the remainder of the day as well. The objective in this phase is to match the available capacity as closely as possible to the estimated demand. Given the allocation at 8:00 A.M., the train types are assigned for the remainder of the trains. After this assignment of train types, a route is determined for each unit of rolling stock. In some routes, a rolling stock unit is decoupled from a train and parked at a shunting yard for a while. The transportation between the shunting yard and the stations is considered in the local stage. We refer to Abbink et al. (2004) for more information on the process of planning the rolling stock at Netherlands Railways.

The final component in the planning process is crew planning. From the timetable and the rolling stock schedule, a set of *tasks* can be derived. A task is an indivisible amount of work that must be assigned as a whole to a crew member. An example of a task is driving a train from one station to the next. Driving a long-distance train from its begin to its end point consists of many tasks, which may be performed by different crew members. The crew planning process assigns all tasks to a crew member. The crew schedule has to satisfy many rules (see Abbink et al. (2005) and Abbink et al. (2011)). In a first stage, a set of anonymous duties is constructed, such that all tasks are present in at least one duty. Each duty starts and ends at one of the 29 crew bases. The objective in this global stage is to generate the minimum set of duties that cover all tasks. Of course, each duty should contain a break and respect the maximum duration. Furthermore, the set of duties should distribute the work fairly over the crew bases. This means that all crew bases should have the same amount of early and night duties, and roughly the same division between regional and long-distance trains, for example. In a second stage, the duties are assigned to actual drivers and guards. This process is called crew rostering.

A new timetable is constructed every year and revised roughly six times a year. After every revision, a rolling stock and crew planning is created for a generic day. For example, a set of duties for the crew is generated that can be operated on every Monday. Usually, these schedules are ready a few weeks before the day of operation. There could, however, be different requirements on specific days in the year. For example, more capacity is required when a large event takes place. Or due to maintenance work, it could be impossible to use parts of the network. In such cases, the timetable and the schedules for both rolling stock and the crew should be adapted, usually a few days in advance. Because less time is available on such short notice, finding a feasible schedule is more important than finding the most efficient one. Besides, an additional objective in the short-term phase is to maintain as much of the original plan as possible. However, the main problem characteristics remain the same as in the yearly stage. Therefore, we do not discuss the short-term planning process in detail, but refer to Huisman (2007).

1.2.2 Delay management and dispatching

One of the tasks of the infrastructure manager is the safe execution of the timetable. Currently, dispatchers consider only the operational aspects of the timetable. They focus mainly on larger delays that would cause the rolling stock or crew schedules to become infeasible. If this would happen, the delay would propagate to other trains in the network and induce new problems. On the contrary, smaller delays will be absorbed by the buffer times in the timetable and will not cause any problems for other trains.

As long as all trains run as planned, the signals and switches are adjusted automatically by the route setting system ARI (Automatische Rijweg-Instelling). When a train arrives at a junction on time, ARI will set the signals and the switches in the correct position so that the train can continue on the planned speed. ARI also implements some very basic dispatching rules to deal with trains that arrive at a junction with a small delay. Each switch is preceded by a measuring point, where trains driving towards the switch are detected. If a train arrives at the measuring point within a predefined time interval, the signals and switch are adjusted automatically. As long as the railway system faces only small delays, ARI will thus be able to set the routes for the trains automatically. Recall that the planned running times contain buffers to absorb delays. From an operational point of view, such small delays do not cause problems.

For the passengers however, a small delay can enlarge the travel time by half an hour or more, when a connection is missed. From the passengers' point of view, it can then be better to delay the connecting train also, in order to maintain the connection. Deciding whether connecting trains should wait for a delayed train is known as delay management (see Schöbel (2001)). The objective is to minimize the total or weighted delay of all passengers. When it is decided that a train should wait for delayed passengers, this train will probably arrive with a delay at the next station, where other passengers want to transfer. The decision to maintain a connection thus propagates through the entire railway network. Furthermore, if a train is delayed before it enters a busy track, it might keep up other trains that are scheduled on the same track. This illustrates that delay management is a complex problem, where a decision to delay a train can have consequences for many other trains as well.

Currently, the dispatchers apply a simple rule to determine whether a train should wait or not. For each connection, a maximal waiting time is specified. If a train arrives at a station with a delay, one first determines the waiting time needed to maintain the connection. If this waiting time is smaller than the maximal waiting time, the connecting train waits; otherwise it departs on time. Although this guideline is easy to implement, it discards two important aspects. First, although the guideline contains different waiting times for the peak and off-peak hours, it does not distinguish between the evening and the afternoon. During the afternoon, there are more trains running between two stations. This means that if a passenger misses a connection, the delay that this passenger will incur is smaller than during the evening. Furthermore, delaying a train in the afternoon will have much more consequences for other trains. These differences should be considered when the wait-depart decisions are made. Second, the guideline does not consider the number of passengers that transfer to the train. When there are many passengers that use the connection, one should wait longer for these transferring passengers. Note that the passenger flows during the morning peak hours are quite different from those during the afternoon peak hours.

1.2.3 Disruption management

Despite the efforts in the planning process, on the day of operations one often faces events that prevent the resource schedules from being executed as planned. We will distinguish between small and large disturbances. With small disturbances, only one or a few trains are involved. Larger disturbances involve more trains and are also known as disruptions. A blockage of a certain part of the infrastructure, broken rolling stock or an employee that does not show up at work are typical examples of a disruption. Formally, one speaks of a disruption in case the rolling stock or crew schedules become infeasible (see Jespersen-Groth et al. (2009)).

Recall that railway activities in the Netherlands are split between an infrastructure manager on the one hand and the railway operators on the other hand. The railway operators are responsible for the rolling stock, for the crew and for the passengers. The infrastructure manager is responsible for the execution of the timetable. It monitors the train movements of all railway operators and controls the signals and switches. We now first introduce the parties that are involved in the disruption management process. Then, we describe how a typical disruption is dealt with.

Since 2010, Netherlands Railways and ProRail have been centralizing the organizations involved in the disruption management process. In 2011, the *Operations Control Center Rail* (OCCR) was opened, in which both the *Network Traffic Control Center* (NTCC) and the *Network Operations Control Center* (NOCC) are located. The NTCC monitors the train traffic from the perspective of the infrastructure manager. The NOCC is its equivalent from the side of the operator and monitors the rolling stock, the crew and the passengers. The main reason to locate the NTCC and the NOCC together in the OCCR was to facilitate the communication between them.

Besides the NTCC, the infrastructure manager has 12 *Regional Traffic Control Centers* (RTCCs). Each RTCC is located at a larger station and is responsible for monitoring the train traffic in the area around that station, including some other, smaller stations. Furthermore, it controls the switches in that area and determines the train routes through the stations.

At the operator, there are five *Regional Operations Control Centers* (ROCCs). Tasks of an ROCC include the planning of shunting operations and monitoring and rescheduling the crew and rolling stock duties.

For common disruptions, such as the unavailability of a certain track, emergency scenarios are available that describe how to deal with the disruption. The scenario prescribes which trains should be canceled and indicates how the rolling stock will be rescheduled. If a disruption occurs, the NTCC determines which emergency scenarios are applicable.

In cooperation with the NOCC a specific scenario is then selected and the duration of the disruption is estimated. The NOCC proposes how to reschedule the rolling stock allocation. The implementation of the measures defined in the emergency scenario is delegated to the RTCCs and the ROCCs. The RTCCs remove the canceled train movements from the system and adjust the routes for trains that are still running. The ROCC adjusts the rolling stock duties, deals with local rolling stock issues and reschedules the crew duties. If different shunting operations are required, the ROCC makes sure that crew and rolling stock are available to perform the shunting task. Planning new shunting operations should be communicated with the corresponding RTCC, as infrastructure must be available to perform the shunting operation. Because of the canceled trains, many crew duties will become infeasible. Each ROCC is responsible for rescheduling the duties of crew members that are in the area of the ROCC during the disruption.

1.3 Applications of Operations Research

In the previous section, we described scheduling problems that arise at a railway operator. Many of these scheduling problems can be solved by using techniques from Operations Research. In this section, we will give an overview of such applications in the railway context.

1.3.1 Planning process

We will first discuss Operations Research applications that can be found in the planning process. Research on planning problems at Netherlands Railways started already in the 90s, and many of the solution methods have been implemented in practice.
Given the line plan, the timetabling problem determines first the arrival and departure times of all trains. If the timetable to be generated should be cyclic, this problem can be formulated as a *periodic event scheduling problem* (PESP, see Peeters (2003)). The PESP was first described by Serafini and Ukovich (1989). A special purpose algorithm to solve the PESP was developed by Schrijver and Steenbeek (1993). This algorithm, CADANS, applies constraint programming to find a feasible solution. If no feasible solution exists, CADANS reports which constraints are conflicting.
When a feasible set of departure and arrival times is found, the routes through the stations should be determined. These routes also determine the platform at which trains stop. STATIONS is a system that solves this train routing problem (see Zwaneveld et al. (2001)). To find feasible train routes, STATION first lists for each train the set of possi-

ble routes. Then, it selects one route for each train, such that no two selected routes are conflicting. Two routes are conflicting if they use the same part of the infrastructure at the same time.

CADANS and STATIONS form the major elements of DONS (*Designer of Network Schedules*), an integrated system that solves the complete timetabling problem (see Hooghiemstra et al. (1999)). DONS also provides an interface to SIMONE (*Simulation Model for Networks*), a simulation model to evaluate the quality of the timetable (see Middelkoop and Bouwman (2001)). Note that SIMONE can only be used to estimate performance indicators of the simulated timetable; it cannot be used to optimize any performance measures. The DONS system is applied yearly to generate a new timetable.

Kroon et al. (2008b) developed a *stochastic optimization model* (SOM) to optimize the punctuality of the timetable. SOM takes a given timetable as input, and rearranges the buffers in the driving times in such a way, that robustness against delays is maximized. To do so, a sample average approximation is applied. SOM was tested on a part of the Dutch railway network and is now being implemented as part of DONS. Other instruments to improve the reliability of the timetable are developed by Vromans (2005).

After the timetable has been generated, the rolling stock schedule is determined. The aim in this process is to assign a number of rolling stock units to each trip in the timetable. When determining the rolling stock schedule, three conflicting aspects must be balanced: (1) service to the passengers, (2) efficiency, and (3) robustness. Service to the passengers is high when every passenger is offered a seat. On the contrary, a schedule is efficient if it minimizes the rolling stock utilization, measured as the number of units that are used, and the total distance traveled by all units, measured in carriage kilometers. In order to reduce the number of carriage kilometers, rolling stock units can be coupled to trains shortly before the peak hours and decoupled afterwards. A drawback of (de)coupling units is that it decreases the infrastructure availability: Coupling and decoupling a train require infrastructure that would otherwise be available to regular train services. They also introduce one more scheduling problem: After being decoupled, the rolling stock units must be parked at shunting yards. The operational planning of these shunting operations are considered by Lentink (2006) and Freling et al. (2005).

As explained in the previous section, the required capacity peaks at 8:00 A.M. in the morning. Therefore, one first allocates the train units during that period. Abbink et al. (2004) describe a model that determines the type and number of train units for each train that is driving at 8:00 A.M. The solution of this model is used to divide the available train units over the different lines. Given this capacity distribution, a rolling stock circulation is developed for each train line. More specifically: For each train on a line a rolling stock *composition* is selected. To find this selection, one first determines the set of feasible compositions for each trip. The length of a composition is limited by the length of the shortest platform during the trip, hence the set of feasible compositions is bounded. Then, one determines the *composition changes* that are allowed at the stations. For example, it may be possible to remove a train unit at the front or at the end of the train, but not at both. Rolling stock scheduling can now be formulated in a graph, where a node corresponds to the assignment of a composition to a trip, and the arcs represent a composition change at a station. A rolling stock circulation is then equivalent to a flow through this network. This composition model is described in Maróti (2006) and further developed in Fioole et al. (2006) and Nielsen (2011). A prototype was first implemented in ROSA (*Rolling Stock Allocation*) and a full system is now known as TAM (*Tool voor de Aanpassing van de Materieelinzet*).

In the crew scheduling problem, a driver and a number of guards must be assigned to each task in the timetable. The crew scheduling problem is modeled as a set covering problem by Caprara et al. (2002). In this model, the columns correspond to the duties of the crew and the rows are the tasks that must be covered. To solve such a set covering model, one commonly applies a column generation approach to deal with the enormous amount of possible duties. The restricted master problem is formulated as the Lagrangian relaxation of the set covering problem and solved by the subgradient method. An advantage of the Lagrangian relaxation is that it gives a natural procedure to obtain feasible solutions heuristically. The pricing problem corresponds to a shortest-path problem that can be solved by dynamic programming.

The set covering model is adapted to the situation in the Netherlands and implemented in the TURNI system (see Abbink et al. (2005)). TURNI has been applied at Netherlands Railways until 2008. It generates a crew schedule for each individual day of the week. However, some of the constraints apply to all duties during the complete week. To find solutions that satisfy these constraints over the week, a decomposition algorithm has been developed in Abbink et al. (2007).

TURNI played a crucial role during the strikes in 2001, when the crew of Netherlands Railways did not agree on the new policy of the board. In this new policy, the duties for the drivers would contain several trips from the base up and down to a nearby station. Such duties were assumed to be much easier to reschedule during disruptions. However, the crew members feared such duties would decrease the variation in their work, and termed the new plan “Circling the Church”. After heavy negotiations between the board and the unions, a new set of rules was introduced, that are known as “Sharing Sweet and Sour”. These rules were so complex that a solution obeying to these rules could not be found manually. Using TURNI, a crew plan could be generated that respected the new requirements, and the strikes ended. This process is described in much more detail in Abbink et al. (2005).

Since 2008, Netherlands Railways applies the crew scheduling algorithm LUCIA, which is implemented as part of the CREWS system that is used by the planning department. LUCIA is based on the same mathematical model as TURNI, and allows for parallel execution of the pricing problem. Furthermore, it can solve instances that contain all tasks in a complete week at once. Recall that some of the rules apply to all duties in a week. By solving the problem in an integrated way, solutions can be found that are about 1% better than solving the problem for the individual days (see Abbink et al. (2011)).

In 2007, Netherlands Railways introduced a new line plan and timetable that were built completely from scratch. A new timetable was needed to accommodate the increasing number of passengers and because the punctuality had to be improved. The reason to introduce this new timetable in 2007 were several infrastructure projects that would become available by the end of 2006. The most important projects are quadrupling of the tracks between Utrecht and Amsterdam, the Betuwe line for freight trains from the port of Rotterdam to Germany and the high speed line from Amsterdam to Belgium. Although these projects turned out to finish with some delay, the board of Netherlands Railways decided to introduce the new timetable nonetheless.

The timetable of 2007 and the rolling stock and crew schedules were all generated with the models described above. It was the first time that all three resources were planned with algorithms based on Operations Research techniques. For this application of Operations Research, Netherlands Railways was awarded the INFORMS Franz Edelman Award in April 2008. Kroon et al. (2009) describe the planning process and the challenges that were faced during the implementation of the scheduling systems mentioned above.

1.3.2 Dispatching

Research on dispatching in the Netherlands has mainly focused on the train scheduling problem. Given the current position of all trains in the area, the aim is to determine the order of trains at switches and junctions, such that the maximal delay is minimized. As the infrastructure in the stations is very dense, a microscopic model can be used to capture the required level of detail. In such a microscopic model, the infrastructure is represented by a set of individual block sections, that cannot be entered by two trains simultaneously.

In close cooperation with ProRail, D'Ariano (2008) applied the concept of alternative graphs (see Mascis and Pacciarelli (2002)) to model the train scheduling problem. In this graph, a node represents the entrance of a train in a specific block section. Arcs between these nodes represent precedence constraints between these events. For example, driving from one block section to the next and stopping at a platform are arcs in the network. Besides these operational constraints, an order has to be determined for two trains that compete for a common block section in the network. The two possible choices are represented by a pair of alternative arcs, one of which must be selected. A complete ordering is given by a choice for one alternative arc from each pair. Such an ordering is feasible if the resulting network does not contain cycles. In that case, the entrance times can be found by dynamic programming.

A branch-and-bound algorithm to optimize the ordering decisions is developed by D'Ariano et al. (2007). Here, the routes of the trains through the stations cannot be changed. In Corman (2010), several extensions to this model and solution methods are developed. In D'Ariano et al. (2008), an integrated model is presented that optimizes both the train routes and the ordering decisions simultaneously. A tabu search algorithm to solve this model is studied by Corman et al. (2010a). All methods described so far consider only an isolated area around a station. However, delays arising at one station will propagate to the next, if the buffer in the running time is insufficient to absorb the delay. For that reason, the ordering decisions in the stations should be coordinated amongst each other. Instances that describe the complete network in the Netherlands would become too large, therefore Corman et al. (2012) describe a decomposition technique that allows to solve smaller instance and synchronize the decisions in each part.

Delay management has attracted minor attention in the train scheduling literature. In Corman et al. (2010b), a bi-objective model is described that minimizes the maximal delay of the trains on the one hand, and the number of missed connections on the other.

1.3.3 Disruption management

We now describe applications of Operations Research in disruption management. The literature on Operations Research methods in this area has focused on rolling stock and crew rescheduling. It is assumed that an emergency scenario is available that gives the adjusted timetable. The first task to perform in this situation is the rolling stock rescheduling. This process is described in full detail by Nielsen (2011). The approach to solve this problem is very similar to that in the planning phase. When a disruption occurs, a new composition must be assigned to the trains that are affected by the disruption. Besides, after the disruption, the rolling stock schedules should return to the original plan as soon as possible. An additional objective is to find a circulation that is as similar as possible to the original rolling stock schedule. Another important aspect in rolling stock scheduling is that by the end of the day, the number of rolling stock units at each shunting yard should match the amounts that are required on the next day. If one requires these balances to be restored by the end of the disruption, many shunting operations are needed. These shunting operations require much capacity at the stations, and therefore negatively influence robustness. To obtain solutions that are more robust, Nielsen et al. (2012) apply a rolling horizon approach. During the first part of the disruption, the off-balances are not penalized in the objective function. If the rolling horizon reaches the end of the day, rolling stock mismatches are penalized more heavily. Doing so, more time is available to restore the inventories at the end of the day, and the number of shunting operations can be reduced. Furthermore, the rolling horizon approach reduces the need for an accurate estimate for the duration of the disruption. In practice, it is usually not known when the infrastructure will be fully available again. With the rolling horizon approach, a schedule is computed with the most recent estimate of the duration of the disruption. When a new estimate becomes available, this new information is used in the rescheduling run for the next planning horizon.

If a certain track is blocked for several hours, Netherlands Railways advises the passengers to travel via another route, if possible. On these alternative routes, the planned seat capacity will not be sufficient. Kroon et al. (2010) describe an iterative algorithm that deals with this situation. In each iteration, a new rolling stock schedule is determined first. Then, the capacity shortage in each train is determined by simulating the passenger behavior. These shortages are fed back to the rolling stock rescheduling algorithm in the next iteration. The method converges to a solution that balances the rescheduling cost and the capacity shortages.

The crew rescheduling problem during disruptions is studied by Potthoff (2010). The solution methodology for the rescheduling problem closely resembles the methodology in crew planning. However, instead of generating complete duties for each base in the pricing problems, a set of replacement duties is determined for each duty that has to be rescheduled. Such a replacement duty contains the work in the duty that has already been performed, and extends the duty with tasks that can be performed in the future. The aim is now to select a feasible replacement duty for each original duty, such that all tasks from the emergency timetable are covered. A crucial aspect in the disruption process is that only limited time is available. Therefore, one only determines a replacement duty for the crew members whose duty cannot be executed as a consequence of the disruption, plus some duties that contain tasks in the geographical area around the disruption. This procedure is called dynamic duty selection and was developed by Potthoff et al. (2010). In a computational study, it is shown that this algorithm produces solutions of higher quality than the manual solutions found by the dispatchers. In most of the cases, a crew member was found for all tasks from the emergency timetable. However, few cases were reported in which some tasks could not be covered. In those cases, it could be useful to delay a train slightly. This approach, in which timetabling and crew rescheduling are integrated to some extent, allows for more tasks to be covered. For more information on crew rescheduling with retiming, we refer to Veelenturf et al. (2011).

Similar as for rolling stock rescheduling, the uncertainty about the duration of the disruption should be taken into account in the crew rescheduling process. This aspect is considered in the quasi-robust crew rescheduling problem, that is introduced by Veelenturf et al. (2012). In this model, both an optimistic, short estimate and a pessimistic, longer estimate for the duration of the disruption is given. To deal with the uncertainty, first a solution is determined with the optimistic estimate. If the duration turns out to be longer, the duties are rescheduled again. As a consequence, the model distinguishes between the first-stage costs with the optimistic estimate, and the second-stage costs that are only incurred if the duration is longer and the duties should be rescheduled a second time. In the solution approach, a replacement duty is called quasi-robust if it is feasible for the shortest duration and can easily be adjusted in case the duration is longer. Solutions with many quasi-robust duties will have a higher first-stage cost. However, in case the pessimistic estimate was correct, the second-stage costs will be lower. By varying the number of quasi-robust duties in a solution, a trade-off can be made between the first-stage costs and the second-stage costs.

1.4 Outline of this thesis

The topics of this thesis are delay management and dispatching in railways. Recall that delay management deals with the transfers between trains. If a delayed feeder train arrives at a station where passengers have a short connection to another train, this connecting train might have departed before the passengers can enter it. For the transferring passenger, it would then have been better if the connecting train had waited, but this would have introduced new delays for the passengers that are already in the connecting train. If the connecting train departs on time, the transferring passengers have to wait for the next train to their destination. The aim of delay management is to decide on the wait-depart decisions in such a way, that the overall delay for the passengers is minimized.

In this thesis we consider the off-line delay management problem: We assume that all delays in the system are known for a given period of time. Given these delays, the planned timetable, and the travel plans of all passengers, the aim is to determine for each connection whether it should be maintained or not. If it has been determined whether trains should wait for a delayed feeder train or not, it is easy to compute the *disposition timetable*, which gives the rescheduled times when all departures and arrivals take place. A first integer programming formulation for the delay management problem was developed by Schöbel (2007). In this first model, two simplifying assumptions are made.

1. The delay for passengers who miss a connection equals one cycle time.

2. The infrastructure capacity can be neglected.

The impact of the limited capacity of the tracks is studied by Schachtebeck (2010), who introduced the *capacitated* delay management problem. A model is presented that includes headway constraints for trains that utilize the same piece of the infrastructure. These headway constraint represent the two possible orderings of trains on the same track, one of which should be selected.
In Chapters 2 and 3 of this thesis, we investigate the validity of the first assumption. Instead of assuming a delay of one cycle time for passengers who miss a connection, we measure the delay for these passengers exactly. First, in Chapter 2, we introduce *Delay Management with Rerouting of Passengers*, that simultaneously optimizes the wait-depart decisions and the passenger routes. This approach, where routing decisions are integrated in network problems, is studied by Schmidt (2012) in a wide range of applications. To model the passenger flows in the network, we introduce the notion of origin-destination pairs (OD pairs), that represent a group of passengers with the same origin and destination who enter the railway system at a specified time. We assume that passengers prefer the fastest route to their destination and integrate the routing decisions in the integer programming formulation. As such, we are able to determine the delays for the passengers exactly and can thus minimize their true delays. For medium-sized real-world instances from Netherlands Railways, the integer programs can be solved by standard optimization software. We show in a computational study that the delay is reduced by up to 8% when the routing decisions are incorporated in the models, with respect to a model without passenger rerouting. However, for large-scale instances, solving the integer program takes too much time for practical applications. Therefore, in Chapter 3, we propose several heuristics for the delay management problem with rerouting decisions. The first heuristic is based on the model without passenger rerouting. Instead of assuming that passengers who miss a connection have a delay of one cycle time, we view the extra delay after missing a connecion as a parameter and search for the best value of this parameter by

enumeration. In the second heuristic, we replace this global parameter by an OD pair-specific penalty that is updated iteratively. We first assume that there is no extra delay for passengers who miss a connection and determine an optimal disposition timetable. Then, we determine for each OD pair, the fastest route to their destination. For passengers who miss a connection, we compute the additional delay with respect to their planned arrival time and use this extra delay as the value for the parameter in the next iteration. By repeating this procedure, a good estimate for the additional delay of each OD pair is obtained that can be used in the model without passenger rerouting. Finally, we apply a set of simple dispatching strategies as a third heuristic. These simple rules are currently used by the dispatchers. In a computational study, we show that the iterative heuristic performs very well. It is able to compute solutions that are at most 1% worse than the optimal solution within seconds. It thus combines the fast solution method for the model without rerouting and the improved quality of the model that includes the passenger routes. However, the simple dispatching rules perform very badly. The solutions found with these heuristics are at most 5% better than the solution in which no train waits at all.

In Chapters 4 and 5, we consider the second assumption of the delay management model from Schöbel (2007). In Chapter 4, we extend the model of Schachtebeck (2010) by incorporating the limited capacity of the stations. To model the station capacity, we view a station as a set of platform tracks. For trains that are scheduled on the same platform track, headway activities are introduced that make sure that a train can only enter the station after the preceding train has left the platform. We also allow to reschedule the platform assignment within the station dynamically. We thus allow a train to be assigned to another platform track. We evaluate the effect of a dynamic platform assignment on instances from Netherlands Railways and show that it reduces the delay of the passengers significantly. For larger instances, the integer programs for delay management with a dynamic platform assignment become very large. We therefore propose a heuristic that iteratively solves first a delay management problem with a static platform assignment and then optimizes the platform assignment in each station separately.

Inspecting the solutions with a dynamic platform assignment, we observe that the delay reduction is accompanied by many changes in the platform assignment. These platform track changes are annoying for the passengers and put pressure on the dispatching organization. Therefore, we propose the bi-objective delay management problem with capacities of stations that minimizes the passengers' delay on the one hand and the number of platform track changes on the other. We show that a big portion of the delay reduction can be obtained by allowing only few platform changes.

In Chapter 5, we aim at closing the gap that currently exists between the train scheduling and the delay management literature. In train scheduling, a microscopic model is commonly applied to capture all details of the railway infrastructure. These models distinguish each individual signal and junction in the network and have a precision of seconds. The aim of train scheduling algorithms is to schedule all train movements within a small, congested area in the network. In contrast, delay management models take a macroscopic view and generally consider larger parts of the railway network. We propose an iterative algorithm that first optimizes the macroscopic delay management problem. Given the connections to be maintained, we then apply the microscopic train scheduling algorithm to determine the exact arrival and departure times in the stations. The objective in the train scheduling model is to minimize the maximal train delay with respect to the disposition timetable from the delay management model. In the solution of the train scheduling problem, some trains will arrive or depart later than in the disposition timetable. These deviations are used as input for the delay management problem in the next iteration. We show that the iterative heuristic converges for instances with small source delays. When larger delays are considered, the behavior of the iterative approach is unstable.

Chapters 2-5 are based on papers that are either published in or submitted to scientific journals. As such, they come with their own introduction, literature review and conclusions. The references to these papers are given below.

Chapter 2 T. Dollevoet, D. Huisman, M. Schmidt, and A. Schöbel, "Delay Management with Rerouting of Pasengers". *Transportation Science*, 46(1):74-89 (2012b)

Chapter 3 T. Dollevoet and D. Huisman "Fast Heuristics for delay management with passenger rerouting". *Submitted to Public Transport* (2011)

Chapter 4 T. Dollevoet, D. Huisman, M. Schmidt, and A. Schöbel, "Delay Management including Capacities of Stations". *Submitted to Transportation Science* (2012c)

Chapter 5 T. Dollevoet, F. Corman, A. D'Ariano, and D. Huisman, "An iterative optimization framework for delay management and train scheduling". *Submitted to Flexible Services and Manufacturing* (2012a)

In Chapter 6 we summarize the main findings of each chapter and draw some general conclusions on delay management. Furthermore, we will present some possible directions for further research.

Chapter 2

Delay Management with Rerouting of Passengers

2.1 Introduction and Motivation

Passenger railway transport plays an important role in the European mobility. Especially, during peak hours or for distances between 20 and 800 kilometers, passengers often choose to travel by train. To ensure a high frequency and an easy-to-remember timetable, most European railway companies have opted for a cyclic timetable (see Liebchen (2008) or Kroon et al. (2009) for two recent publications on the subject). In such a timetable, each line has to be operated in a cyclic, or periodic, pattern: the trains run, for example, every 30, 60 or 120 minutes. A weak point in such a system is that passengers often have to change trains, because it is impossible to give a direct connection between all origin-destination pairs. To minimize the inconvenience of changing from train A to train B, the timetable is often constructed in such a way that train B departs shortly after train A arrives preferably with a cross-platform connection, i.e. both trains stop at two adjacent tracks of the same platform. However, if train A has a delay during the operations, the question is whether train B should wait or depart. Such decisions are called *delay management* (DM).

DM deals with (small) source delays of a railway system as they occur in the daily operations. In case of such delays, the scheduled timetable is not feasible anymore and has to be updated to a *disposition timetable*. Because delays are often transferred if a connecting train waits for a delayed feeder train, such connections are often not maintained in case of delays.

There exist various models and solution approaches for DM. The main question, which has been treated in the literature so far, is to decide which trains should wait for delayed feeder trains and which trains better depart on time (*wait-depart decisions*). A first integer programming formulation for this problem has been given in Schöbel (2001) and has been further developed in De Giovanni et al. (2008) and Schöbel (2007); see also Schöbel (2006) for an overview about various models. The complexity of the problem has been investigated in Gatto et al. (2005). The online version of the problem has been studied in Gatto et al. (2007), Gatto (2007) and Berger et al. (2011). Further publications about DM include a model in the context of max-plus algebra (de Vries et al., 1998; Goverde, 1998), a formulation as a discrete time-cost trade-off problem (Ginkel and Schöbel, 2007) and simulation approaches (Suhl and Mellouli, 1999; Suhl et al., 2001). Recently, also the limited capacity of the track system has been taken into account, see Schöbel (2009) for modeling issues and Schachtebeck and Schöbel (2010) or Schachtebeck (2010) for heuristic approaches solving capacitated DM problems.

What has been neglected in most studies so far is the aspect of rerouting. In most of the available models, it is assumed that passengers take exactly the lines they planned, i.e. if they miss a connection, they have to wait a complete period of time (the cycle time) until the same connection takes place again. This assumption is usually not valid in practice. Often, there is an earlier connection using another line or even changing the path of the trip that a passenger can take. In our work, we show how such rerouting can be incorporated within the DM process.

DM with rerouting has found little attention in the literature so far. Berger et al. (2011) investigate an online DM model, where passengers have a fixed route through the infrastructure network but are allowed to adapt the choices of the trains they take according to the delays. Gatto et al. (2005) show strong NP-hardness for a DM problem, where passengers are allowed to choose their route according to the current delays.

A real-world example of a situation where rerouting passengers in case of delays is beneficial, is given next. Consider the black nodes in the railway network in Figure 2.1. An intercity service runs from Zwolle to Utrecht via Amersfoort. There are also intercities from Utrecht to Amsterdam, and from Amersfoort to Amsterdam, and a regional train from Amersfoort to Amsterdam. The intercities only stop at larger stations, while the regional train stops at all stations on its route. A large number of passengers want to travel from Zwolle to Amsterdam, and they thus have a transfer at Amersfoort. In the current timetable, the transfer time for those passengers is rather small, hence these passengers will miss the connecting intercity to Amsterdam if their train is slightly delayed.

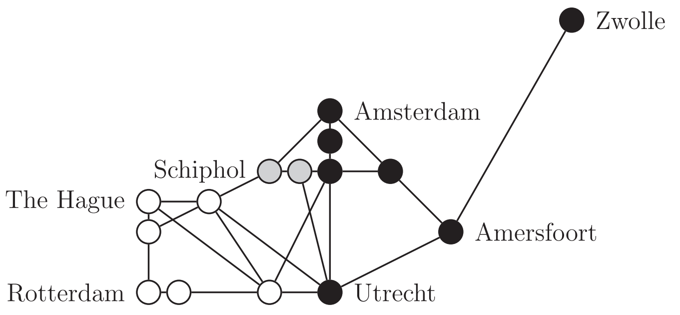

Figure 2.1: Part of the railway network in the Netherlands. The western part of the country is depicted. A circle in the picture indicates a station where long distance trains stop. The stations where only regional trains stop are not depicted. A line indicates that there is a direct connection between two stations. For each line, there are two or four long-distance trains and two regional trains per hour.

- If the possibility of rerouting the passengers is not taken into account, the intercity from Amersfoort to Amsterdam will be forced to wait to avoid that these passengers miss their connection and have to wait for one hour for the next intercity.
- However, if we allow the passengers to adapt their route to the current delay situation, it turns out that missing the intercity from Amersfoort to Amsterdam is not as bad as thought. There are two other possibilities: The passengers can stay in the delayed train and transfer in Utrecht instead, or they can use the regional train from Amersfoort to Amsterdam. Both alternatives lead to delays which are significantly less than one hour and finally to the decision that the intercity from Amersfoort to Amsterdam better departs on time.

This small example shows that the delay of passengers that miss a connection is often much smaller than the cycle time of the timetable. To find optimal wait-depart decisions, rerouting passengers should therefore be taken into account.
In this thesis, DM is treated as an offline problem, signifying that all delays are known before the optimization process starts. Offline DM can be used for short-term adaption of timetables whenever delays can be anticipated, e.g. in the case of construction works. Furthermore, given the source delays that are currently in the system, many secondary delays can be predicted. Offline DM can then propose how to react to these secondary delays. Moreover, regarding computational complexity, our offline DM is contained in NP, which makes it tractable by an integer programming approach, while PSPACE-hardness

was shown for an online DM allowing free choices of trains in Berger et al. (2011).
In this chapter, we will investigate how such a re-routing of passengers can be incorporated into the DM model. We denote the resulting model by *delay management with rerouting decisions (DMwRR)*. The contributions of this chapter are as follows. Firstly, we have developed a new model and integer programming formulation for DMwRR. Secondly, we investigate the complexity of DMwRR in several special cases. We prove that DMwRR is polynomially solvable for only one origin-destination pair, whereas in the slightly more general case of having all passengers starting at the same origin, the problem results to be strongly NP-hard. Furthermore, we prove strong NP-hardness for a DM problem with simplified delay costs. And finally, our third contribution is that we show that DMwRR can be solved for large real-world instances and performs significantly better than the existing models, where re-routing of the passengers is only taken into account after the wait-depart decisions have been made.
The remainder of this chapter is structured as follows. In Section 2.2, we show how the DM model can be modified to include rerouting of passengers. An integer program using event-activity networks is formulated in Section 2.3. In Section 2.4, we present a polynomially solvable special case of the problem. We show that a slight generalization of this case is already NP-hard. Furthermore, we discuss another simplified variant in which we assume fixed delay costs for each maintained changing activity. In Section 2.5, we report the results of several experiments based on real-world data of Netherlands Railways, the largest passenger operator on the Dutch railway network. Finally, we conclude the chapter by mentioning ideas for further research.

2.2 Model

Given some source delays from outside the system, the delay management problem is to decide which trains should wait for delayed feeder trains and which trains should depart on time. The goal is to find a solution, which is best for the passengers. In our work, we want to minimize the sum of all delays over all origin-destination pairs (OD pairs) assuming that all passengers take shortest paths.
In classical DM, passenger routes are determined before the optimization step given the timetable as it was planned to be operated. In the optimization step, passengers are assumed to stick to their predefined routes. If a connection on such a route is dropped, the passengers are assumed to wait until the connection is available again in the next period, thus the delay of one cycle time is added to their travel time. However, the actual delay of the passengers might not equal one cycle time. On one hand, there might be a

faster way for that passenger to arrive at his destination. The passenger will then not wait for the same connection in the next period, but will choose the earlier alternative. The delay will then be less than one cycle time. On the other hand, if there are no earlier alternatives, it is possible that the train in the next period is delayed as well. The delay for the passenger is then larger than one cycle time. To compute the delay for the passengers correctly, their routes have to be determined during the optimization phase. By computing a route for all passengers explicitly, the actual delay for the passengers can be determined correctly. Another advantage of rerouting passengers can be seen at the end of the day. If a passenger misses a connection to the last train of the day, he will have to wait for the first train on the next day. The classical model assumes that the delay is then equal to one cycle time, which is clearly an underestimation. When passengers are rerouted in the optimization phase, an explicit route is found for each passenger. It is therefore certain that all passengers will arrive at their destination.

To include the routing in the model, instead of fixing the route beforehand, we introduce OD pairs. We assume that the number of passengers who want to travel from a given origin to a destination at a certain time is known. For example, 200 passengers want to travel from Zwolle to Amsterdam at 8:00 A.M. We denote such an OD pair by $p = \{u, v, s_{uv}\}$, where u is the origin, v is the destination, and s_{uv} is the planned starting time of the trip. $\mathcal{P}$ denotes the set of all such OD pairs. We use w_p for the number of passengers associated with an OD pair $p \in \mathcal{P}$.

To model the DM problem with rerouting, we will make use of event-activity networks, first introduced by Nachtigall (1998) for timetabling problems and used for the classical DM problems by Schöbel (2006). The event-activity network $\mathcal{N} = (\mathcal{E}, \mathcal{A})$ is a directed graph, where $\mathcal{E}$ denotes the set of events and the set $\mathcal{A}$ consists of the activities. The departure or the arrival of a train g at a station v, denoted by $(g{-}v{-}Dep)$ or $(g{-}v{-}Arr)$, respectively, are the most important events in the network. For each event $e \in \mathcal{E}_{\text{dep}} \cup \mathcal{E}_{\text{arr}}$, we denote the planned departure or arrival time by π_e and the source delay by d_e. To incorporate the routes of the passengers, we introduce for every OD pair $p = \{u, v, s_{uv}\} \in \mathcal{P}$, an origin event $Org(p)$ and a destination event $Dest(p)$. Note that besides the origin and destination, the OD pairs also contain the time at which passengers want to start their journeys. In summary, the set of events in the network, denoted by $\mathcal{E}$, consists of the departure events of the trains, the arrival events of the trains, and the origin and destination events for the passengers for a given OD pair:

$$\mathcal{E} = \mathcal{E}_{\text{dep}} \cup \mathcal{E}_{\text{arr}} \cup \mathcal{E}_{\text{org}} \cup \mathcal{E}_{\text{dest}}.$$

The activities are the arcs in the directed graph $\mathcal{N}$. There are driving activities, waiting activities and changing activities. The driving and waiting activities represent driving from one station to the next and waiting at a station to let the passengers get on and off the train. The changing activities are used by the passengers. They represent the possibility for passengers to transfer from a train that arrives at a certain station to a train that departs at the same station some time later. Note that we only consider transfers that are present in the planned timetable. We do not include transfers that can only be maintained if the feeder train arrives with a delay. It should be noted that the driving and waiting activities impose operational restrictions on the vehicles, whereas a changing activity does not imply that a train has to wait in case of a delay of another train. In fact, the decision "wait or depart on time" is the main decision we want to take during the optimization process. We denote L_a for the minimal time that is needed to perform activity $a \in \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{change}}$.

To take the rerouting of passengers into account, we additionally introduce origin and destination activities. Let an origin event $e = Org(p) \in \mathcal{E}_{\text{org}}$ be given, where $p = \{u, v, s_{uv}\}$ represents the passengers who want to travel from station u to station v at time s_{uv}. This event e is connected to all departure events that depart from u not earlier than time s_{uv}. It remains to connect the arrival events to the destination events. Consider therefore a destination event $Dest(p) \in \mathcal{E}_{\text{dest}}$, where again $p = \{u, v, s_{uv}\}$. Let SP_p denote the earliest arrival time of the passengers if there are no delays. SP_p can be calculated using a shortest-path algorithm in the event-activity network with all connections. As the overall delay at an event cannot exceed $\max_{e' \in \mathcal{E}} d_{e'}$, a connection $a = (e_1, e_2) \in \mathcal{A}_{\text{change}}$ that satisfies $\pi_{e_2} - \pi_{e_1} \geq L_a + \max_{e' \in \mathcal{E}} d_{e'}$ will always be maintained. By solving a shortest-path problem in the event-activity network with only these safe connections, we can find a path in the event-activity network on which all connections will certainly be maintained. Denoting A_p for the planned arrival time on this path, it holds that $A_p + \max_{e' \in \mathcal{E}} d_{e'}$ is an upper bound for the arrival time for OD pair p. An arrival event e should therefore be connected to $Dest(p)$ if e is an arrival event at station v and if the planned time π_e satisfies $\pi_e \in [SP_p, A_p + \max_{e' \in \mathcal{E}} d_{e'}]$. This concludes the description of the activities in the event- activity network. Summarizing

$$\mathcal{A} = \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{change}} \cup \mathcal{A}_{\text{org}} \cup \mathcal{A}_{\text{dest}}.$$

An example of an event-activity network is given in Figure 2.2. This event-activity network corresponds to the example that we described in Section 2.1. The oval nodes represent the origin and destination events that are introduced to model the behavior of

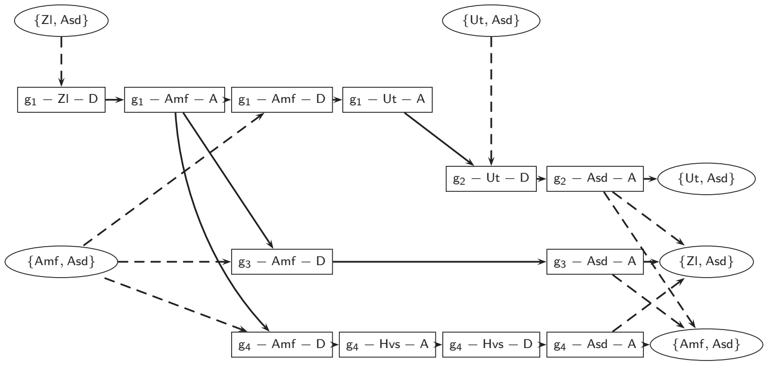

Figure 2.2: The squared nodes are the departure and arrival events where "D" stands for departure and "A" stands for arrival. The origin and destination events are represented by ovals. As we only consider one possible departure time for each OD pair, we did not include the starting time in the origin and destination events. The dashed arcs are the origin and destination arcs, that are introduced to be able to state the shortest-path problem for the passengers. The solid lines represent driving, waiting and changing activities.

passengers when delays occur. The dashed arcs depicting the origin and destination activities are needed to take the routing of passengers into account.

Recall that π_e denotes the planned time for each event $e \in \mathcal{E}_{\text{arr}} \cup \mathcal{E}_{\text{dep}}$, i.e. π corresponds to the timetable as it is planned to be operated. For an origin event $e = Org(p) \in \mathcal{E}_{\text{org}}$ with $p = \{u, v, s_{uv}\}$, we set $\pi_e = s_{uv}$ (which can be interpreted as the time at which a passenger of OD pair p arrives at his or her departure station). For a destination event $e = Dest(p)$, the planned arrival time π_e depends on the path chosen by the passengers in p. Assuming rational behavior of the passengers, we set $\pi_e = SP_p$, that is the earliest possible arrival time for OD pair p.

Given a set of source delays d_e associated to some events $e \in \mathcal{E}_{\text{arr}} \cup \mathcal{E}_{\text{dep}}$, the problem is to decide which trains should wait for passengers to arrive from delayed trains and which should depart without waiting. Thus we have to determine which of the connections $a \in \mathcal{A}_{\text{change}}$ will be maintained and which will be removed. We denote the set of maintained connections by $\mathcal{A}_{\text{fix}}$. For the resulting network

$$\mathcal{N}(\mathcal{A}_{\text{fix}}) := (\mathcal{E}, \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{fix}} \cup \mathcal{A}_{\text{org}} \cup \mathcal{A}_{\text{dest}})$$

in which the set of changing activities has been replaced by $\mathcal{A}_{\text{fix}}$, a new timetable can be constructed using the critical path method (see Schöbel (2007)). The times for the events $e \in \mathcal{E}_{\text{dep}} \cup \mathcal{E}_{\text{arr}}$ in this new timetable will be denoted by x_e. For events $e \in \mathcal{E}_{\text{org}}$, we define $x_e = \pi_e$.

Given such a timetable and the connections that are maintained, for every OD pair, a route through the network has to be found, such that the travel time is minimized. To this end, let P be a directed path from e_1 to e_2 in the network $\mathcal{N}(\mathcal{A}_{\text{fix}})$, where $e_1 \in \mathcal{E} \setminus \mathcal{E}_{\text{dest}}$ and $e_2 \in \mathcal{E} \setminus \mathcal{E}_{\text{org}}$. We now define its length $l(P)$.

- First, assume that $e_1, e_2 \in \mathcal{E}_{\text{dep}} \cup \mathcal{E}_{\text{arr}}$. We define $l(P) = x_{e_2} - x_{e_1}$ to be the travel time or distance between e_1, e_2 in $\mathcal{N}(\mathcal{A}_{\text{fix}})$.

- We now extend this definition to destination events $Dest(p)$ for an OD pair $p = \{u, v, s_{uv}\}$. Let $pre(Dest(p), P)$ be the predecessor of $Dest(p)$ in path P from e_1 to $Dest(p)$. Then, we define $l(P) = x_{pre(Dest(p),P)} - x_{e_1}$.

- We are mainly interested in the travel time for the passengers. For the special case of a path P connecting an OD pair $p = \{u, v, s_{uv}\}$, we hence obtain $l(P) = x_{pre(Dest(p),P)} - s_{uv}$. As we assume that passengers take the fastest paths to arrive at their destinations, we set $l(p) = l(P(p))$, where $P(p)$ is a fastest path from the origin event $Org(p)$ to the destination event $Dest(p)$.

Because an OD pair p chooses a shortest path $P(p)$ from $Org(p)$ to $Dest(p)$ in $\mathcal{N}(\mathcal{A}_{\text{fix}})$, we define the arrival time for the pair p as $t_p^{\mathcal{A}_{\text{fix}}} = x_e$, where e is the predecessor of the destination event $Dest(p)$ on $P(p)$. Note that in this model, passengers who cannot take the path they planned to take because connections on this path are dropped, in most cases, will not have to suffer the delay of one cycle time waiting for the next connection but choose another path to their destination.

In the DM problem, we want to minimize the sum of all delays of the OD pairs. The delay of an OD pair $p = \{u, v, s_{uv}\}$ is given as

$$t_p^{\mathcal{A}_{\text{fix}}} - SP_p.$$

Summarizing, the objective of DMwRR is to find a subset $\mathcal{A}_{\text{fix}} \subset \mathcal{A}_{\text{change}}$, so that we minimize:

$$\min_{\mathcal{A}_{\text{fix}} \subset \mathcal{A}_{\text{change}}} \sum_{p \in \mathcal{P}} w_p \cdot \left(t_p^{\mathcal{A}_{\text{fix}}} - SP_p \right) \quad \text{or, equivalently} \quad \min_{\mathcal{A}_{\text{fix}} \subset \mathcal{A}_{\text{change}}} \sum_{p \in \mathcal{P}} w_p \cdot t_p^{\mathcal{A}_{\text{fix}}}.$$

In words: We minimize the average delay or the sum of the arrival times of the passengers. Since DM without rerouting is NP-hard (Gatto et al., 2005), it is not surprising that DMwRR is NP-hard as well. In Section 2.4, we will investigate the borderline between NP-hardness and tractability by analyzing the complexity of DMwRR for different structures of the underlying OD data.

2.3 Integer Programming Formulation

In this section, we will give an integer programming formulation that takes the routing decisions for the passengers into account explicitly. The model is based on the classical DM model as it was introduced in Schöbel (2007).
The event-activity network is a directed graph. We denote $\delta^{\text{in}}(e)$ and $\delta^{\text{out}}(e)$ for the set of arcs into e and out of e, respectively, for every event $e \in \mathcal{E}$.

2.3.1 Variables

The most important decision is which connections need to be kept alive. For each changing activity $a \in \mathcal{A}_{\text{change}}$, we thus introduce a binary decision variable z_a, which is defined as follows:

$$z_a = \begin{cases} 1 & \text{if connection } a \text{ is maintained,} \\ 0 & \text{otherwise.} \end{cases}$$

The times that the arrival and departure events take place are the next set of decision variables. For each event $e \in \mathcal{E}_{\text{arr}} \cup \mathcal{E}_{\text{dep}}$, we define $x_e \in \mathbb{N}$ as the rescheduled time that event e takes place. The variables $x = (x_e)$ therefore define the disposition timetable. These decision variables are the same as in the classical model.
The new aspect that we have to model are the routes that the passengers take. First note that a route has to be determined for every OD pair. Recall that $\mathcal{P}$ denotes the set of all OD pairs. To model the routing decisions for a given pair $p \in \mathcal{P}$, we introduce binary decision variables q_{ap}, which indicate whether activity $a \in \mathcal{A}$ is used in the path that is chosen for OD pair $p \in \mathcal{P}$. Formally, the variables q_{ap} are defined as follows:

$$q_{ap} = \begin{cases} 1 & \text{if activity } a \text{ is used by passengers in } p, \\ 0 & \text{otherwise.} \end{cases}$$

The arrival time for an OD pair p now depends both on the path that is chosen, and on the disposition timetable x. To be able to incorporate the arrival time of these passengers in a linear model, we introduce a variable $t_p \in \mathbb{N}$, which will represent the arrival time for pair $p \in \mathcal{P}$. The linearization will be given in the next section.

2.3.2 Integer programming formulation

We first present our integer programming formulation for DMwRR and then discuss its meaning.

$$\min \sum_{p \in \mathcal{P}} w_p(t_p - SP_p) \tag{2.1}$$

such that

$$
\begin{aligned}
x_e &\geq \pi_e + d_e && \forall e \in \mathcal{E}_{\text{arr}} \cup \mathcal{E}_{\text{dep}}, && (2.2)\\
x_e &\geq x_{e'} + L_a && \forall a = (e', e) \in \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}}, && (2.3)\\
x_e &\geq x_{e'} + L_a - M_1(1 - z_a) && \forall a = (e', e) \in \mathcal{A}_{\text{change}}, && (2.4)\\
q_{ap} &\leq z_a && \forall p \in \mathcal{P}, a \in \mathcal{A}_{\text{change}}, && (2.5)\\
\sum_{a \in \delta^{\text{out}}(e)} q_{ap} &= 1 && \forall e = Org(p) \in \mathcal{E}_{\text{org}}, && (2.6)\\
\sum_{a \in \delta^{\text{out}}(e)} q_{ap} &= \sum_{a \in \delta^{\text{in}}(e)} q_{ap} && \forall p \in \mathcal{P}, e \in \mathcal{E}_{\text{arr}} \cup \mathcal{E}_{\text{dep}}, && (2.7)\\
1 &= \sum_{a \in \delta^{\text{in}}(e)} q_{ap} && \forall e = Dest(p) \in \mathcal{E}_{\text{dest}}, && (2.8)\\
t_p &\geq x_e - M_2(1 - q_{ap}) && \forall e = Dest(p) \in \mathcal{E}_{\text{dest}}, a \in \delta^{\text{in}}(e), && (2.9)\\
z_a &\in \{0, 1\} && \forall a \in \mathcal{A}_{\text{change}}, && (2.10)\\
q_{ap} &\in \{0, 1\} && \forall p \in \mathcal{P}, a \in \mathcal{A}, && (2.11)\\
x_e &\in \mathbb{N} && \forall e \in \mathcal{E}_{\text{arr}} \cup \mathcal{E}_{\text{dep}}, && (2.12)\\
t_p &\in \mathbb{N} && \forall p \in \mathcal{P}. && (2.13)
\end{aligned}
$$

The objective function (2.1) minimizes the total delay of all passengers. Constraints (2.2) imply that events cannot take place earlier than in the original timetable and that source delays are taken into account. To make sure that delays are propagated through the network correctly, constraints (2.3) transfer the delay from the start of activity a to its end. For maintained connections, that is connections for which $z_a = 1$, constraints (2.4) transfer delays from the feeder train to the connecting train. The value of M_1 should be chosen large enough for these constraints to be correct. In Schöbel (2006), it has been shown that $M_1 = \max_{e \in \mathcal{E}} d_e$ is large enough. Constraints (2.2 - 2.4) are also present in

the classical model.
Constraints (2.5 - 2.9) take the routing decisions into account. First of all, constraints (2.5) make sure that changing activities can only be used if the connection is maintained. Equations (2.6 - 2.8) are the constraints of the shortest-path problem for each OD pair p. For every pair, a path is selected from the origin $Org(p) \in \mathcal{E}_{\text{org}}$ to the destination $Dest(p) \in \mathcal{E}_{\text{dest}}$. Constraint (2.9) defines the arrival time for OD pair p, where M_2 is again a large number. For the arrival event e that is selected and the destination activity a out of this event, $q_{ap} = 1$, forcing that $t_p \geq x_e$ for this particular arrival event. All other path variables q_{ap} are equal to zero, therefore putting no restriction on the value of t_p.
To find the minimal value of M_2 for which (2.9) is correct, consider an arbitrary OD pair $p \in \mathcal{P}$. As mentioned in Section 2.2, if $\max_{e' \in \mathcal{E}} d_{e'} \leq T$ only arrival events $e \in \mathcal{E}_{\text{arr}}$ for which $\pi_e \leq A_p + \max_{e' \in \mathcal{E}} d_{e'}$ should be connected to the destination event $Dest(p)$. Consider now an arbitrary activity $a = (e, Dest(p)) \in \delta^{\text{in}}(Dest(p))$. For $\max_{e' \in \mathcal{E}} d_{e'} \leq T$, it holds that

$$x_e \leq \pi_e + \max_{e' \in \mathcal{E}} d_{e'} \leq A_p + 2 \max_{e' \in \mathcal{E}} d_{e'}.$$

It follows that $M_2 = A_p - SP_p + 2 \max_{e' \in \mathcal{E}} d_{e'}$ is large enough. Indeed, as

$$x_e - M_2 \leq A_p + 2 \max_{e' \in \mathcal{E}} d_{e'} - M_2 = SP_p,$$

the constraint $t_p \geq x_e - M_2(1 - q_{ap})$ does not pose a restriction on t_p when $q_{ap} = 0$.
For a given OD pair $p \in \mathcal{P}$, Constraints (2.6 - 2.8) define a path from source $Org(p)$ to sink $Dest(p)$ in the directed graph $\mathcal{N}$. Many nodes in this graph can never be on such a path: Either it is impossible to arrive at such an event from the origin event $Org(p)$, or there is no path from that event to the destination event $Dest(p)$. The events that cannot be used on a path can be found easily by dynamic programming. This observation can be used to remove many of the variables q_{ap}. For an arc $a = (e, e') \in \mathcal{A}$ that connects a node that can never be used on a path from $Org(p)$ to $Dest(p)$, the variable q_{ap} will be zero in every feasible solution. We can therefore remove this variable from the formulation. Applying this procedure for all OD pairs $p \in \mathcal{P}$ reduces the number of binary variables drastically.
We remark that the variables z_a are not needed in the above model, because constraints (2.4) and (2.5) can be replaced by the constraints

$$x_e \geq x_{e'} + L_a - M(1 - q_{ap}) \quad \forall a = (e', e) \in \mathcal{A}_{\text{change}} \forall p \in \mathcal{P},$$

leading to an equivalent model. Nevertheless, we have chosen to leave these variables in the model to show the similarity with earlier models. Furthermore, the variables z_a could be used to guide the solution process.
In Section 2.5, we will use this formulation to analyze differences between DMwRR and the classical DM version without rerouting.

2.4 Special Cases of DMwRR and their Complexities

In the previous section, we gave an integer programming formulation for the general problem DMwRR. Now, we will identify simplifications and special cases of DMwRR to understand the border between still polynomial solvable and already NP-hard variants. The knowledge about the reasons for the NP-hardness as well as polynomial approaches for special cases can later serve to construct good heuristics for the general case.
In this section, we will hence examine three special cases of DMwRR. We first present a polynomial algorithm for the case of DMwRR where the demand is given by only one OD pair, and where we assume that there is no path in the event-activity network that enters a train more than once. We will then (slightly) generalize this case and allow that all OD pairs start at the same origin but have different destinations. It will turn out that even in this case, DMwRR is NP-hard. Finally, we will consider another variant with simplified delay costs. Although this is a strong simplification of DMwRR, it will turn out to be NP-hard as well.

2.4.1 DMwRR for one single OD pair

This subsection deals with a simplification of DMwRR: We assume that we are given only one OD pair $p = \{u, v, s_{uv}\}$. To simplify the notation in the following section, we will identify $Org(p)$ with u and $Dest(p)$ with v, so u and v will be regarded as events in the network. We will show that the problem is solvable by a modified version of Dijkstra's algorithm for finding a shortest path. The algorithm finds the optimal solution if there is no path in the network that enters a train more than once. In practice, this holds, for example, if trains do not overtake each other and if trains take the shortest routes between their end points. If such paths do exist, the algorithm finds a feasible solution, and thus an upper bound on the optimal solution value.
Let $\mathcal{N}$ be a network with feasible timetable π, $p = \{u, v, s_{uv}\}$ an OD pair, and $\mathcal{D}$ a set of source delays. Like in the original Dijkstra's algorithm, we determine in every step a feasible path for a pair of events $\{u, i\}$, where $u = Org(p)$ is the origin event of the

OD pair $p = \{u, v, s_{uv}\}$ under consideration and $i \in \mathcal{E}$. To do this formally, we need the following slight extension of DMwRR.

Having in mind the practical application in passenger rerouting, we defined in Section 2.2, the problem DMwRR for a network $\mathcal{N}$ and a set of OD pairs $\mathcal{P}$ consisting of elements of the form $p = \{u, v, s_{uv}\}$, where u is the origin, v the destination, and s_{uv} is the starting time. Now, we also want to deal with OD pairs as elements of the type $p^* = \{u, i, s_{uv}\}$, where $i \in \mathcal{E}$ is an *arbitrary* successor of u in $\mathcal{N}$. We hence extend the problem DMwRR to instances consisting of a network $\mathcal{N}$ and a set of OD pairs $\mathcal{P}$ of type p^*.

Let u be the origin event of the considered OD pair. Determining a feasible path for a fixed pair of events $\{u, i\}$ can hence be seen as solving DMwRR for $\mathcal{N}$ and $\mathcal{P} = \{\{u, i, s_{uv}\}\}$.

The part of Dijkstra's algorithm that has to be modified is the calculation of the node labels that represent the earliest possible times at the events. To calculate the transfer of delays efficiently, we define $tr[e]$ as the train belonging to an event $e \in \mathcal{E}_{\text{dep}} \cup \mathcal{E}_{\text{arr}}$.

Given a set of maintained connections $\mathcal{A}_{\text{fix}} \subset \mathcal{A}_{\text{change}}$, the minimal arrival times $x^{\mathcal{A}_{\text{fix}}}[e]$ in the events e considering the network $\mathcal{N}(\mathcal{A}_{\text{fix}})$ can be calculated iteratively:

Starting with $x^{\mathcal{A}_{\text{fix}}}[u] = s_{uv}$, we use the critical path method and obtain iteratively

$$x^{\mathcal{A}_{\text{fix}}}[e] = \max\{\pi[e] + d_e, \max_{i:(i,e) \in \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{fix}}} \{x^{\mathcal{A}_{\text{fix}}}[i] + L_{(i,e)}\}\}.$$

Let $\tilde{\pi}[i] = x^{\emptyset}[i]$ for all $i \in \mathcal{E}_{\text{arr}} \cup \mathcal{E}_{\text{dep}}$ denote the minimal arrival times calculated by the critical path method for the empty set of maintained connections and set $\tilde{\pi}[u] = \tilde{\pi}[v] = s_{uv}$.

We observe that for every set $\mathcal{A}_{\text{fix}} \subset \mathcal{A}_{\text{change}}$ and every event $e \in \mathcal{E}_{\text{arr}} \cup \mathcal{E}_{\text{dep}}$,

$$x^{\mathcal{A}_{\text{fix}}}[e] \geq \tilde{\pi}[e] \geq \pi[e] + d_e.$$

So we can equivalently determine the minimal arrival times in the events e for a given set $\mathcal{A}_{\text{fix}} \subset \mathcal{A}_{\text{change}}$ as

$$x^{\mathcal{A}_{\text{fix}}}[e] = \max\{\tilde{\pi}[e], \max_{i:(i,e) \in \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{fix}}} \{x^{\mathcal{A}_{\text{fix}}}[i] + L_{(i,e)}\}\}.$$

In Lemma 2.1 and Lemma 2.2, we will prove properties of the optimal set of connections $\mathcal{A}_{\text{fix}}$ and the path used by the passengers in $\mathcal{N}(\mathcal{A}_{\text{fix}})$ for the case of one single OD pair. These will lead to a simplification of the calculation of the minimal arrival times as given in Lemma 2.3.

Lemma 2.1 states that in any optimal solution for $p^* = \{u, e, s_{uv}\}$ with $e \in \mathcal{E}$, only the connections on the path used from u to e have to be maintained.

Lemma 2.1 *Let $\tilde{A}_{fix}$ be a set of maintained connections such that for an event $e \in \mathcal{E}$, the arrival time for $p^* = \{u, e, s_{uv}\}$ is minimal. Let $P = (\mathcal{E}^P, \mathcal{A}^P)$ be an optimal path from u to e in $\mathcal{N}(\tilde{A}_{fix})$. Then, there exists an optimal set of maintained connections $\mathcal{A}^P_{fix}$ such that $\mathcal{A}^P_{fix} = \mathcal{A}^P \cap \mathcal{A}_{change}$.*

Proof Obviously $\tilde{A}_{\text{fix}} \supset \mathcal{A}^P \cap \mathcal{A}_{\text{change}}$. On the other hand, if $a \in \tilde{A}_{\text{fix}} \setminus \mathcal{A}^P$, no passenger on path P uses a, so we can remove it. □

The statement of Lemma 2.2 is the following. There is always a path with minimal arrival time such that the passengers use every train at most once.

Lemma 2.2 *Let e be an event in $\mathcal{N}$. For every set $\mathcal{A}_{fix}$ for which there exists a path from the origin u to e in $\mathcal{N}(\mathcal{A}_{fix})$, there also exists a path $P = (\mathcal{E}^P, \mathcal{A}^P)$ from u to e that fulfills the following condition:*
If $j, k \in \mathcal{E}^P$ such that $x^{\mathcal{A}_{fix}}[j] < x^{\mathcal{A}_{fix}}[k]$ and $tr[j] \neq tr[k]$, then $tr[j] \neq tr[l]$ for all $l \in \mathcal{E}^P$ with $x^{\mathcal{A}_{fix}}[k] < x^{\mathcal{A}_{fix}}[l]$.

Proof Assume that P_0 is a path from u to e in $\mathcal{N}(\mathcal{A}_{\text{fix}})$ such that $tr[l] = tr[j] \neq tr[k]$ and $x^{\mathcal{A}_{\text{fix}}}[j] < x^{\mathcal{A}_{\text{fix}}}[k] < x^{\mathcal{A}_{\text{fix}}}[l]$ for $j, l, k \in P_0$ We construct a new path P_1 in the following way: P_1 consists of the same events as P_0 between u and j. Then P_1 continues on $tr[j]$ until l is reached. From l to e, P_1 once again contains the same events as P_0. Note that P_1 is also contained in $\mathcal{N}(\mathcal{A}_{\text{fix}})$. Repeating this construction, we obtain a path with the claimed property. □

The following lemma states that for computing the time of an event of path P, only the events of P are relevant.

Lemma 2.3 *For a path P from u to e in $\mathcal{N}(\mathcal{A}_{change})$ fulfilling the condition of Lemma 2.2 and the set $\mathcal{A}^P_{fix} = \mathcal{A}_{change} \cap \mathcal{A}^P$, the minimal time of event e can be calculated as*

$$x^{\mathcal{A}^P_{fix}}[e] = \max\{\tilde{\pi}[e], x^{\mathcal{A}^P_{fix}}[j] + L_{(j,e)}\}$$

for the predecessor j of e on the path P.

Proof If $(j, e) \in \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}}$, or if $(j, e) \in \mathcal{A}^P_{\text{fix}}$ and e is the departure of the train $tr[e]$, then (j, e) is the only activity terminating in e in $\mathcal{N}(\mathcal{A}^P_{\text{fix}})$, because $\mathcal{A}^P_{\text{fix}} = \mathcal{A}_{\text{change}} \cap \mathcal{A}^P$.

Thus

$$\begin{aligned}x^{\mathcal{A}^P_{\text{fix}}}[e] &= \max\{\tilde{\pi}[e], \max_{i:(i,e)\in\mathcal{A}_{\text{drive}}\cup\mathcal{A}_{\text{wait}}\cup\mathcal{A}^P_{\text{fix}}}\{x^{\mathcal{A}^P_{\text{fix}}}[i] + L_{(i,e)}\}\}\\ &= \max\{\tilde{\pi}[e], x^{\mathcal{A}^P_{\text{fix}}}[j] + L_{(j,e)}\}.\end{aligned}$$

Now let $(j,e) \in \mathcal{A}^P_{\text{fix}}$ and $(k,e) \in \mathcal{A}$ with $tr[k] = tr[e]$. Because of Lemma 2.2, $x^{\mathcal{A}^P_{\text{fix}}}[k] = \tilde{\pi}[k]$ and $\tilde{\pi}[e] \geq \tilde{\pi}[k] + L_{(k,e)} = x^{\mathcal{A}^P_{\text{fix}}}[k] + L_{(k,e)}$. Using $\mathcal{A}^P_{\text{fix}} = \mathcal{A}_{\text{change}} \cap \mathcal{A}^P$, it follows that

$$\begin{aligned}x^{\mathcal{A}^P_{\text{fix}}}[e] &= \max\{\tilde{\pi}[e], \max_{i:(i,e)\in\mathcal{A}_{\text{drive}}\cup\mathcal{A}_{\text{wait}}\cup\mathcal{A}^P_{\text{fix}}}\{x^{\mathcal{A}^P_{\text{fix}}}[i] + L_{(i,e)}\}\}\\ &= \max\{\tilde{\pi}[e], x^{\mathcal{A}^P_{\text{fix}}}[j] + L_{(j,e)}, x^{\mathcal{A}^P_{\text{fix}}}[k] + L_{(k,e)}\}\\ &= \max\{\tilde{\pi}[e], x^{\mathcal{A}^P_{\text{fix}}}[j] + L_{(j,e)}\}.\end{aligned}$$

□

Let's come back to our modified Dijkstra's algorithm. We solve problem DMwRR for different events i. In any iteration, we store

- $T[i]$: Minimal arrival time in event i for passengers traveling from u to i with starting time s_{uv}.
- $\mathcal{A}_{\text{fix}}[i]$: Changing activities that have to be maintained in the optimal solution of DMwRR with OD pair $\{u, i, s_{uv}\}$.
- $TD[i]$: Set of "forbidden trains" = trains that were used on the minimal path from u to i, not including $tr[i]$ since these may not be used anymore according to Lemma 2.2. Note that if one would allow passengers to reenter a train, solutions can be found that are infeasible.

Let $PERM$ be the set of events for which DMwRR has been solved and the above values have been determined. For every e with a direct predecessor $i \in PERM$, we determine the preliminary arrival time

$$\tilde{T}[e] = \min_{i\in PERM:(i,e)\in\mathcal{A}, tr[e]\notin TD[i]}\{\tilde{\pi}[e], T[i] + L_{(i,e)}\}.$$

Like in Dijkstra's algorithm, we fix the event $\hat{e}$ with smallest $\tilde{T}[e]$.
To calculate the set of fixed connections $\mathcal{A}_{\text{fix}}[\hat{e}]$ and the set of delayed trains $TD[\hat{e}]$, we distinguish two cases. Let $i_{\hat{e}}$ be the predecessor of $\hat{e}$ in the solution of DMwRR for $\{u, \hat{e}, s_{uv}\}$.

- If $\hat{a} = (i_{\hat{e}}, \hat{e})$ is a changing activity, we obtain $\mathcal{A}_{\text{fix}}[\hat{e}] = \mathcal{A}_{\text{fix}}[i_{\hat{e}}] \cup \{(i_{\hat{e}}, \hat{e})\}$ and $TD[\hat{e}] = TD[i_{\hat{e}}] \cup \{tr[i_{\hat{e}}]\}$.
- Otherwise, we simply set $\mathcal{A}_{\text{fix}}[\hat{e}] = \mathcal{A}_{\text{fix}}[i_{\hat{e}}]$ and $TD[\hat{e}] = TD[i_{\hat{e}}]$.

The algorithm is summarized below.

Algorithm: Modified Dijkstra for DMwRR for one OD pair

Input: Instance of DMwRR with network $\mathcal{N}$, feasible timetable π, delays d_e, and one OD pair $p = \{u, v, s_{uv}\}$.

Step 1. Generate the timetable $\tilde{\pi}$ by the critical path method. Set $\tilde{\pi}[u] = \tilde{\pi}[v] = s_{uv}$.

Step 2. Set $PERM = \{u\}$, $TEMP = \mathcal{E} \setminus \{u\}$, $T[u] = s_{uv}$, $\tilde{T}[e] = \infty$ for every $e \in TEMP$, $TD[u] = \emptyset$, $\mathcal{A}_{\text{fix}}[u] = \emptyset$, $\hat{e}^{\text{old}} = u$.

Step 3. For every $e \in TEMP$ such that $(\hat{e}^{\text{old}}, e) \in \mathcal{A}$, $tr[e] \notin TD[e^{\text{old}}]$
set $\tilde{T}[e] = \min\{\tilde{T}[e], \max\{\tilde{\pi}[e], T[\hat{e}^{\text{old}}] + L_{(\hat{e}^{\text{old}}, e)}\}\}$.

Step 4. Choose $\hat{e} \in \text{argmin}_{e \in TEMP} \tilde{T}[e]$. Set $i_{\hat{e}}$ the corresponding predecessor of $\hat{e}$, $PERM = PERM \cup \{\hat{e}\}$, $TEMP = TEMP \setminus \{\hat{e}\}$, $T[\hat{e}] = \tilde{T}[\hat{e}]$.

Step 5. If $\hat{e} = v$, go to Step 7.

Step 6. If $(i_{\hat{e}}, \hat{e}) \in \mathcal{A}_{\text{change}}$, set $\mathcal{A}_{\text{fix}}[\hat{e}] = \mathcal{A}_{\text{fix}}[i_{\hat{e}}] \cup \{(i_{\hat{e}}, \hat{e})\}$ and $TD[\hat{e}] = \{TD[i_{\hat{e}}] \cup \{tr[i_{\hat{e}}]\}\}$.
Otherwise, set $\mathcal{A}_{\text{fix}}[\hat{e}] = \mathcal{A}_{\text{fix}}[i_{\hat{e}}]$ and $TD[\hat{e}] = TD[i_{\hat{e}}]$.
Set $\hat{e}^{\text{old}} = \hat{e}$. Go to Step 3.

Step 7. Set $\mathcal{A}_{\text{fix}} = \mathcal{A}_{\text{fix}}[v]$ and $t_p = T[v]$.

Output: Optimal set $\mathcal{A}_{\text{fix}}$ for the given instance of DMwRR.

Theorem 2.4 *The algorithm finds an optimal solution $\mathcal{A}_{fix}$ to DMwRR with one OD pair if no path in the network enters a train more than once. In general, it finds a feasible solution and an upper bound. The running time of the algorithm is $O(n^2)$, where n is the number of events in the network $\mathcal{N}$.*

Proof First, we observe that adding changing activities to a set A_1 does not influence the time for events e that happen before the added activities take place.
That means for two sets $A_1 \subset A_2 \subset \mathcal{A}_{\text{change}}$ and an event e, the following statement holds: If e takes place before any activity in $A_2 \setminus A_1$ starts, i.e.

$$x^{A_2}(e_1) \geq x^{A_2}(e) \text{ for all } (e_1, e_2) \in A_2 \setminus A_1,$$

it follows that

$$x^{A_1}[e] = x^{A_2}[e]. \tag{2.14}$$

Now, we will show inductively that in every iteration of the algorithm for every event $e \in PERM$, it holds that

$$T[e] = x^{\mathcal{A}_{\text{fix}}[e]}[e]$$

for the labels $T[e]$ and the sets of changing activities $\mathcal{A}_{\text{fix}}[e]$ calculated by the algorithm. As $T[u] = s_{uv} = x^{\emptyset}[u] = x^{\mathcal{A}_{\text{fix}}[u]}[u]$, the assumption holds for the origin u. Assume that in the k-th iteration of the algorithm, the assumption holds for the events in $PERM$. Let $\hat{e}$ be the event that is chosen in Step 4 of the algorithm and $i_{\hat{e}} \in PERM$ such that $(i_{\hat{e}}, \hat{e}) \in \mathcal{A}$ and $\max\{\tilde{\pi}[\hat{e}], x^{\mathcal{A}_{\text{fix}}[i_{\hat{e}}]}[i_{\hat{e}}] + L_{(i_{\hat{e}},\hat{e})}\}$ is minimal. Let $\tilde{T}^{\text{old}}[\hat{e}]$ be the label of $\hat{e}$ at the beginning of Step 3, and let $\hat{e}^{\text{old}}$ be the event that was added to $PERM$ in the $(k-1)$-th iteration. Then the new label of $\hat{e}$ is calculated as

$$\begin{aligned}\tilde{T}[\hat{e}] &= \begin{cases} \tilde{T}^{\text{old}}[\hat{e}] & \text{if } tr[\hat{e}] \in TD[\hat{e}^{\text{old}}] \\ \min\{\tilde{T}^{\text{old}}[\hat{e}], \max\{\tilde{\pi}[\hat{e}], T[\hat{e}^{\text{old}}] + L_{(\hat{e}^{\text{old}},\hat{e})}\}\} & \text{if } tr[\hat{e}] \notin TD[\hat{e}^{\text{old}}] \end{cases} \\ &= \min_{i \in PERM:(i,\hat{e}) \in \mathcal{A}, tr[e] \notin TD[i]} \max\{\tilde{\pi}[\hat{e}], T[i] + L_{(i,\hat{e})}\}.\end{aligned}$$

Let $\mathcal{A}^P_{\text{fix}} = \mathcal{A}^P \cap \mathcal{A}_{\text{change}}$ be the set of changing activities contained in P like in Lemma 2.3. We note that because of the construction of $\mathcal{A}_{\text{fix}}[e]$ for an event e,

$$\mathcal{A}_{\text{fix}}[e] = \mathcal{A}^P_{\text{fix}} \tag{2.15}$$

for the path P used by the Dijkstra algorithm as a feasible path from u to e.

Let P be the path from u to $\hat{e}$ and P' be the path from u to the predecessor $i_{\hat{e}}$ of e in P with $P' \subset P$. Then

$$
\begin{aligned}
T[\hat{e}] &= \min_{i \in PERM : (i,\hat{e}) \in \mathcal{A}, tr[e] \notin TD[i]} \max\{\tilde{\pi}[\hat{e}], T[i] + L_{(i,\hat{e})}\} && (2.16)\\
&= \min_{i \in PERM : (i,\hat{e}) \in \mathcal{A}, tr[e] \notin TD[i]} \max\{\tilde{\pi}[\hat{e}], x^{\mathcal{A}_{\text{fix}}[i]}[i] + L_{(i,\hat{e})}\} && (2.17)\\
&= \max\{\tilde{\pi}[\hat{e}], x^{\mathcal{A}_{\text{fix}}[i_{\hat{e}}]}[i_{\hat{e}}] + L_{(i_{\hat{e}},\hat{e})}\} && (2.18)\\
&= \max\{\tilde{\pi}[\hat{e}], x^{\mathcal{A}_{\text{fix}}^{P'}}[i_{\hat{e}}] + L_{(i_{\hat{e}},\hat{e})}\} && (2.19)\\
&= \max\{\tilde{\pi}[\hat{e}], x^{\mathcal{A}_{\text{fix}}^{P}}[i_{\hat{e}}] + L_{(i_{\hat{e}},\hat{e})}\} && (2.20)\\
&= x^{\mathcal{A}_{\text{fix}}^{P}}[\hat{e}] && (2.21)\\
&= x^{\mathcal{A}_{\text{fix}}[\hat{e}]}[\hat{e}], && (2.22)
\end{aligned}
$$

where we use (2.15) in (2.19) and (2.22). (2.17) holds because of the assumption $T[i] = x^{\mathcal{A}_{\text{fix}}[i]}[i]$ for all $i \in PERM$ and (2.20) is true because of the initial observation (2.14). Assume now that there is no path that enters a train more than once. It remains to show that the set $\mathcal{A}_{\text{fix}}[\hat{e}]$ and the label $T[\hat{e}] = x^{\mathcal{A}_{\text{fix}}[\hat{e}]}[\hat{e}]$ are optimal for the event $\hat{e}$ chosen in Step 4 of the algorithm, that means that there is no set $A \subset \mathcal{A}_{\text{change}}$ such that there is a path from u to $\hat{e}$ in $\mathcal{N}(A)$ and

$$x^{A}[\hat{e}] < x^{\mathcal{A}_{\text{fix}}[\hat{e}]}[\hat{e}].$$

This assumption will also be proven inductively. For the origin event u, setting $\mathcal{A}_{\text{fix}}[u] = \emptyset$ leads to $T[u] = s_{uv}$, which is optimal. Suppose that in the iterations 1 to $k-1$ of the algorithm, the choice of $\mathcal{A}_{\text{fix}}[e]$ and the labels $T[e]$ are optimal for the regarded events e. Now, let $\hat{e}$ be the event chosen in Step 4 in the k-th iteration, i.e. such that $T[\hat{e}] = \tilde{T}[\hat{e}] \leq \tilde{T}[e]$ for every $e \in TEMP$. Suppose that there is a set $A \subset \mathcal{A}_{\text{change}}$ such that there is a path from u to $\hat{e}$ in $\mathcal{N}(A)$ and

$$x^{A}[\hat{e}] < x^{\mathcal{A}_{\text{fix}}[\hat{e}]}[\hat{e}]. \tag{2.23}$$

Let $P^{A}_{u\hat{e}}$ be an optimal path from u to $\hat{e}$ in $\mathcal{N}(A)$, that satisfies the conditions of Lemma 2.2. We assume that no path enters a train more than once. Therefore the condition $tr[\hat{e}] \notin TD[e]$ holds trivially for all $e \in TEMP$.

(1) If the predecessor e_0 of $\hat{e}$ in $P^A_{u\hat{e}}$ is in $PERM$, because of the assumption that the labels $T[e]$ and chosen sets $\mathcal{A}_{\text{fix}}[e]$ are optimal for all $e \in PERM$,

$$\begin{aligned}
x^A[\hat{e}] &= \max\{\tilde{\pi}[\hat{e}], T[e_0] + L_{(e_0,\hat{e})}\} \\
&\geq \min_{i \in PERM:(i,\hat{e}) \in \mathcal{A}, tr[\hat{e}] \notin TD[i]} \max\{\tilde{\pi}[\hat{e}], T[i] + L_{(i,\hat{e})}\} \\
&= T[\hat{e}] = x^{\mathcal{A}_{\text{fix}}[\hat{e}]}[\hat{e}],
\end{aligned}$$

which contradicts (2.23).

(2) If the predecessor e_0 of $\hat{e}$ in $P^A_{u\hat{e}}$ is in $TEMP$, let e_1 denote the last event in $PERM$ on the path $P^A_{u\hat{e}}$ (e_1 exists because $u \in PERM$) and $e_2 \in TEMP$ its successor. So as $T[\hat{e}] = \tilde{T}[\hat{e}] \leq \tilde{T}[e]$ for every $e \in TEMP$,

$$\begin{aligned}
x^A[\hat{e}] &\geq x^A[e_2] \\
&\geq \max\{\tilde{\pi}[e_2], T[e_1] + L_{(e_1,e_2)}\} \\
&\geq \min_{i \in PERM:(i,e_2) \in \mathcal{A}, tr[e_2] \notin TD[i]} \max\{\tilde{\pi}[e_2], T[i] + L_{(i,e_2)}\} \\
&= \tilde{T}[e_2] \geq \tilde{T}[\hat{e}] = T[\hat{e}] = x^{\mathcal{A}_{\text{fix}}[\hat{e}]}[\hat{e}],
\end{aligned}$$

which contradicts (2.23).

For a set of fixed connections A, denote by t^A_p the traveling time for OD pair p in $\mathcal{N}(A)$. It remains to show that $\mathcal{A}_{\text{fix}} = \mathcal{A}_{\text{fix}}[v]$ and $t^{\mathcal{A}_{\text{fix}}}_p = T[v]$ is an optimal solution to DMwRR for the OD pair $p = \{u, v, s_{uv}\}$. As defined in Section 2.2, $\mathcal{A}_{\text{fix}}[v]$ is optimal if it minimizes $t^{\mathcal{A}_{\text{fix}}}_p = x^{\mathcal{A}_{\text{fix}}}[e]$ for the predecessor e of v on a minimal path from u to v in the network $\mathcal{N}(\mathcal{A}_{\text{fix}})$. Suppose that the set $\mathcal{A}_{\text{fix}}[v]$ and the predecessor e calculated by the algorithm are not optimal with regard to an optimal path from u to v. Let A be an optimal set, P^A_{uv} an optimal path in $\mathcal{N}(A)$, and e_0 the optimal predecessor. Then

$$t^A_p < t^{\mathcal{A}_{\text{fix}}}_p. \tag{2.24}$$

(1) If $e_0 \in PERM$, because of the assumption that the labels $T[i]$ and chosen sets $\mathcal{A}_{\text{fix}}[i]$ are optimal for all $i \in PERM$,

$$\begin{aligned} t_p^A &= \max\{\tilde{\pi}[v], x^A[e_0] + L_{(e_0,v)}\} \\ &\geq \min_{i \in PERM:(i,v)\in\mathcal{A}} \max\{\tilde{\pi}[v], T[i] + L_{(i,v)}\} \\ &= \min_{i \in PERM:(i,v)\in\mathcal{A}} \max\{s_{uv}, T[i] + L_{(i,v)}\} \\ &= T[v] \\ &= t_p^{\mathcal{A}_{\text{fix}}}, \end{aligned}$$

which contradicts (2.24).

(2) If the predecessor e_0 of v in P_{uv}^A is in $TEMP$. let e_1 denote the last event in $PERM$ on the path P_{uv}^A and $e_2 \in TEMP$ its successor. So as $T[v] = \tilde{T}[v] \leq \tilde{T}[e]$ for $e \in TEMP$.

$$t_p^A = x^A[e_0] \geq x^A[e_2] = \max\{\tilde{\pi}[e_2], T[e_1] + L_{(e_1,e_2)}\} \geq \tilde{T}[e_2] \geq \tilde{T}[v] = T[v] = x^{\mathcal{A}_{\text{fix}}[v]}[v],$$

which contradicts (2.24).

The generation of the timetable in Step 1 is done in time $O(n^2)$ by the procedure given in Schöbel (2007). Inspecting Step 3, we note that for the setting of the labels $\tilde{T}$, summing up over all iterations every activity $a \in \mathcal{A}$ has to be considered at most once. As Steps $4-6$ are done in time $O(n)$, Steps $3-6$ can be executed in $O(n^2)$. As Step 7 is also in $O(n^2)$, the running time of the modified Dijkstra algorithm is $O(n^2)$. □

We can use this algorithm to determine a lower bound on the optimal solution for the general DMwRR problem as follows: We apply the modified Dijkstra algorithm for DMwRR with one OD pair for every OD pair $p \in \mathcal{P}$, and sum up over the solution values weighted with w_p. This gives us a lower bound on the solution of DMwRR.

Lemma 2.5 *Let DMwRR be given together with a set of OD pairs $\mathcal{P}$ and let z^* be its optimal objective value. Assume that no path in the network enters a train more than once. For any OD pair $p \in \mathcal{P}$, let f_p denote the objective value of the reduced problem DMwRR with only the OD pair $p = \{u, v, s_{uv}\}$. Then*

$$t_p \geq f_p$$

is a valid inequality for our integer programming formulation for DMwRR for any $p \in \mathcal{P}$. In particular, we have

$$\sum_{p \in \mathcal{P}} w_p f_p \leq z^*.$$

Moreover, this bound on the objective value can be calculated in in $O(|\mathcal{P}|n^2)$.

This bound significantly improves the time needed to solve the integer program for DMwRR as will be shown in Section 2.5. Note that it is easy to generate an upper bound by holding all connections (set all $z_a = 1$) such that we can now bound the objective value from above and below.

2.4.2 DMwRR for a tree-like structure of the demand

In Section 2.4.1, we have seen that for the case of only one OD pair, DMwRR is solvable in polynomial time and we have given a Dijkstra-type solution algorithm for this case. Dijkstra-type algorithms can usually be generalized to trees. We hence investigate the question if the case of multiple OD pairs in which all passengers have the same origin and starting time can still be solved by the approach of the previous section. However, this generalization of our problem already turns out to be strongly NP-hard.

Theorem 2.6 *DMwRR is strongly NP-hard, even if only one delay occurs, all origin and destination events are connected to only one event in the network and all OD pairs $\{u_k, v_k, s_{u_k v_k}\}$ have the same origin $u_k := u$ and same starting time $s_{u_k v_k} := 0$.*

Proof This theorem will be proven by reduction to the NP-complete decision problem *minimum cover* (see Garey and Johnson (1979)). An instance of minimum cover consists of a finite set $S = \{s_j : j = 1, \ldots, n\}$, a collection $C = \{c^i : i = 1, \ldots, m\}$ of subsets of S, and a positive integer $K \leq |C|$. The question to decide is whether there is a subset C' of C with $|C'| \leq K$ such that every element of S is contained in at least one element of C'. The structure of the minimum cover problem can be represented by a matrix $M = (m_{ij})_{i=1,\ldots,m,j=1,\ldots,n}$ with

$$m_{ij} = \begin{cases} 1 & \text{if } s_j \in c^i \\ 0 & \text{otherwise.} \end{cases}$$

We construct an instance of the DMwRR in which S corresponds to the OD pairs and C to connections for which we have to decide whether they are maintained or not. We have to cover all OD pairs (i.e. make sure that all passengers reach their destinations)

with a minimal set of maintained connections since maintaining a connection causes costs (represented as delays) to other passengers. Our construction is the following.
For a given instance (S, C, K) of minimum cover, we transfer the matrix M to a set of stations $A = \{a_{ij} : m_{ij} = 1\}$, that means whenever $m_{ij} = 1$ for $1 \leq i \leq m$ and $1 \leq j \leq n$, there is a station a_{ij}. There are trains tr^i for $i = 1, \ldots, m$ starting all at a station a_0 and running through the existing stations a_{ij} in increasing order of j and trains tr_j running through the existing stations a_{ij} in increasing order of i. The stations a_{ij} offer the possibility to change from train tr^i to train tr_j, so if and only if $m_{ij} = 1$, it is possible to change from tr^i to tr_j. There are no slack times on the driving or waiting activities of the trains tr^i and tr_j, as well as on the changing activities between these trains, so any delay of a train tr^i will propagate to all following events of the train and to the events of the trains tr_j if the changing activity between these trains at station a_{ij} is maintained.
There are destinations v_j for $j = 1, \ldots, n$ that are reached by train tr_j after the last station a_{ij} that this train passes.
We introduce an origin u with one train starting there, going to a station a_0. At station a_0 there are changing activities to the departure events of the trains tr^i, having no slack times. In this construction, the passengers wanting to travel from u to v_j at station a_0 have to choose a train tr^i such that they will be able to change to the train tr_j and reach their destination, that means a train tr^i such that a_{ij} exists.
We suppose that a delay of 1 at the arrival event of train tr^0 at station a_0 occurs. As there are no slack times on the changing activities to the trains tr^i, if a connection to a train tr^i is maintained, this train will receive a delay of 1. When passengers change to a train tr_j at station a_{ij}, this delay will be transferred to the train tr_j, thus all passengers of the OD pairs $\{u, v_j, 0\}$ arrive at their destinations with a delay of 1.
Now, we assume that there are more OD pairs $\{u, v^1, 0\}, \ldots, \{u, v^m, 0\}$ that can reach their destination via the trains tr^i. To this end, for every $i = 1, \ldots, m$, we introduce two more stations a_{in+1} and v^i. We assume that the train tr^0 after leaving a_0 runs through the stations a_{in+1} for all $i = 1, \ldots, m$. At the driving activity from a_0 to a_{1n+1}, we set the slack time to be 1, thus when the train arrives at station a_{1n+1}, despite of the delay of 1 occurring at the arrival event in a_0, it is not delayed anymore. We allow the passengers to change from tr^0 to tr^i at every station a_{in+1}. So we can assume that all passengers wanting to travel from u to v^i for an $i = 1, \ldots, m$ will take train tr^0 until station a_{in+1}, and thus will be delayed if and only if the connection from tr^0 to tr^i at station a_0 is maintained. We will denote the constructed event-activity network by $\mathcal{N}$.
In Figure 2.3, the station network for an instance of DMwRR constructed from an instance of minimal cover with $S = \{s_1, s_2\}$ and $C = \{\{s_1\}, \{s_2\}, \{s_1, s_2\}\}$ is pictured.

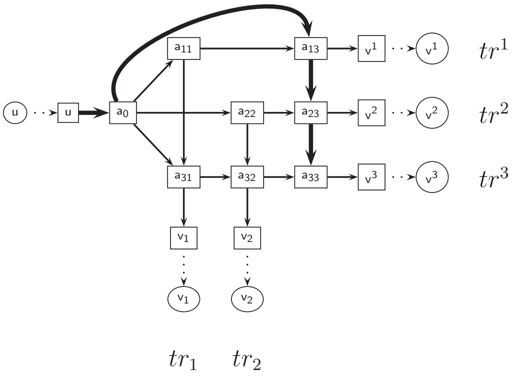

Figure 2.3: The station network for the instance of the DM problem with rerouting constructed from an instance of minimum cover with $S = \{s_1, s_2\}$ and $C = \{\{s_1\}, \{s_2\}, \{s_1, s_2\}\}$. The squared nodes represent stations; the oval nodes represent origin and destination events. There are six trains, tr^0 represented by the thick line, tr^1, tr^2 and tr^3 starting at station a_0 and going from left to right, and tr_1 and tr_2 going top down.

Setting the number of passengers to be $w_{uv^i} = w_{uv_j} = 1$, we can now show that the instance (S, C, K) of minimum cover has a solution if and only if there is a set $\mathcal{A}_{\text{fix}}$ such that the sum over the delays of the OD pairs in the network $\mathcal{N}(\mathcal{A}_{\text{fix}})$ is smaller than or equal to $\tilde{K} = n + K$.

We divide $\mathcal{A}_{\text{change}}$ into two sets: the changing activities at station a_0 $\mathcal{A}^1_{\text{change}} := \{(tr^0 - a_0 - Arr, tr^i - a_0 - Dep) : i = 1, \ldots, m\}$ and all other changing activities $\mathcal{A}^2_{\text{change}} := \{(tr^i - a_{ij} - Arr, tr_j - a_{ij} - Dep)\} \cup \{(tr^0 - a_{in+1} - Arr, tr^i - a_{in+1} - Dep)\}$. Now, we observe that maintaining a connection in $\mathcal{A}^2_{\text{change}}$ does not yield a delay for any OD pair. So we choose to maintain all connections in $\mathcal{A}^2_{\text{change}}$.

For a solution C' of minimum cover, we set $\mathcal{A}_{\text{fix}}(C') := \mathcal{A}^2_{\text{change}} \cup \{(tr^0 - a_0 - Arr, tr^i - a_0 - Dep) : c^i \in C'\}$ and vice versa for a solution $\mathcal{A}_{\text{fix}} \supset \mathcal{A}^2_{\text{change}}$, we define $C'(\mathcal{A}_{\text{fix}}) = \{c^i : (tr^0 - a_0 - Arr, tr^i - a_0 - Dep) \in \mathcal{A}_{\text{fix}}\}$. Thus we have a bijection between subsets $C' \subset C$ and $\mathcal{A}_{\text{fix}} \supset \mathcal{A}^2_{\text{change}}$.

Let $A \subset \mathcal{A}^1_{\text{change}}$. We see that in $\mathcal{N}(A \cup \mathcal{A}^2_{\text{change}})$, there exists a path from u to v_j if and only if at least for one i with $s_j \in c^i$, the connection $(tr^0 - a_0 - Arr, tr^i - a_0 - Dep)$ is maintained.

Thus a set $\mathcal{A}_{\text{fix}}$ is feasible for DMwRR (that means for every OD pair, there exists a path

from origin to destination) if and only if the corresponding set C' is feasible for minimum cover.
Furthermore, we observe that the OD pairs $\{\{u, v_j, 0\} : j = 1, \dots, n\}$ will reach their destination with a delay of 1, because there are no slack times on the paths from u to v_j for all $j = 1, \dots, n$. For the other OD pairs $\{u, v^i, 0\}$, the delay is 1 if the connection $(tr^0 - a_0 - Arr, tr^i - a_{ij} - Dep)$ is maintained, and 0 otherwise.
Thus for a pair of solutions $\mathcal{A}_{\text{fix}} \supset \mathcal{A}^2_{\text{change}}$ and $C' \subset C$ with solution values $z(\mathcal{A}_{\text{fix}})$ and $z(C')$

$$z(\mathcal{A}_{\text{fix}}) = z(C') + n.$$

That means there is a solution to the constructed instance of DMwRR with solution value $\leq \tilde{K}$ if and only if there is a solution to (S, C, K) with solution value $\leq K$.
So every instance of minimum cover can be transformed polynomially to an instance of DMwRR with the claimed properties. □

2.4.3 Rerouting with simplified costs

The delays that arise in DMwRR for the passengers by the wait-depart decisions for the connections can be divided into the following two types:

(1) A connection is maintained: The waiting train and the passengers on the waiting train are delayed.

(2) A connection is not maintained: The passengers who wanted to take this connection have to travel along another, probably longer path.

Calculating the delay of the first type by a heuristic approach motivates the following rerouting problem with simplified costs.
Let $\mathcal{N} = (\mathcal{E}, \mathcal{A})$ be a directed network with edge lengths L_a for all $a \in \mathcal{A}$. Let $\mathcal{A}_{\text{change}} \subset \mathcal{A}$ be a set of connections that can be maintained or removed. We assume that maintaining a connection $a \in \mathcal{A}_{\text{change}}$ yields a fixed delay of d_a for the passengers. This delay can be regarded as a cost for opening a connection that is added to the solution value. Let $\mathcal{P}$ be a set of OD pairs, given as a subset of $\mathcal{E} \times \mathcal{E}$ with demand w_p for each $p = \{u, v\} \in \mathcal{P}$. The objective of this variant is to minimize the simplified costs arising as fixed delays for maintaining connections plus the travel costs of the OD pairs. Hence the objective

function is

$$\min_{\mathcal{A}_{\text{fix}} \subset \mathcal{A}_{\text{change}}} \sum_{p \in \mathcal{P}} w_p \cdot D_{\mathcal{A}_{\text{fix}}}(u,v) + \sum_{a \in \mathcal{A}_{\text{fix}}} d_a,$$

where $D_{\mathcal{A}_{\text{fix}}}(u,v) = \sum_{a \in P_{uv}} L_a$ with P_{uv} being a shortest path from u to v in the network in which all connections $a \in \mathcal{A}_{\text{change}} \setminus \mathcal{A}_{\text{fix}}$ are removed.
In contrast to DMwRR, in this simplified variant, we are trying to minimize shortest path distances regarding the edge lengths L_a and not arrival times. The simplified delays arise as costs or penalties, whenever a connection $a \in \mathcal{A}_{\text{fix}}$ is maintained and do not influence other parts of the network, while in DMwRR, they can propagate through big parts of the network and delay the following events.
Like in DMwRR this problem can be solved in polynomial time if there is only one OD pair (by adding the simplified costs d_a divided by the demand of the OD pair w_p for a connection a to its length L_a and applying Dijkstra's algorithm). However, by modifying the proof of Theorem 2.6, it can be shown that even this simplified variant is strongly NP-hard.

Theorem 2.7 *Rerouting with simplified costs is strongly NP-hard, even if all origin and destination events are connected to only one event in the network and all OD pairs $\{u_k, v_k\}$ have the same origin $u_k := u$.*

Proof Analogously to the proof of strong NP-hardness for DMwRR, we can prove this theorem by constructing an equivalent rerouting with a simplified costs problem for each instance of minimum cover. For this proof, we simplify the network $\mathcal{N}$ from the proof of Theorem 2.6 to a network $\tilde{\mathcal{N}}$ by removing all events at the stations a_{in+1} and v^i, the destination events v^i, and all related activities. The set of OD pairs is $\mathcal{P} = \{\{u, v_j\} : j = 1, \ldots, m\}$ with unit demand. For the changing activities from tr^0 to tr^i, we set the simplified costs to 1, for the ones from tr^i to tr^j to 0. Similar to the proof of Theorem 2.6, we observe that we can assume the connections $(tr^i - a_{ij} - Arr, tr_j - a_{ij} - Dep)$ to be maintained because their simplified costs are 0. Like in that proof for a given set C' of subsets of S, we define

$$\mathcal{A}_{\text{fix}}(C') := \{(tr^0 - a_0 - Arr, tr^i - a_0 - Dep) : c_i \in C'\} \cup \{(tr^i - a_{ij} - Arr, tr_j - a_{ij} - Dep) : m_{ij} = 1\},$$

and for a given subset $\mathcal{A}_{\text{fix}} \supset \{(tr^i - a_{ij} - Arr, a_{ij} - tr_j - Dep) : m_{ij} = 1\}$, we set

$$C'(\mathcal{A}_{\text{fix}}) = \{c_i : (tr^0 - a_0 - Arr, tr^i - a_0 - Dep) \in \mathcal{A}_{\text{fix}}\}.$$

Now, a set C' of subsets of S and the associated subset $\mathcal{A}_{\text{fix}} \subset \mathcal{A}_{\text{change}}$ are both feasible or infeasible and have the same objective value as can be seen analogously to the proof of Theorem 2.6. □

2.5 Computational Experiments

We have created six cases to evaluate the integer programming formulation given in Section 2.3. We will first describe the cases that we consider. Then, we will discuss the solution of the integer programs and show that the polynomial algorithm for one OD pair can be applied to improve the overall running times. Finally, we will demonstrate that taking rerouting into account reduces the total delay significantly.

2.5.1 Cases

In all cases, we consider a part of the railway network in the Netherlands during a period in the late evening. For all possible OD pairs $p = \{u, v, s_{uv}\}$, the *average* number of passengers over several days is known. We denote this average by $\bar{w}_p$. The first case corresponds to the example described in Section 2.1. The stations that are included in the first case are represented by black circles in Figure 2.1. We focus on the situation where the train from Zwolle arrives in Amersfoort with a delay. Therefore, only a short time of two hours in the evening is taken into account. As described in Section 2.1, three intercities and one regional train are considered. In the first case, we consider the average number of passengers. For each OD pair p, we thus set $w_p = \bar{w}_p$. A time period of four hours is considered in the other cases. As all trips of passengers and trains take less than four hours, it does not make sense to consider longer time periods. The second case considers the stations that are indicated by a grey or black circle in Figure 2.1. All long-distance trains between those stations are included. The third case contains all-long distance trains in the Randstad, which is the Western, most populated part of the Netherlands. This case includes all stations in Figure 2.1. Note that in the second and third cases, the regional trains are not considered. As the long-distance trains only stop at larger stations, the average number of passengers for each OD pair is quite large. It is therefore reasonable to set $w_p = \bar{w}_p$ here as well. The fourth case is an extension of the second one. It includes also the regional trains. When the regional trains are taken into account, the number of changing activities grows enormously, because much more trains depart at each station during a given time interval. As the regional trains also stop at smaller stations, there are many OD pairs for which the average number of passengers is small. We therefore create two cases: one with the average passenger figures, and one with a possible realization of the passenger figures. For the former, we again set $w_p = \bar{w}_p$. For the latter, let $\bar{w}_p$ be the average number of passengers for OD pair p. Denote $f_p = \bar{w}_p - \lfloor \bar{w}_p \rfloor$ for the fractional part of $\bar{w}_p$. For each OD pair p, we determine the number of passengers as

$$w_p = \begin{cases} \lfloor \bar{w}_p \rfloor + 1 & \text{with probability } f_p, \\ \lfloor \bar{w}_p \rfloor & \text{otherwise.} \end{cases}$$

Note that the expected number of passengers now equals $\bar{w}_p$. Finally, the sixth case considers all trains between all stations in the picture. To overcome the enormous amount of possible OD pairs, we also consider a realization of the passenger figures in this case. Note that Figure 2.1 shows only a part of the railway network in the Netherlands.

The timetable and the passenger figures are obtained from Netherlands Railways. For the first case, only a delay for the train from Zwolle to Amersfoort is interesting, because otherwise no connections are violated. This delay can take values between 0 and 30 minutes. For all other cases, we have generated 100 delay scenarios. In each scenario, each arrival event has a probability of 10% to be delayed. If a train is delayed, the size of this delay is a uniformly distributed integer number between 1 and 15 minutes.

In Table 2.1, we present some information about the cases and sizes of the resulting event-activity networks. The second column gives the number of OD pairs that are included. Recall that an OD pair $p \in \mathcal{P}$ is characterized by its origin and destination station and start time. If it is possible to travel from one station to another at several times, these possibilities correspond to multiple OD pairs. To each OD pair $p \in \mathcal{P}$, we associate a passenger figure w_p. The third column gives the total number of passengers that are considered in each case. The number of passengers is scaled for secrecy and does not represent the true number of passengers that travel in the given time period. For both the number of OD pairs and number of passengers, we have reported the total number and the percentage that need a transfer. The fourth column indicates the number of trains. If a train runs from station A to B and continues from station B to station C, and so on, these trips are counted as one train. Finally, the last columns present the dimensions of the corresponding event-activity networks and the number of binary variables in the resulting integer program.

Comparing Cases II and IV, we see that the number of trains is doubled, while the number of changing activities in the event-activity network is 10 times as large. We also observe that there are far more departure and arrival events, as the regional trains stop more often

Case	OD pairs	Passengers	Trains	Size of the event-activity network $\lvert\mathcal{E}_{\text{dep}}\rvert$	$\lvert\mathcal{A}_{\text{op}}\rvert$	$\lvert\mathcal{A}_{\text{ch}}\rvert$	$\lvert\mathcal{A}_{\text{org}}\rvert$	$\lvert\mathcal{A}_{\text{dest}}\rvert$	Binary Variables
I	111 (15%)	23 (2.3%)	7	28	49	10	131	135	507
II	355 (36%)	147 (12%)	117	219	321	1074	662	705	6219
III	914 (55%)	345 (21%)	168	349	530	1723	1842	2041	21255
IV	3940 (65%)	261 (12%)	284	1022	1760	8068	13832	14665	461494
V	908 (28%)	289 (12%)	284	1022	1760	8068	2342	2548	46533
VI	2875 (39%)	775 (17%)	404	2053	3702	13812	7732	8110	147775

Table 2.1: The cases investigated: For each case, the number of OD pairs, number of passengers, and number of trains are presented. Furthermore, the size of the event-activity network that models each case is given. Note that $|\mathcal{E}_{\text{arr}}| = |\mathcal{E}_{\text{dep}}|$ and that $\mathcal{A}_{\text{op}} = \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}}$.

Case	Objective Value	Delayed events	Dropped connections	Delayed OD pairs	Delayed passengers
I	30462	2.7	1.9	11.4 %	9.0 %
II	169464	41.7	18.1	22.5 %	16.0 %
III	514325	65.4	25.3	24.7 %	21.4%
IV	527543	339.6	192.6	45.7 %	27.5%
V	5584	390.7	209.2	36.9 %	26.4 %
VI	18428	842.2	358.8	42.7 %	33.1 %

Table 2.2: DMwRR: For each case, the objective value and the average number of delays, dropped connections, and delayed trips and passengers are given.

than the long-distance trains. Similar effects can be observed when cases III and VI are compared. Regarding the passengers' data, we see that 65% of the OD pairs in Case IV need a transfer, but that the percentage of passengers who transfer is only about 12%.

2.5.2 Computational results

We used CPLEX 11.1 on an Intel Xeon Quad PC (3.0 GHz) with 3 GB of memory to solve the mathematical models presented in Section 2.3. Table 2.2 reports the characteristics of the optimal solutions for all cases. Each entry in the table is the average value over all delay scenarios. The second column gives the average objective value. Then, the number of events with a delay and the number of dropped connections are reported. The next columns give the percentage of OD pairs and percentage of passengers who have a delay.

Case	II	III	IV	V	VI
$t_p \geq SP_p$	0.3	2.0	741	28.8	161
$t_p \geq f_p$	0.1	0.8	606	22.2	150

Table 2.3: The average running times for different lower bounds on the arrival times t_p. The running times include the time needed to obtain the bound.

The first case can be solved within one second. It turns out that the optimal routes for the passengers are very sensitive to the delays of the trains. At Amersfoort, the intercity to Amsterdam waits at most one minute for the delayed train from Zwolle. If the train from Zwolle arrives later, passengers should travel via Utrecht and transfer there. However, if the connection in Utrecht cannot be made, the passengers should transfer in Amersfoort to the regional train in the direction of Amsterdam.

The arrival times of the passengers for an OD pair $p \in \mathcal{P}$ are bounded from below by SP_p, as passengers cannot arrive earlier than planned. As explained in Lemma 2.5, the polynomial algorithm for one OD pair $p \in \mathcal{P}$ can be used to find a better lower bound on the arrival time t_p. For all cases except the first one, we have evaluated the effect on the computation time of adding the inequality $t_p \geq f_p$ for every $p \in \mathcal{P}$ to the integer program. As can be seen in Table 2.3, the average running time for all cases is reduced. For the largest case, the improvement is about 20%. We conclude that applying the polynomial algorithm to obtain better bounds on the arrival times improves the solution process.

The second and third cases can be solved within one second. With the bound from the polynomial algorithm, solving the fourth case takes, on average, 10 minutes. For the real-time setting that we consider, such computation times are acceptable. If we consider a realization of the passenger demand instead of the averages over several days, computation time are less than one minute. The last case can then be solved within three minutes. These computation times are fast enough for practical application.

2.5.3 The impact of passenger rerouting

To evaluate the effect of taking rerouting into account explicitly, we have compared the performance of our model to a no-wait (NW) policy and to the classical model without rerouting from Schöbel (2007). In a NW policy, all trains depart as early as possible. The timetable is then determined by the operational constraints only. The classical model assumes that a passenger who misses a connection will have a delay of one cycle time. We denote this model by DM. For both NW and DM, we have implemented the model only to decide which connections to maintain. Given these wait-depart decisions, the passengers

	Objective Value						Dropped connections		
Case	NW	DM	DMwRR	NW	DM	DMwRR	NW	DM	DMwRR
I	31217	31889	30462	100 %	102.2 %	97.6 %	2.2	1.7	1.9
II	177053	167873	165110	100 %	94.8 %	93.3 %	26.1	22.4	18.8
III	540393	510427	495111	100 %	94.5 %	91.6 %	39.4	30.2	26.7
IV	558667	544978	512885	100 %	97.5 %	91.8 %	244.7	219.4	192.6
V	6068	5902	5568	100 %	97.3 %	91.7 %	244.7	242.5	209.3
VI	20735	19950	18343	100 %	96.2 %	88.5 %	472.4	415.0	359.4

Table 2.4: Comparison of different models. Model NW implements a no-wait policy. Model DM is the classical DM model. Model DMwRR is our model, which includes rerouting. In all three models, passengers are allowed to take the best-possible path (i.e., to reroute) after the wait-depart decisions have been made.

are then rerouted to compute their actual delay. In this way, a fair comparison can be made between the policies.

As we consider a time period in the late evening, some passengers miss their connection to the last train if rerouting is not included in the optimization of the wait-depart decisions. It is impossible for these passengers to arrive at their destination. Our model, that takes rerouting into account during the optimization of the wait-depart decisions, finds a route for all passengers explicitly. This ensures that all passengers can arrive at their destination, which is a clear advantage of taking rerouting into account. It is hard to assign a specific delay to the passengers for which no route is found. To be able to compare the delay in the three models, we therefore excluded these passengers from our data. To do so, we first evaluated the NW policy. Then, we determined which passengers could not arrive at their destination and removed them from our input.

Table 2.4 reports the characteristics of the solutions for the cases without the excluded passengers. For each case, we report the average objective value and number of missed connections.

For the first case, even the NW policy performs better than the classical model. Note that this case is specifically selected to explain why rerouting should be taken into account, so it could be expected that the classical model without explicit rerouting does not perform well on this case. The model with rerouting gives the best solution. On average, over all scenarios for the first case, the delay is reduced by 4.5% with respect to the classical model and by 2.4% with respect to a NW policy.

For all other cases, the NW policy performs worst. In the second and third cases, taking rerouting into account reduces the delay by about 8% with respect to a NW policy and by about 2% in comparison to the classical DM model. For the larger cases that include the regional trains, we find even larger improvements. The delay is reduced by about 7% with respect to the classical DM model, and by 10% in comparison to a NW policy. The better performance is a consequence of considering also the regional trains. The regional trains give many passengers an alternative route when a connection is missed. This explains why rerouting passengers gives larger improvements for these cases.

2.6 Conclusion and Further Research

In this chapter, we introduced a model that allows to react to delayed trains not only by wait-depart decisions for the following trains but also by rerouting of passengers. For this purpose, we introduced the origin and destination of the passengers as events in the event-activity network used in DM and connected the wait-depart decisions to a shortest-path problem in the resulting network. We proved that this problem is NP-hard. Furthermore, we developed an integer programming formulation for the DM problem with rerouting. This novel formulation was tested on real-world instances of Netherlands Railways. We showed that improvements of 2%-8% can be obtained by incorporating the rerouting aspect in the model. Furthermore, our model ensures that all passengers can reach their destination.

We propose two directions of further research on DMwRR. First, further special cases of the problem should be considered. For these special cases, faster solution procedures can be developed. For example, if the event-activity network has a special structure, this structure can be exploited to solve the DM problem more efficiently. The methods to solve these easier problems can be used in heuristics to solve large-scale DM problems in a short time. Second, decomposing methods could be developed that split the problem in the wait-depart decisions on one hand and the rerouting of the passengers on the other. Heuristics of this type are studied in Chapter 3.

In practice, the limited capacity of the infrastructure has a large impact on the real-time performance of a railway operator. Therefore, capacity constraints should also be integrated in the DM model with rerouting that we presented in this chapter. The capacity of the stations is considered in Chapters 4 and 5.

Chapter 3

Fast Heuristics for Delay Management with Passenger Rerouting

3.1 Introduction

Most regular train passengers will recognize the frustration of missing a connecting train when their feeder train arrives at the transfer station with a small delay. In low-frequency railway systems, missing a connection can have a severe impact on the travel time of the passengers, even if the delay of the incoming train is only small. In such cases, it might be better to delay the connecting train slightly. By delaying the connecting train, passengers from the delayed train are able to transfer to the connecting train and do not have to wait for the next train. A train waiting for passengers from a delayed feeder train reduces the punctuality, which is the main performance indicator for most railway operators. However, it improves the reliability of the system as a whole and thereby increases passenger satisfaction. Netherlands Railways, the largest passenger operator in the Netherlands, endorses the importance of the reliability of the railway system and has recently introduced the *passenger punctuality* as a performance indicator. The passenger punctuality measures the ratio of passengers who arrive at their destination with a delay smaller than a certain threshold value.
In railway operations management, determining which connections to maintain in case of a delayed feeder train is the subject of *delay management*. Delay management thus decides which trains should wait for a delayed feeder train and which trains should depart on time. The aim is to minimize the total delay for the passengers. Deciding on the wait-depart

decisions is a complex problem: If a train waits for a delayed feeder train, passengers in that train will arrive with a delay at the next station, where subsequent transfers take place. This shows that the effects of the wait-depart decisions propagate throughout the entire railway network. Therefore, when solving the delay management problem, the entire railway network should be considered at once. Nevertheless, the current practice at most railway operators is to apply simple rules of thumb to determine which trains should wait. As an example, Netherlands Railways applies a so-called Waiting Time Rule. For each connection, a threshold is determined. If the delay of the incoming train is smaller than the threshold, the connecting train waits. Otherwise, the train departs on time. Kliewer and Suhl (2011) evaluate a wide range of such simple dispatching rules.

In this thesis, we consider *off-line* delay management. In the off-line delay management problem, all primary delays in the system are known before the optimization process starts. Off-line delay management is useful when delays can be predicted, for example when construction works restrict the maximal speed of the trains. Furthermore, when a set of primary delays is known in the system, the secondary delays can be determined easily. Off-line delay management can then propose how to react to these secondary delays. Finally, solution methods for off-line delay management can be applied to solve on-line problems in a real-time setting. This application requires solution methods that solve the off-line delay management within a short computation time. The delay management problem that we consider is: Given a set of source delays in a railway system, determine a *disposition* timetable with a new set of maintained connections, such that the total delay of the passengers is minimized.

A crucial aspect in delay management is to determine the delay for the passengers. To evaluate the delay for passengers who miss a connection, one has to determine how passengers react if a connecting train is missed. Early delay management models assume that such passengers wait for the next train on the same line (Schöbel, 2001). The delay can then be approximated by the cycle time of the timetable. This approximation is correct if the never-meet-property holds (Schöbel, 2007). However, in dense railway networks such as the Dutch, this property is mostly not respected. Gatto et al. (2005) show that the general delay management problem is NP-hard, even under the assumption that passengers wait a complete cycle time. To overcome the difficulty of computing the delay in the case of missed connections, Ginkel and Schöbel (2007) consider the delay management problem as a bi-criteria problem, and optimize the delay of the trains and the number of missed connections simultaneously.

The delay management models mentioned so far ignore the limited infrastructure capacity of the railway system: The limited number of tracks and platforms are not taken into account. In order to obtain solutions to the delay management problem that are feasible in practice, one should consider the infrastructure capacity explicitly. A first approach to take the limited capacity of the tracks into account is presented in Schöbel (2009). Computational tests and heuristics are described in Schachtebeck and Schöbel (2010). In Chapter 4, the limited capacity of the stations is taken into account. Another line of research originates from the conflict detection and resolution literature (D'Ariano et al., 2008), which takes a microscopic view of the railway network to find conflict-free disposition timetables. In Corman et al. (2010b), this microscopic view is applied to minimize the train delay and number of missed connection simultaneously.

Another extension to the classical delay management problem concerns the computation of the delay for passengers who miss a connection. In Chapter 2, we compute this delay more realistically: It is assumed that passengers who miss a connection will take the fastest alternative route to their destinations. To do so, we included the routing decisions of the passengers in the integer programming formulation from Schöbel (2007). In this way, a route is determined for all passengers explicitly. In an experimental study, it is shown that determining the routing decisions already during the optimization of the wait-depart decisions reduces the delay for the passengers significantly. However, for large-scale real-world instances, the integer programs become too large to be solved by standard integer programming techniques.

In this chapter, we propose several heuristic solution approaches for the delay management problem that incorporate the routes of the passengers. Our aim is to find solution methods that balance the computation time on one hand and the quality of the solution on the other hand. We will compare the heuristic solutions to the ones obtained by the exact algorithm for the delay management problem with passenger rerouting and to solutions that are obtained by simple dispatching rules. We will show that a solution can be found for large-scale real-world instances in a reasonable amount of time without compromising the solution quality too much.

The remainder of this chapter is organized as follows. In Section 3.2, we describe the delay management model and review the integer programming formulation for delay management with passenger rerouting. Section 3.3 introduces the heuristic solution methods that we consider. In Section 3.4 we describe our experimental setup and compare the various solution methods. Finally, in Section 3.5 we draw some conclusions and discuss possibilities for further research.

3.2 Delay Management Model

We will now first describe our delay management model formally and then present an integer programming formulation. The integer programming formulation will be used as a benchmark for our heuristic methods and will serve as the basis for some of the heuristics.

Most approaches model the off-line delay management problem with an event-activity network $\mathcal{N} = (\mathcal{E}, \mathcal{A})$. In this directed graph, the events are the departures and arrivals of the trains, that need to be scheduled. We denote the set of arrival and departure events as $\mathcal{E}_{\text{dep}}$ and $\mathcal{E}_{\text{arr}}$, respectively. The events are connected by activities, that represent constraints on the times when these events take place. For example, a waiting activity makes sure that the departure of a train from a station takes place after its arrival there. Driving activities ensure that a minimal driving time between the departure of a train from a station and its arrival at the next station is respected. We denote the waiting and driving activities by $\mathcal{A}_{\text{wait}}$ and $\mathcal{A}_{\text{drive}}$, respectively. The transfers from one train to another are represented by activities that can be removed from the network and denoted by $\mathcal{A}_{\text{change}}$. Only transfers that are maintained pose restrictions on the departure times of the connecting trains. For each activity $a \in \mathcal{A} = \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{change}}$, we denote the minimal time required for that activity by L_a.

The passenger data is represented as a set $\mathcal{P}$ of origin-destination pairs (OD-pairs). Each OD-pair $p \in \mathcal{P}$ represents a group of w_p passengers that want to travel from an origin to a destination at a given time. To determine a route for the passengers explicitly, we solve a shortest-path problem in the event-activity network $\mathcal{N}$. In order to do this, we introduce for each OD-pair an artificial source and sink node in the network. These nodes are referred to as origin and destination events, respectively, and denoted by $Org(p)$ and $Dest(p)$. The source and sink nodes are connected to the regular events by origin and destination arcs. The origin arcs connect the source $Org(p)$ for an OD-pair p to all departure events $e \in \mathcal{E}_{\text{dep}}$ that correspond to a departure from the origin station of p after the time when the passengers in p start their journey. $\mathcal{A}_{\text{origin}}$ denotes the set of all origin arcs. Similarly, all arrival events at the destination station of OD-pair p are connected to the sink node. $\mathcal{A}_{\text{destination}}$ denotes the set of destination arcs.

Note that the arcs $a \in \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{change}}$ can be used by any OD-pair p. On the contrary, an origin arc $a = (Org(p), e) \in \mathcal{A}_{\text{origin}}$ can only be used by OD-pair $p \in \mathcal{P}$. To ease the notation later on, we denote the set of activities that can be used by an OD-pair $p \in \mathcal{P}$ by

$$\mathcal{A}(p) = \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{change}} \cup \delta^{\text{out}}(Org(p)) \cup \delta^{\text{in}}(Dest(p)).$$

We assume that the original timetable and a set of delays is given. For each event $e \in \mathcal{E} = \mathcal{E}_{\text{dep}} \cup \mathcal{E}_{\text{arr}}$, the time when event e is planned is denoted by π_e. The delay at event e is denoted by d_e. Note that $d_e = 0$ for events $e \in \mathcal{E}$ that are not delayed.
For each changing activity $a \in \mathcal{A}_{\text{change}}$, we now determine whether the corresponding transfer is maintained. In order to do so, we introduce for each $a \in \mathcal{A}_{\text{change}}$ a binary decision variable

$$z_a = \begin{cases} 1 & \text{if connection } a \text{ is maintained,} \\ 0 & \text{otherwise.} \end{cases}$$

For each event $e \in \mathcal{E}$, we determine a new time x_e when event e takes place. The values x_e together determine the disposition timetable. Furthermore, for each OD-pair $p \in \mathcal{P}$, we determine which trains the passengers in that OD-pair take. This corresponds to determining a path in the event-activity network for each OD-pair. To model this, we introduce for each OD-pair $p \in \mathcal{P}$ and each activity $a \in \mathcal{A}(p)$ a binary decision variable

$$q_{ap} = \begin{cases} 1 & \text{if OD-pair } p \text{ uses activity } a, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, for each OD-pair $p \in \mathcal{P}$, we introduce an auxiliary variable t_p that represents the arrival time for passengers in OD-pair p. The integer programming formulation for delay management with passenger rerouting reads as follows (see Chapter 2 for more details).

$$\min \sum_{p \in \mathcal{P}} w_p t_p \tag{3.1}$$

such that

$$
\begin{aligned}
x_e &\geq \pi_e + d_e, && \forall e \in \mathcal{E}, && (3.2)\\
x_e &\geq x_{e'} + L_a, && \forall a = (e', e) \in \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{drive}}, && (3.3)\\
M(1 - z_a) + x_e &\geq x_{e'} + L_a, && \forall a = (e', e) \in \mathcal{A}_{\text{change}}, && (3.4)\\
q_{ap} &\leq z_a, && \forall p \in \mathcal{P}, a \in \mathcal{A}_{\text{change}}, && (3.5)\\
1 &= \sum_{a \in \delta^{\text{out}}(Org(p))} q_{ap}, && \forall p \in \mathcal{P}, && (3.6)\\
\sum_{a \in \delta^{\text{in}}(e) \cap \mathcal{A}(p)} q_{ap} &= \sum_{a \in \delta^{\text{out}}(e) \cap \mathcal{A}(p)} q_{ap}, && \forall p \in \mathcal{P}, e \in \mathcal{E}, && (3.7)\\
\sum_{a \in \delta^{\text{in}}(Dest(p))} q_{ap} &= 1, && \forall p \in \mathcal{P}, && (3.8)\\
t_p &\geq x_e - M(1 - q_{ap}), && \forall a = (e, Dest(p)) \in \mathcal{A}_{\text{destination}}, && (3.9)\\
x_e &\in \mathbf{N}, && \forall e \in \mathcal{E}, && (3.10)\\
z_a &\in \{0, 1\}, && \forall a \in \mathcal{A}_{\text{change}}, && (3.11)\\
q_{ap} &\in \{0, 1\}, && \forall p \in \mathcal{P}, a \in \mathcal{A}(p) && (3.12)\\
t_p &\in \mathbf{N}, && \forall p \in \mathcal{P}. && (3.13)
\end{aligned}
$$

The objective function (3.1) minimizes the weighted sum of realized arrival times. As the planned arrival times are fixed, this is equivalent to minimizing the weighted sum of delays. Constraints (3.2) make sure that no event e takes place earlier than planned and that the source delays are taken into account. (3.3) propagate delays along waiting and driving activities a, while (3.4) propagate them along *maintained* changing activities a. Constraints (3.5) state that passengers of OD-pair p can only use a changing activity a if the corresponding transfer is maintained. Equations (3.6)-(3.8) formulate a shortest-path problem for each OD-pair p. Finally, Constraints (3.9) compute the arrival times of the passengers. The constant M in (3.4) and (3.9) is a sufficiently large number. In Chapter 2, we propose a finite value for M that is large enough.

3.3 Heuristic solution approaches

In this section we will describe several heuristic methods to solve the delay management problem. Recall from Section 3.1 that the delay management problem asks for wait-depart decisions, a disposition timetable and new routes for the passengers. At this stage we should note that if the wait-depart decisions are given, finding the optimal timetable and routes is trivial. The disposition timetable can be found by the critical path method (Schöbel, 2007), while the routes for the passengers can be found by solving a shortest-path problem in the event-activity network. This implies that every method that determines the wait-depart decisions can be used as a heuristic.

An easy policy to implement in practice is not to consider the connections at all. In that case, delays are propagated through the network only by the driving and waiting activities. As a consequence, all trains depart as early as possible. We call this heuristic a no-wait policy (NW). We will use this heuristic as a benchmark for all the methods that we develop next.

3.3.1 Simple dispatching rules

In the current operations, railway traffic controllers usually apply simple dispatching rules to determine whether or not to maintain a connection. In Kliewer and Suhl (2011), a large set of such dispatching rules is compared. We have implemented two rules to compare our heuristic solutions to. The first one, the Waiting Time Rule (WTR), corresponds to the current practice at Netherlands Railways.

Under a WTR policy, a maximal waiting time is determined for each connection. This maximal waiting time can be differentiated for different types of connections or even for each connection individually. However, we will assume a constant time $d_{\max}$ for all connections for simplicity. Let now a changing activity $a = (e, e') \in \mathcal{A}_{\text{change}}$ be given and assume that the arrival event e is delayed. It is easy to determine the time that event e' has to be delayed in order to maintain the connection. Denoting this delay to maintain connection a by d_a, it holds that

$$d_a = \pi_e + d_e + L_a - \pi_{e'}.$$

A connection a is maintained if $d_a \leq d_{\max}$ and dropped otherwise. We have experimented with different values of $d_{\max}$. Note that $d_{\max} = 0$ corresponds to a no-wait policy.

The drawback of the previous dispatching rule is that it does not take into account the number of transferring passengers. To improve the solutions, the second rule that we consider is the Ratio of Transferring Passengers (RTP). For each connection, first the number of passengers who use a connection is determined. This number is then compared to the number of passengers that have planned to take the connecting train. For a connection $a = (e', e) \in \mathcal{A}_{\text{change}}$, we thus compute the following ratio

$$\rho_a = \frac{\text{Number of passengers that } \textit{planned} \text{ to use connection } a}{\text{Number of passengers that } \textit{planned} \text{ to use driving activity } (e, f)}.$$

Note that for each departure event e, there is exactly one driving activity $(e, f) \in \mathcal{A}_{\text{drive}}$. If the ratio of these numbers is larger than a certain threshold $\rho_{\min}$, *i.e.*, $\rho_a \geq \rho_{\min}$, then connection a is maintained. Again, for values $\rho_{\min}$ larger than 1, we obtain the no-wait policy.

3.3.2 The classical model as a heuristic

The next heuristic method applies the delay management model from Schöbel (2007). Recall from the introduction that this model assumes that passengers wait for a complete cycle time when they miss their connection. As input, the model needs the number of passengers w_e that plan to end their journey at event $e \in \mathcal{E}_{\text{arr}}$ and the number of passengers $a \in w_a$ that use a connection $a \in \mathcal{A}_{\text{change}}$. Given the set of OD-pairs $\mathcal{P}$, these numbers can be computed as

$$w_e = \sum_{p \in \mathcal{P}(e)} w_p, \qquad w_a = \sum_{p \in \mathcal{P}(a)} w_p,$$

where $\mathcal{P}(e)$ denotes the set of OD-pairs that planned to arrive at their destination with event e and $\mathcal{P}(a)$ denote that set of OD-pairs that planned to use a connection $a \in \mathcal{A}_{\text{change}}$. Denoting T for the cycle time of the timetable, the delay for the passengers is approximated by

$$\sum_{e \in \mathcal{E}_{\text{arr}}} \sum_{p \in \mathcal{P}(e)} w_p x_e + \sum_{a \in \mathcal{A}_{\text{change}}} \sum_{p \in \mathcal{P}(a)} w_p T(1 - z_a). \tag{3.14}$$

An integer programming formulation for the classical model is now given by the objective function (3.14) together with the constraints (3.2)-(3.4) and (3.10)-(3.11). Note that passengers who miss a connection are counted twice in the above objective function: Both the train they planned to arrive with is included in the first term, and the transfer that they missed is included in the second term. The model is correct if the so-called never-meet-property holds (Schöbel, 2007), but for dense railway networks as the one in the Netherlands, this property is often not satisfied.

The classical model assumes that passengers who miss a connection have to wait for one cycle time T, and therefore adds a penalty T to the objective function for every connection that is dropped. Instead of assuming that passengers wait for one cycle time, we can also assume that there is an estimate D of the *additional* delay for passengers who miss a connection, and add this estimate as a penalty to the objective function for every missed connection. The objective function then becomes

$$\sum_{e\in\mathcal{E}_{\text{arr}}}\sum_{p\in\mathcal{P}(e)} w_p x_e + \sum_{a\in\mathcal{A}_{\text{change}}}\sum_{p\in\mathcal{P}(a)} w_p D(1-z_a).$$

We now view the estimate D as a parameter and find the best value D^* by enumeration. Note that the classical model gives both the wait-depart decisions and the disposition timetable. Given the disposition timetable, it is easy to determine the fastest path for each OD-pair $p \in \mathcal{P}$ in order to evaluate the total delay.

3.3.3 An iterative heuristic

The previous set of heuristics assumes a fixed estimate D for the additional delay: This delay is equal for all OD-pairs. We will now relax this assumption, and allow for an estimate D_p that differs among the OD-pairs $p \in \mathcal{P}$. Given a set of values D_p, we can solve the classical delay management model with objective function

$$\sum_{e\in\mathcal{E}_{\text{arr}}}\sum_{p\in\mathcal{P}(e)} w_p x_e + \sum_{a\in\mathcal{A}_{\text{change}}}\sum_{p\in\mathcal{P}(a)} w_p D_p(1-z_a).$$

We will find the best value for D_p with an iterative approach. Given the values D_p, we can solve the classical delay management model. With the wait-depart decisions and disposition timetable that we then obtain, we can determine new routes for the passengers. If we find an OD-pair that misses a connection, we can compute the actual delay for this OD-pair, and use this to update the estimate D_p. This process can be repeated until the best values D_p are found. A more formal outline of the iterative algorithm is given below.

1. Initialize: Set $D_p = 0$ for all $p \in \mathcal{P}$.
2. Repeat until convergence or until a maximum number of iterations has been reached:
   (a) Solve the classical delay management model with the current values D_p.
   (b) Compute the fastest routes for the passengers. This gives the additional delay d_p for an OD-pair p that misses a connection.
   (c) Update the values D_p for all passengers who miss a connection:

   $$D_p = d_p - x_e,$$

   where e is the arrival event that OD-pair p planned to arrive with.

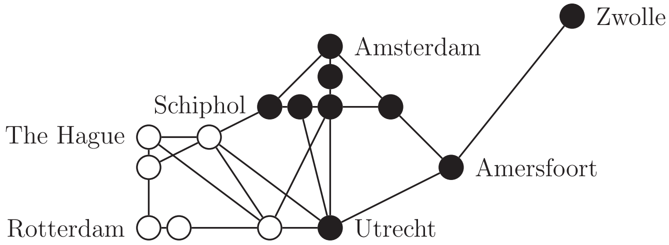

Figure 3.1: The railway network that is considered in the numerical experiments

In the initialization step, all values D_p are set to 0. This means that passengers are not delayed if they miss a connection. In particular, the first solution of the classical delay management model will drop all connections and the timetable that we then find will equal that of a no-wait policy. For the passengers in an OD-pair $p \in \mathcal{P}$ who miss a connection in this no-wait policy, the value D_p will be updated in Step 2c. Recall that an OD-pair p is counted twice in the objective function. D_p should therefore be an estimate of the *additional* delay of OD-pair $p \in \mathcal{P}$. This explains the term x_e in the update of the estimates in the final step. In the subsequent iterations, the classical model will find a balance between this additional delay for OD-pair p and the delay that is obtained when the connection is maintained.

3.4 Numerical experiments

We have performed an experimental study to compare the performance of the heuristics that are described in the previous section. We will now first describe our experimental setup and then discuss the results.

3.4.1 Cases

We obtained detailed passenger data from Netherlands Railways, the largest passenger operator in the Netherlands. We have created six real-world instances of varying size to determine how the solution time and solution quality depend on the size of the networks and on the number of passengers. Five of these instances are also used in Chapter 2[1].
In Figure 3.1, the railway network that we consider is depicted. The graph represents part of the Dutch railway network in the mid-Western part of the country. It contains

[1]Five of these instances were also used in Chapter 2, but given a different number. Below you find between brackets the corresponding case in Chapter 2 I(II), II(IV), III(V), IV(III), V(-), VI(VI).

a node for each transfer station in the network. Two nodes are connected by an edge if there is a train service between the stations. On most links, both a long distance and a regional train service is operated, typically with a frequency of two trains per hour. Note that there can be many smaller stations on a link, that are serviced by the regional trains only.

All cases consider a period of four hours in the evening. In the first three cases, we consider all stations indicated by a black dot in this figure. The first case includes all long distance trains. As passenger weights, we consider for each OD-pair the *average* number of passengers. The second and third case consider both the long distance and the regional trains on this network. In the second case, we again take the average number of passengers as the weights. Recall however that the regional trains stop at all stations in the network. As a consequence, the number of OD-pairs is much larger than in the first case. For many of them, the average number of passengers is relatively small. In order to deal with this enormous amount of OD-pairs, in the third case we consider a possible realization of passenger figures for one day. The realization is constructed in such a way that the expectation of the passenger figures equals the average. We refer to the previous chapter for more details on the process of generating the realizations of the passenger figures.

The remaining three cases consider the entire network in Figure 3.1. The fourth case includes only the long distance trains. The fifth case considers also the regional trains. Both cases use the average passenger figures as weights. Finally, the sixth case considers all trains on the network, but considers a possible realization of the passenger demand.

In Table 3.1, some characteristics of the cases are given. For each case, we list the number of stations, the number of trains, the number of OD-pairs, the number of passengers, and the number of departures and transfers in the event-activity network. The number of passengers has been scaled for secrecy and does not represent the true numer of passengers. In the columns with the number of passengers and the number of OD-pairs, we have indicated in brackets the percentage of passengers that have to transfer during their trip. It can be seen that the percentage of passengers that transfer is much smaller than the percentage of OD-pairs with a transfer. This implies that the passenger weights w_p are much larger for OD-pairs $p \in \mathcal{P}$ that need no transfer.

To evaluate the heuristic methods, we have simulated 100 delay scenarios for each case. The scenarios are constructed as follows. First, each arrival has a probability of 10% to be delayed. Second, if a train is delayed, its delay is a uniformly distributed number between 1 and 15 minutes.

Case	Stations	Trains	OD-pairs	Passengers	Departures	Transfers
I	10	117	355 (36%)	147 (12%)	219	1074
II	34	284	3940 (65%)	261 (12%)	1022	8068
III	34	284	908 (28%)	289 (12%)	1022	8068
IV	16	168	914 (55%)	345 (21%)	349	1723
V	82	404	22256 (81%)	705 (17%)	2053	13812
VI	82	404	2875 (39%)	775 (17%)	2053	13812

Table 3.1: Some characteristics of the instances

$d_{\max}$	I	II	III	IV	V	VI
0	177053	558667	6068	540393	1917679	20736
1	175178	<u>551059</u>	<u>5992</u>	531780	1877530	20293
2	<u>174863</u>	554903	6033	<u>526938</u>	<u>1869495</u>	<u>20206</u>
3	178453	574323	6246	527224	1903082	20578
4	182580	612444	6662	533188	1986597	21501
5	189927	670575	7298	543174	2119217	22969

Table 3.2: The results of the Waiting Time Rule for the first three cases. For each case, the best result is underlined.

3.4.2 Dispatching rules

Our first heuristics apply simple dispatching rules to find the wait-depart decisions. These heuristics maintain a connection if a property of this connection exceeds a certain threshold value. We will now show how to find the best values for these threshold values.

Waiting Time Rule

The Waiting Time Rule (WTR) maintains a connection if the connecting train has to wait at most $d_{\max}$ minutes. To find the best value for the threshold parameter $d_{\max}$, we have evaluated the heuristics for values ranging from 0 to 5 minutes. In Table 3.2, the results are presented. The first column in this table gives the value of the parameter $d_{\max}$. Then, the average objective value is given for each of the cases. Recall that for $d_{\max} = 0$, WTR corresponds to a no-wait policy. For the first three cases, with a value $d_{\max} > 2$ for the threshold, the WTR heuristic performs worse than the no-wait policy. It turns out that allowing all connecting trains to wait for more than 2 minutes for a delayed feeder train gives a bad policy. For Cases IV and for Cases V and VI, WTR performs worse than the no-wait policy if $d_{\max} \geq 5$ and $d_{\max} \geq 4$, respectively. In all cases, the WTR performs better than a no-wait policy for $d_{\max} \in \{1, 2\}$: The delay is reduced on average

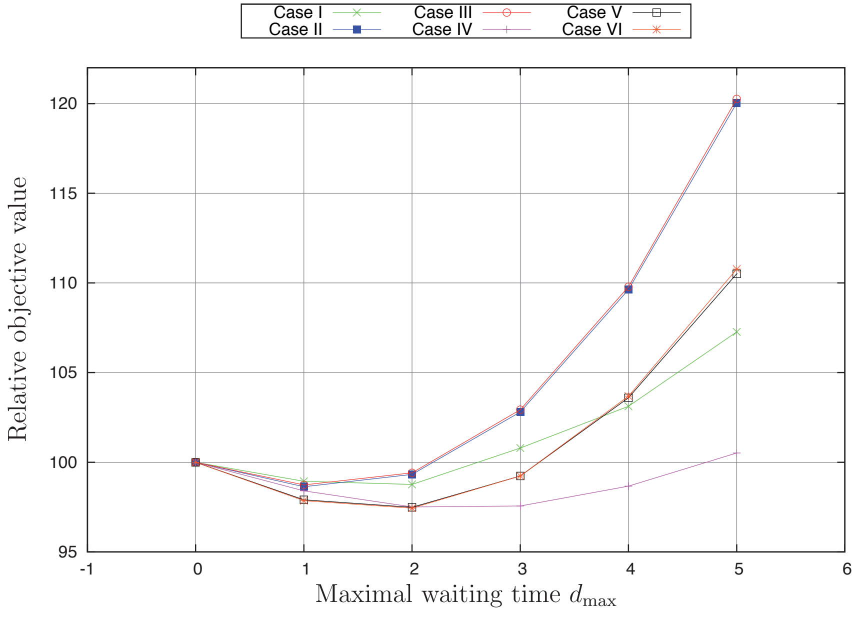

Figure 3.2: The results of the Waiting Time Rule

by 1.9%. For the second and third case, the best results are found for $d_{\max} = 1$. For all other cases, $d_{\max} = 2$ gives the best performance.

For easier comparison of the results among the cases, we have normalized the objective value and plotted these relative objective values in Figure 3.2. The objective value for the no-wait policy is set to 100 percent. For values of $d_{\max} > 0$, we have computed the delay relative to the no-wait policy. One immediately notices that the behavior of the policy is very similar for Cases II and III, and for Cases V and VI. Recall that these cases consider the same railway system, and differ only in the OD-pairs that are considered. It follows from this graph that a small subset of OD-pairs can be used to evaluate the performance of the heuristic for the case that includes all OD-pairs.

Ratio of Transferring Passengers

The Ratio of Transferring Passengers (RTP) heuristic computes the ratio of the number of passengers who plan to use a connection and the number of passengers who plan to use the connecting train. If this ratio exceeds a given threshold $\rho_{\min}$, the connection is maintained; otherwise it is dropped. We experimented with values of $\rho_{\min}$ between 0 and 100 percent. Note that we obtain a no-wait policy if $\rho_{\min} > 100\%$. In order to compare

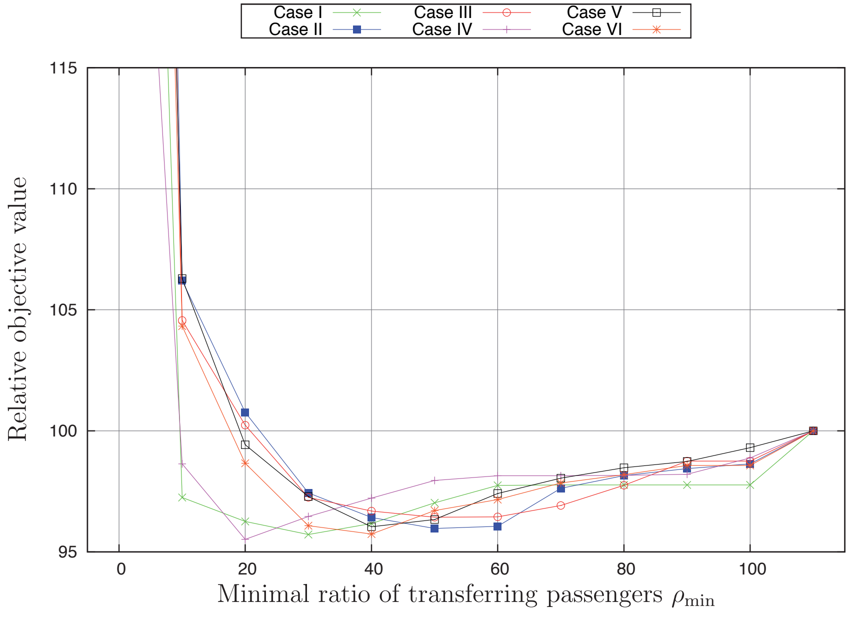

Figure 3.3: Results for the Ratio of Transferring Passengers rule

the results to the no-wait policy, we therefore also applied the heuristic with $\rho_{\min} = 110\%$.

In Figure 3.3, the results of the RTP heuristic are presented for all cases. From the graph we see that with the best value for the parameter $\rho_{\min}$, the RTP dispatching rule performs about 4% better than the no-wait policy. Contrary to the WTR rule, the best value for the parameter $\rho_{\min}$ differs among the cases. For Cases I and IV, that consider only the long distance trains, the best value is found at 30% and 20%, respectively. For the other cases, that include also the regional trains, the best heuristics are found with $\rho_{\min}$ equal to 40% or 50%. We conclude that if both long distance and regional trains are considered, the optimal ratio $\rho_{\min}$ should be higher. In order to explain this, recall that the regional trains stop at all stations along the railway line. Consider a transfer at a large station to a regional train that then travels in the direction of another large stations and stops at some smaller stations along the way. Many passengers will enter the regional train in one of the smaller stations and travel towards the larger station. When we determine whether to maintain the connection at the transfer station or not, these passengers are not considered, as they will enter the train at a later time. However, if the connection is maintained, they will be delayed. The RTP heuristic then underestimates the delay that arises if the connection is maintained. Choosing a higher value for $\rho_{\min}$ makes up for this underestimation.

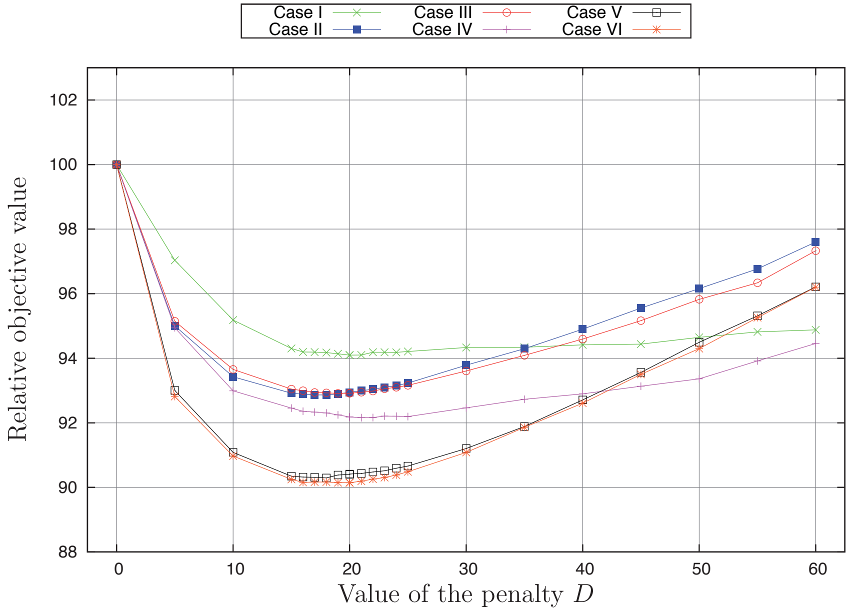

Figure 3.4: The results for the heuristic that applies the classical model

With the optimal value for the parameter, the RTP rule reduces the delay roughly by 4%. Comparing the RTP rule to the WTR rule, we see that the performance of the RTP rule is much better. It thus pays off to compare the number of passengers that want to use a connection to the number of passengers that are delayed if the connecting train waits.

3.4.3 Classical Delay Management

Our second set of heuristics applies the classical delay management model, but views the penalty D in the objective function as a parameter. In the original model, this parameter was set to T, the cycle time of the timetable. If we set $D = 0$, the heuristic equals the no-wait policy. That allows us again to normalize the objective value, in order to be able to compare the results among the cases. We have experimented with values for D ranging from 0 to 60 minutes. Note that $T = 60$ minutes in our timetable.

In Figure 3.4, the results for all cases are presented. To find the best value for the parameter D, we have evaluated the heuristic for more values around $D = 20$. For all cases, we found that $18 \leq D^* \leq 21$. Furthermore, the differences in objective value are very small among these values. We conclude that the additional delay for passengers that miss a connection is about 20 minutes. The performance of the heuristic with the best

	I	II	III	IV	V	VI
Instances that did not convergence	0	0	1	0	1	0
Average number of iterations	2.23	3.48	3.09	2.78	3.90	4.24

Table 3.3: The number of instances for which the iterative heuristic did not converge and the average number of iterations

value for the parameter D is much better for the last three cases. For the smaller network, the quality of the solutions is 6% better than a no-wait policy. For the larger network, the objective values are reduced by 9%.
Considering the shape of the graph for each case individually, one sees that the graph for Cases I and IV is very flat at the end. If the value of D is increased, this does not give objective values that are much worse. On the contrary, for the other cases the objective value increases if the value of the parameter is increased. Furthermore, we again see that the results are very similar for Cases II and III, and for Cases V and VI. This suggests that one can use a small set of OD-pairs to find the best value of the parameter D.

3.4.4 Iterative heuristic

Our final heuristics applies the classical delay management model in an iterative fashion. Contrary to the previous heuristic, it uses a parameter D_p that differs among the OD-pairs $p \in \mathcal{P}$. In each iteration, the heuristic first solves the classical model and then reroutes the passengers. It stops when the method converges or when the maximal number of iterations has been reached. In Table 3.3, we present the number of iterations that the heuristic needs. The first row gives the number of instances that did not converge. The second row gives the average number of iterations among the instances that did converge. We see in the table that the heuristic converges for almost all instances. Only in Cases III and V, there is one instance that does not converge. For these instances, the heuristic cycles between two solutions, that have almost identical objective values. In both instances, there is one OD-pair that is rerouted in one solution, and takes the planned route in the other. On the other instances, the iterative heuristic converges on average in less than five iterations.

	I	II	III	IV	V	VI
No-wait	177053	558667	6068	540393	1917679	20736
WTR	174863	551059	5992	526938	1869495	20206
RTP	169477	536135	5852	516173	1841761	19853
Classical	166601	518809	5638	498031	1731831	18692
Iterative	165610	515573	5600	496529	1720748	18534
Exact	165110	512878	5567	495111	-	18343
No-wait	100.0	100.0	100.0	100.0	100.0	100.0
WTR	98.8	98.6	98.7	97.5	97.5	97.4
RTP	95.7	96.0	96.4	95.5	96.0	95.7
Classical	94.1	92.9	92.9	92.2	90.3	90.1
Iterative	93.5	92.3	92.3	91.9	89.7	89.4
Exact	93.3	91.8	91.7	91.6	-	88.5

Table 3.4: The absolute and relative objective value for each heuristic

3.4.5 Comparison of the heuristics

In the previous sections we have shown how to find the best parameters for the simple dispatching rules and for the heuristic based on the classical model. We will now compare both the quality and the running time of the heuristics to each other and to the exact approach. In Table 3.4, the best objective value is presented for each heuristic and for each case. The upper rows contain the absolute objective values, the lower ones contain relative objective values. Again, we used the no-wait policy to normalize the objective values.

We see in the table that the simple dispatching rules give the worst results. Although they are easy to implement, the quality of the solutions they produce is bad. Using the classical model as a heuristic, by viewing the penalty D as a parameter, reduces the delay on average by 8%. The iterative heuristic improves slightly over the classical model, reducing the delay by an additional 0.5%. For the first five cases, the problem could also be solved by the exact algorithm. Comparing the objective values obtained with the iterative and exact algorithm for those cases, we see that the iterative heuristic finds solutions that are close to optimal. The relative deviation is on average only 0.4%.

Our aim with the off-line delay management heuristics is to apply them in an algorithm for the on-line delay management problem. As the on-line delay management problem should be solved in a short computation time, we are also interested in the running times for the different heuristics. For the heuristics that apply simple dispatching rules, the running times are neglectable. For each case, the running time was so short that it could not reliably be measured. We thus compare only the running times for the classical and

	I	II	III	IV	V	VI
Classical	0.04	0.5	0.25	0.08	3	0.5
Iterative	0.05	1	0.3	0.11	6.1	1
Exact	0.48	357	10	1.4	-	62

Table 3.5: The average running times for the classical, iterative and exact algorithm, in seconds

iterative heuristic and for the exact algorithm. In Table 3.5, these average running times are reported.

The running times for the classical and iterative model are very short. Even the largest case can be solved within seconds. For the larger cases, the classical heuristic is about two times faster than the iterative heuristic. The exact algorithm needs much more computation time. It takes on average almost six minutes to solve Case III. For the real-time setting of delay management, such computation times are too long. The running times of the classical and iterative heuristic allow a real-time application.

3.5 Conclusions

To evaluate the quality of any delay management policy, the routes that passengers choose have to be taken into account. This rerouting aspect of delay management should be considered when solution methods for the delay management problem are developed. In this chapter, we have constructed several heuristic methods to solve the off-line delay management problem. We have compared these heuristics among each other and to the exact algorithm in an experimental study. In this study, it is shown that an iterative heuristic, that solves the classical delay management model iteratively, leads to good solutions. On average, the performance of the iterative algorithm is only 0.4% worse than the exact algorithm. This decrease in quality is compensated by the running time of the method; the iterative approach could solve all instances within seconds. We have also implemented simple dispatching rules, that are currently used in practice. For example, the Waiting Time Rule is applied at Netherlands Railways. In our numerical experiments, it is shown that these dispatching rules do not perform well.

Our delay management model ignores the limited capacity of the railway infrastructure. To obtain solutions that can be implemented in practice, these capacity considerations should be incorporated in the delay management models with passenger rerouting. An exact algorithm for delay management models that includes both passenger rerouting and capacity constraints might be too ambitious. However, any algorithm for the capacitated delay management problem without rerouting can be applied in an iterative fashion to obtain a heuristic for the capacitated delay management problem with passenger rerouting. Our results on the uncapacitated delay management problem suggest that the total delay for the passengers can then be reduced significantly. In the next chapters, we develop several delay management models that incorporate the limited capacity of the infrastructure inside the stations.

Chapter 4

Delay Management including Capacities of Stations

4.1 Introduction and Motivation

Since the first integer programming formulation for delay management in 2001 (Schöbel, 2001), there has been a significant amount of research on extensions of the basic delay management problem. Delay management deals with the question whether a train should better wait for a delayed feeder train or depart on time (*wait-depart decisions*). The goal is to minimize total (weighted) passenger delay. Because missing a connection is an enormous frustration for railway passengers, delay management has also received much attention from railway companies. Unfortunately, the models so far do not include station capacities, which is a crucial aspect in practice, because station capacities are often limited. The topic of this chapter is delay management with station capacities.
In most European countries, railway transport plays an important role. Many people travel by train for distances between 20 and 800 kilometers, especially during peak hours when there are many traffic jams on the highways. Passengers prefer a direct connection, however it is impossible that there is a direct connection between all possible origin-destination pairs. The Dutch line plan is constructed in such a way that most passengers (about 75%) have a direct connection. In addition, the most important transfers are cross-platform and have a short connection time. A typical example is station Amersfoort, where trains from the North and East arrive at the same time, and continue towards the directions of Utrecht and Schiphol Airport. Passengers in the train from Zwolle towards Utrecht (see Figure 4.1) that have Amsterdam Central Station as their final destination,

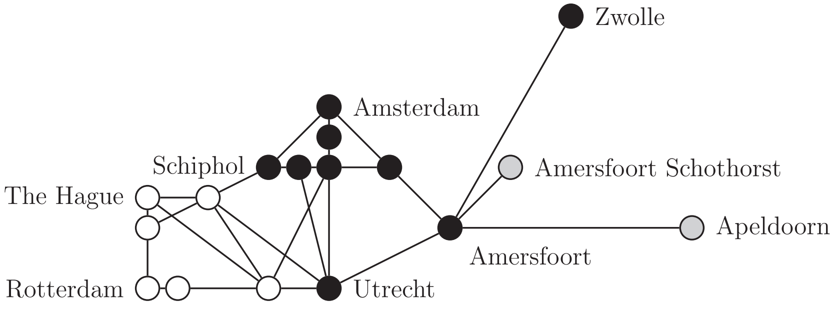

Figure 4.1: Part of the railway network in the Netherlands.

Figure 4.2: A schematic representation of the assignment of trains to platform tracks at Amersfoort. The left track is depicted below the platform, the right track above it.

can change on the same platform, where their train to Amsterdam departs a few minutes later.

We use this example to illustrate why station capacities should be taken into account in a delay management model. In the regular timetable the trains from Zwolle to Utrecht and from Apeldoorn to Schiphol Airport depart and arrive on the left and right side of the same platform in Amersfoort (see Figure 4.2) at minute .22 and .24, respectively. The train from Amersfoort Schothorst towards Amsterdam arrives on the left side of this platform at minute .28 and leaves at .29. Now suppose that the train from Zwolle has a delay of 10 minutes and that there are many transferring passengers towards Amsterdam.

- The optimal solution of a basic delay management model suggests that the train towards Amsterdam waits for the transferring passengers from Zwolle. In addition, it assumes that this train arrives on time in Amersfoort. However, if this train would arrive on time and waits for the connecting train from Zwolle, this feeder train could never arrive, because the platform track is blocked.

- A possible strategy to obtain a feasible solution is to use the delay management model only to obtain the wait-depart decisions, but take the regular order of the trains. In that case, the train from Zwolle arrives at .32 with 10 minutes delay and then departs at .34. Then, after this train has departed, the train towards Amsterdam arrives at the same platform track. As a minimum headway time of 3 minutes is required between two trains using the same platform track, this means that this train will never arrive before minute .37. As a consequence, the train to Amsterdam will depart with a large delay.

- However, the right side of the platform is already empty for some time. If the train towards Amsterdam is rescheduled to this platform track, it can wait for the transferring passengers there. As the train from Zwolle arrives at .32 and two minutes of minimal transfer time are required, the train towards Amsterdam can leave at .34. This solution gives the minimum possible delay for the situation.

Of course, the real-time rescheduling of the platform assignment contains also some disadvantages. It requires additional work for dispatchers of the traffic control centers, and it is annoying for the passengers, especially, if they have to move to another platform.

From this example, we can draw the following conclusions.

- The optimal solution value of basic delay management models provides a lower bound on the optimal solution value of delay management with station capacities. However, this solution can be infeasible in practice.

- Fixing the wait-depart decisions of an optimal delay management solution and fixing the order of the trains leads to feasible solutions. These solutions are typically of low quality: Passengers face very large delays in these solutions.

- When re-assigning platform tracks is allowed, this may result in less passenger delays.

In this chapter, we incorporate station capacities in the delay management model. We compare solutions with a fixed platform assignment to solutions in which we can re-assign a platform track during the operations. The contributions of this chapter are as follows. Firstly, we present a new integer programming formulation for the delay management problem taking into account station capacity constraints. Secondly, we develop an iterative approach to solve this problem heuristically. Thirdly, we compare the optimal solution of the new model and the solution of the iterative heuristic with methods based on the traditional delay management model. Finally, we investigate the effect of flexibility in the platform assignment on the total passenger delay in several real-world problem instances of Netherlands Railways. Based on our findings, railway companies can make a trade-off between changing platform tracks at the last moment versus the total passenger delay.

The remainder of the chapter is structured as follows. Section 4.2 reviews the relevant literature. In Section 4.3 we present an integer programming formulation for the delay management model with station capacities. In Section 4.4 we discuss an iterative approach to solve this model. Computational results are discussed in Section 4.5. In Section 4.6, we discuss the balance between the passenger delay on the one hand and the number of platform track changes on the other. Finally, we finish the chapter with some concluding remarks and suggestions for further research in Section 4.7.

4.2 Literature Review

There exist various models and solution approaches for delay management. The main question, which has been treated in the literature so far, is to decide which trains should wait for delayed feeder trains and which trains better depart on time. A first integer programming formulation for this problem has been given in Schöbel (2001) and has been further developed by De Giovanni et al. (2008) and Schöbel (2007); see also Schöbel (2006) for an overview about various models. The complexity of the problem has been investigated in Gatto et al. (2005). An online version of the problem has been studied by Gatto et al. (2007), Gatto (2007), Kliewer and Suhl (2011), and Krumke et al. (2011). Berger et al. (2011) show that it is PSPACE-hard. All models mentioned so far assume that passengers have to wait a complete cycle time in case they miss their connection. In order to compute the delay for the passengers more accurately, we reroute passengers that have missed a connection in Chapter 2. In order to solve large-case real-world instances, several heuristics for this delay management model with passenger rerouting are presented in 3.

In railway transportation, an important issue concerns the limited capacity of the track system. Schöbel (2009) presents a first model for delay management that includes capacity constraints. Schachtebeck and Schöbel (2010) and Schachtebeck (2010) give an integer programming formulation and propose heuristic methods for the capacitated delay management problem. The idea is to add *headway constraints*, which make sure that there is enough distance between two train departures and hence prevent two trains from using the same piece of track at the same time. Using machine scheduling models, it turns out that the model with headway constraints is NP-hard even in the case that no wait-depart decisions have to be made, see Conte and Schöbel (2007).

Another line of research dealing with railway operations is based on the alternative graph formulation (Mascis and Pacciarelli, 2002), originally used to model job shop variants. A branch-and-bound algorithm for finding a conflict-free train schedule, minimizing the largest delay, is developed in D'Ariano et al. (2007) and Caimi et al. (2011). In Corman et al. (2010a), the authors suggest a tabu search to solve both the train sequencing and train routing problem, where a set of possible routes is given as input. The alternative graph formulation has been used in Corman et al. (2010b) in the context of delay management, in a bi-objective approach that optimizes the maximal delay and the number of missed connections simultaneously. A similar bi-objective approach is presented by Ginkel and Schöbel (2007) for the macroscopic model. Here, the total train delay and the number of missed connections are minimized.

In Chapter 5, we make a first step towards a complete integration of delay management and train scheduling. We will propose an optimization framework that iteratively solves first a delay management model and then a train scheduling model.

The problem of taking station capacities into account is also relevant in timetabling. Here, one has to check for a given timetable whether the capacity in every station is sufficient. Instead of considering the number of platforms as the capacities of the stations, it is even more realistic to look at the *train routing problem*, *i.e.*, to find routes through the stations for all the trains using the detailed track topology. This feasibility problem has been extensively studied. In Kroon et al. (1997), a set of inbound and outbound routes is given for each train. If a train chooses one of these routes, all track sections of it are reserved at once but released section-wise. It is shown that deciding whether a feasible schedule exists is NP-complete already for three possible routes per train. In Caprara et al. (2011), the problem is modeled as an integer program using clique inequalities in a conflict graph. For a recent survey on railway track allocation problems, see Lusby et al. (2011).

4.3 Integer Programming Formulations

In this section we present an integer programming formulation for the delay management problem that takes the capacities within stations into account. As basis for this model we use the integer programming formulation that includes capacities of the tracks as it was introduced in Schachtebeck and Schöbel (2010). Note that other formulations of the DM problem can analogously be extended to take the stations' capacities into account. We now first describe the integer programming formulation without station capacities and then show how to incorporate the limited capacity of the station infrastructure.

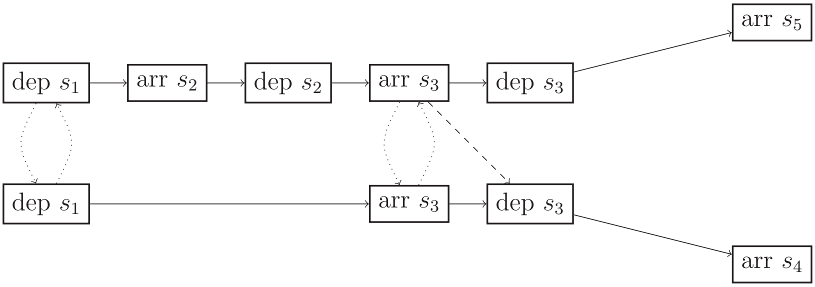

Figure 4.3: An event-activity network with a transfer activity and headway activities

4.3.1 Formulation without station capacities

For modeling delay management problems as integer programs, usually an *event-activity network* $\mathcal{N} = (\mathcal{E}, \mathcal{A})$ is used as underlying directed graph. The set of nodes $\mathcal{E}$ corresponds to the arrival and departure events of all trains at all stations. In Figure 4.3, an event-activity network is depicted for two trains. The lower train is a long-distance train traveling from s_1 to s_4 via s_3. The upper train is a regional train that also travels from s_1 to s_3, but stops at station s_2 also. After s_3, it continues in the direction of s_5. The events are represented as rectangles in the picture. The set $\mathcal{A}$ consists of the following activities. Between the arrival i and the departure j of a train in the same station, there is a *waiting activity* $a = (i, j) \in \mathcal{A}_{\text{wait}}$; between a departure i of a train in a station and its arrival j in the next station there is a *driving activity* $a = (i, j) \in \mathcal{A}_{\text{drive}}$. The set $\mathcal{A}$ furthermore contains *transfer activities* $\mathcal{A}_{\text{change}}$ linking an arrival of a train in a station to a departure of another train in the same station. In Figure 4.3, there is a transfer at s_3 from the regional to the long-distance train, which is depicted by a dashed arrow. Finally, *headway activities* are needed for any pair of trains competing for the same infrastructure. These headway activities represent pairs of precedence relations, one of which must be selected. To illustrate this, let i and j be two departures that continue over a common track. We denote the corresponding arrivals at the next common station by i' and j', respectively. In Figure 4.3, i' and j' are the arrivals of the trains at s_3. If departure i takes place before departure j, then arrival i' should also be scheduled before arrival j'. This choice for the order of the trains is represented by a pair of headway activities $a_1 = (i, j), a_2 = (j, i) \in \mathcal{A}_{\text{head}}$. For each pair of headway activities, we define both a pair of constraints for the departures of the trains and a pair for the arrivals of the trains. In our example, these two pairs of constraints corresponding to *one* pair of headway activities are shown with dotted arcs. Each activity $a \in \mathcal{A}$ requires a minimal duration that is denoted by L_a.

The most important decision in delay management is which connections need to be maintained. For each changing activity $a \in \mathcal{A}_{\text{change}}$ we thus introduce a binary decision variable z_a, which is defined as follows.

$$z_a = \begin{cases} 0 & \text{if connection } a \text{ is maintained,} \\ 1 & \text{otherwise.} \end{cases}$$

In order to take the capacity constraints on the tracks into account, one defines a binary decision variable g_{ij} for each headway activity $(i, j) \in \mathcal{A}_{\text{head}}$, given as

$$g_{ij} = \begin{cases} 0 & \text{if event } i \text{ takes place before event } j, \\ 1 & \text{otherwise.} \end{cases}$$

For each event $i \in \mathcal{E}_{\text{arr}} \cup \mathcal{E}_{\text{dep}}$, we define $x_i \in \mathbb{N}$ as the rescheduled time when event i takes place. The set of variables $x = (x_i)$ defines the disposition timetable. If the wait-depart decisions z_a and the priority decisions g_{ij} are fixed, the values of x_i, $i \in \mathcal{E}$ can easily be calculated by the critical path method (see Schöbel (2006)).
Given the original timetable π_i, $i \in \mathcal{E}$ and a set of exogenous source delays d_i at events and d_a at activities (being zero if there is no delay), the integer programming formulation (DM) without station capacities reads as follows.

$$(\mathbf{DM}) \qquad \min f(x, z, g) = \sum_{i \in \mathcal{E}_{\text{arr}}} c_i(x_i - \pi_i) + \sum_{a \in \mathcal{A}_{\text{change}}} z_a c_a T \tag{4.1}$$

such that

$$\begin{aligned}
x_i &\geq \pi_i + d_i & &\forall i \in \mathcal{E}, & (4.2)\\
x_j - x_i &\geq L_a + d_a & &\forall a = (i,j) \in \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{drive}}, & (4.3)\\
M z_a + x_j - x_i &\geq L_a & &\forall a = (i,j) \in \mathcal{A}_{\text{transfer}}, & (4.4)\\
M g_{ij} + x_j - x_i &\geq L_a & &\forall a = (i,j) \in \mathcal{A}_{\text{head}}, & (4.5)\\
M g_{ij} + x_{j'} - x_{i'} &\geq L_a & &\forall a = (i,j) \in \mathcal{A}_{\text{head}}, & (4.6)\\
g_{ij} + g_{ji} &= 1 & &\forall (i,j) \in \mathcal{A}_{\text{head}}, & (4.7)\\
x_i &\in \mathbb{N} & &\forall i \in \mathcal{E}, & (4.8)\\
z_a &\in \{0,1\} & &\forall a \in \mathcal{A}_{\text{change}}, & (4.9)\\
g_{ij} &\in \{0,1\} & &\forall (i,j) \in \mathcal{A}_{\text{head}}. & (4.10)
\end{aligned}$$

The objective function in this model counts the sum of delays of all events (weighted with the number of passengers c_i who arrive at their final destination at event i) and adds a penalty of T for every passenger who misses a connection. In a periodic timetable, T is often chosen as its cycle time. We weigh the transfer activity a with the number of passengers c_a who planned to use it. The objective is an approximation of the overall delay of all passengers and is commonly used in delay management. It gives the exact value if the never-meet property for headways holds (see Schachtebeck and Schöbel (2010)). A more realistic model taking into account the real paths that passengers would use in case of delays has been developed in Chapter 2. It can also be used as basis for our extension, but it is technically more difficult and computationally harder to solve.
The interpretation of the constraints is as follows. (4.2) makes sure that trains do not depart earlier than planned and that source delays at events are taken into account. (4.3) propagates the delay along waiting and driving activities while (4.4) propagates the delay along *maintained* changing activities. For each pair of departure events competing for the same infrastructure, (4.7) makes sure that exactly one of the two precedence relations is respected. (4.5) propagates the delay along the corresponding headway activity between the departures of the trains. Similarly, (4.6) propagates the delay between the arrivals of the trains. Here i' and j' are the arrivals that follow the departures i and j, respectively.

4.3.2 Formulation with a dynamic platform assignment

To include the limited capacity within the stations, we now present a formulation for delay management which allows a dynamic assignment of trains to platform tracks. Preliminary computational results showed that this assignment-based formulation performs much better than a packing-based formulation modeling the same problem (see Dollevoet et al. (2011)). Our assignment-based integer programming formulation views a station as a set of platforms, and introduces headway constraints for trains that make use of the same platform track. As a consequence, this formulation determines an explicit allocation of the events to the available platforms.
In order to allocate the trains to the platforms, we first define for each station $s \in S$ the set P_s of platforms at s and the set $\mathcal{E}^s_{\text{arr}}$ of arrival events at s. Then, we introduce binary decision variables y_{ip} for each event $i \in \mathcal{E}^s_{\text{arr}}$ and $p \in P_s$, that are defined as

$$y_{ip} = \begin{cases} 1 & \text{if arrival } i \text{ and the corresponding departure are assigned to platform track } p, \\ 0 & \text{otherwise.} \end{cases}$$

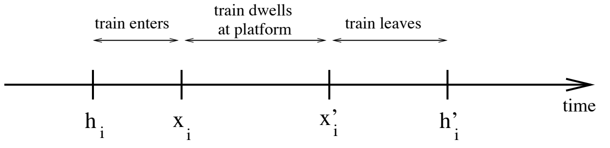

Figure 4.4: Illustration of the enter time, the leave time and the time during which the platform track is occupied.

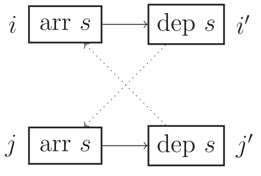

Figure 4.5: When two trains use the same platform track, a pair of headway activities is introduced from the departure of one train to the arrival of the other.

Of course, each arrival event must be assigned to exactly one platform track. This is enforced by the following constraint.

$$\sum_{p \in P_s} y_{ip} = 1, \quad \forall s \in S, i \in \mathcal{E}^s_{\text{arr}}. \tag{4.11}$$

In order to model the limited capacity of the stations, we determine the order in which the trains arrive at a certain platform track. Consider a pair of trains (t_1, t_2) that arrive at the same station corresponding to two events i and j. If the two trains are assigned to the same platform track, we must determine the order in which the events i and j take place. To this end, we introduce a pair of binary variables $\bar{g}_{ij}$ and $\bar{g}_{ji}$ that are defined as follows

$$\bar{g}_{ij} = \begin{cases} 0 & \text{if arrival } i \text{ takes place before arrival } j \text{ on the same platform track,} \\ 1 & \text{otherwise.} \end{cases}$$

If the trains are assigned to the same platform track, either t_1 must have departed before t_2 arrives, or t_2 must have departed before t_1 arrives. It should be noted that a train starts entering a station at a time h_i before it stops there at time x_i and passengers can board. The time h_i when the train starts to enter the station is called *enter time*. In the same way, the departure time $x_{i'}$ of a train is smaller than the *leave time* $h_{i'}$, which is the time when the last car of the train leaves the platform track and hence the time when the next train can start to enter. Thus $[h_i, h_{i'}]$ denotes the interval during which a platform track is occupied (see Figure 4.4).

We define $l_i = x_i - h_i$ for arrival events and $l_{i'} = h_{i'} - x_{i'}$ for departure events. By construction, l_i and $l_{i'}$ are non-negative. We define the headway time $L_{ij} = l_i + l_{j'}$ as the time that is minimally needed between the departure i and the arrival j.

In Figure 4.5 the event-activity network for the trains t_1 and t_2 is depicted. Let $a_i = (i, i')$ be the waiting activity of train t_1 and let $a_j = (j, j')$ be the waiting activity of train t_2. We define a pair of platform track activities $a_1 = (i', j), a_2 = (j', i) \in \mathcal{A}_{\text{plat}}$ and introduce the following set of constraints.

$$M\bar{g}_{ij} + x_j - x_{i'} \geq L_{ij} = l_{i'} + l_j, \tag{4.12}$$

$$M\bar{g}_{ji} + x_i - x_{j'} \geq L_{ji} = l_{j'} + l_i, \tag{4.13}$$

$$\bar{g}_{ij} + \bar{g}_{ji} \leq 3 - y_{ip} - y_{jp} \quad \forall p. \tag{4.14}$$

These constraints can be interpreted in the following way. Assume first that trains t_1 and t_2 are not assigned to the same platform track. Then $3 - y_{ip} - y_{jp} \geq 2$ for all p. Hence, both $\bar{g}_{ij}$ and $\bar{g}_{ji}$ can be set to 1 and (4.12) and (4.13) are satisfied. Otherwise, if trains t_1 and t_2 are assigned to the same platform track p, then $3 - y_{ip} - y_{jp} = 1$ for that p, forcing either $\bar{g}_{ij}$ or $\bar{g}_{ji}$ to zero. In that case, one of the headway constraints enforces a minimal headway time between the two trains.

The above constraints must be introduced for each pair of trains (t_1, t_2) that dwell at a common station $s \in S$. Recall that the set of platform track activities is denoted by $\mathcal{A}_{\text{plat}}$. This formulation thus introduces one binary variable $\bar{g}_{ij}$ for each $a = (i, j) \in \mathcal{A}_{\text{plat}}$ and one binary variable y_{ip} for each $i \in \mathcal{E}_{\text{arr}}$ and $p \in P_s$, where $s \in S$ is the station corresponding to arrival i. Furthermore, it adds one constraint for each arrival event $i \in \mathcal{E}_{\text{arr}}$ and $2 + |P_s|$ constraints for each pair of trains (t_1, t_2) that dwell at a common station $s \in S$. Note that this type of constraints are referred to as blocking constraints in the context of job-shop scheduling (see Mascis and Pacciarelli (2002)).

Adding the constraints (4.11)-(4.14), $y_{ip} \in \{0, 1\}$ for all $s \in S$, $i \in \mathcal{E}^s_{\text{arr}}, p \in P_s$ and $\bar{g}_{ij} \in \{0, 1\}$ for all $(i, j) \in \mathcal{A}_{\text{plat}}$ to the formulation (4.1)-(4.10) we obtain an integer programming formulation (DM-Cap) for the delay management problem with a *dynamic platform assignment*.

4.3.3 Formulation with a static platform assignment

Note that the planned timetable provides us with a platform assignment. To avoid platform track changes for the passengers and, at the same time, simplify our calculations, we could fix this platform assignment, *i.e.*, the variables y_{ip}. More generally, we call the delay management problem for which a platform assignment is determined in advance *delay management with a static platform assignment*. For delay management with a static platform assignment, the above integer programming formulation reduces to a problem of type (DM).

Lemma 4.1 *For fixed variables y_{ip} for all $i \in \mathcal{E}^s_{arr}, p \in P_s$ the formulation (DM-Cap) reduces to an instance of (DM),* i.e., *can be solved as a delay management problem with headway constraints.*

Proof If all y_{ip} variables are fixed we have two possibilities for (4.14): Either both y_{ip} variables are 1, then $\bar{g}_{ij} + \bar{g}_{ji} \leq 1$ and (4.12)-(4.13) reduce to a headway constraint of type (4.5)-(4.7), or at least one of the y_{ip} variables is 0, then (4.12)-(4.14) becomes redundant. □

According to this lemma, we can derive the following two bounds for delay management with a dynamic platform assignment, which can easily be calculated using an algorithm that solves problem (DM). First, it is clear that (DM) is a relaxation of (DM-Cap), hence its objective value z^{DM} is a lower bound. Second, if we fix the assignment y of trains to stations in (DM-Cap) and solve the delay management problem with a static platform assignment, we obtain an upper bound $z^*(y)$ which can also be calculated by any algorithm for (DM) according to Lemma 4.1. Hence, we can compute an upper and a lower bound, *i.e.*, $z^{DM} \leq z^* \leq z^*(y)$.

4.4 An iterative approach

In the previous section we developed a model that simultaneously optimizes the platform assignment and the priority decisions in the stations. It is well known in timetabling that this problem is computationally challenging. As we consider a real-time setting, solutions to the delay management problem should be available within a short computation time. For large instances, optimizing the wait-depart decisions, the priority decisions and the platform assignments simultaneously might be intractable. For these instances, we propose an iterative approach: We first fix the assignment of trains to platforms as given in the original timetable. This results in a problem of type (DM) which can be solved according to Schachtebeck and Schöbel (2010). For the resulting solution we then try to improve the platform assignment within the stations and iterate until no further improvement is found. Using formulation (DM-Cap) we obtain:

1. Fix the station assignment y_{ip} in (DM-Cap) according to the planned timetable.
2. Solve the resulting problem (DM-Cap) with fixed y_{ip} and obtain a solution with disposition timetable x_i, wait depart decisions z_a and priority decisions g_{ij} and $\bar{g}_{ij}$
3. For every station, determine a more promising platform assignment y_{ip} and new priority decisions $\bar{g}_{ij}$ within the station such that $(x, z, y, g, \bar{g})$ is feasible.
4. Go to Step 2. Stop if no further improvement has been found.

If for large delay management instances decomposing the problem into two steps still results in long running times, we can use the approach of Schachtebeck (2010) to decompose Step 2 of the algorithm further into two smaller subproblems making first the priority decisions and then the wait-depart decisions.
In Step 3, a natural idea would be to adjust not only the platform assignment but also the timetable locally. Unfortunately, this can lead to infeasible solutions. Therefore, in Step 3 of the algorithm, we leave the timetable unchanged and adjust only the platform assignment in a way that allows the subsequent delay management step to shift events forward in time, if possible.
In the following we discuss Step 3, *i.e.*, how to find an assignment of trains to platforms at a given station s which is feasible for the given disposition timetable x and potentially yields a better disposition timetable in Step 2 of the next iteration. Recall from (4.12) and (4.13) that the headway times L_{ij} between two trains are the sum of a headway time $l_{i'}$ that is needed for the first train to leave the station after its departure event i' and a headway time l_j representing the time that the second train needs to completely enter the station before its arrival event j can take place, *i.e.*, $L_{ij} = l_{i'} + l_j$. Thus instead of scheduling the arrival and departure events x_i, we can instead schedule the enter time $h_i = x_i - l_i$ for arrival events i and the leave time $h_{i'} = x_{i'} + l_{i'}$ for departure events i' in a way that the intervals $(h_i, h_{i'})$ and $(h_j, h_{j'})$ do not overlap for two trains with arrival and departure events i, i' and j, j', respectively, that are assigned to the same platform track. We process each station separately as follows. In a first step we identify for which arrivals $i \in \mathcal{E}$ in this station a new assignment might be beneficial. These are arrivals of delayed trains that directly follow another delayed train. For these train arrivals we

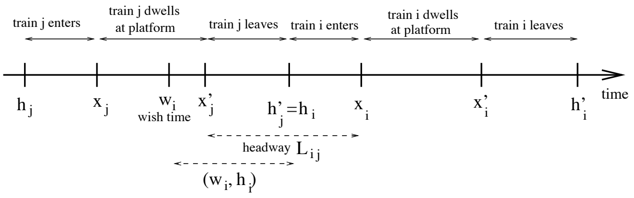

Figure 4.6: Illustration for two trains occupying the same platform track.

determine their *wish (enter) times* w_i. In a second step we find a new assignment for all trains together with new enter times $q_i \geq w_i$ for these trains which should be as close to the wish times as possible. We now first show how the *wish times* are identified.

Let P_s be the set of platforms and $\mathcal{E}^s_{\text{arr}}$ be the set of arrival events in station s. Note that every such event corresponds to one train. For every arrival event i, let i' be the departure event following i (*i.e.*, $(i, i') \in \mathcal{A}_{\text{wait}}$ describes the waiting activity of the train in the station). From the timetable and the headway times we know that the train occupies the station during the time interval $(h_i, h_{i'})$. If a train is delayed, we distinguish two cases (see Figure 4.6).

- There is another train which is in the station during the interval $(h_j, h_{j'})$ with $h_{j'} = h_i$, and is on the same platform track p, *i.e.*, $y_{ip} = y_{jp} = 1$. In this case, a new assignment might help to reduce the delay of i. Assuming that $a = (k, i) \in \mathcal{A}_{\text{drive}}$ is the preceding driving activity of the train we define the *wish time* of i as

$$w_i := x_k + L_a + d_a - l_i.$$

- If no other train is on the same platform track directly before x_i, the delay of i is not due to the station assignment, and hence $w_i := h_i$.

Also if the train is not delayed we set $w_i := h_i$. The *platform assignment* problem (PA) can now be formulated as follows.

(PA) *Given a set of platforms $P_s = \{1, \ldots, P\}$ and for every arrival event $i \in \mathcal{E}^s_{arr}$ an interval $[h_i, h_{i'}]$ and a wish time $w_i \leq h_i$ as well as a weight c_i corresponding to the affected customers on the train, find numbers $q_i \in [w_i, h_i]$ for all $i \in \mathcal{E}^s_{arr}$ and a new platform assignment y_{ip} for all $i \in \mathcal{E}^s_{arr}$ and $p \in P_s$ such that*

$$q_j \in (q_i, h_{i'}) \Longrightarrow y_{ip} + y_{jp} \leq 1 \qquad (4.15)$$

holds for all $i, j \in \mathcal{E}^s_{arr}$ *and* $p \in P_s$ *and* $\sum_{i \in \mathcal{E}^s_{arr}} c_i q_i$ *is minimal.*

Note that $q_j \in (q_i, h_{i'})$ or $q_i \in (q_j, h_{j'}) \iff (q_i, h_{i'}) \cap (q_j, h_{j'}) \neq \emptyset$, *i.e.*, the two trains belonging to i and j cannot be scheduled on the same platform track if and only if the arrival of one train is scheduled at a time when the other train is occupying the platform track.

This problem can be formulated as a mixed-integer program as it is but the formulation does not seem to be promising due to condition (4.15). Instead we show that (PA) is polynomially solvable by first identifying a finite dominating set $\mathcal{C}$ for the q_i variables. This set $\mathcal{C}$ contains a polynomial number of elements. We then notice that for every choice of the q_i variables, we can check feasibility by solving a coloring problem. Naively, in order to check all possible $q \in \mathcal{C}^{|\mathcal{E}^s_{\mathrm{arr}}|}$, we would have to solve an exponential number of coloring problems. Instead, we use that the graph under consideration is an interval graph and code the solvability of the coloring problem in the constraints of an IP formulation. This problem can be solved easily, as the coefficient matrix is totally unimodular. We first identify a dominating set $\mathcal{C}$ with a polynomial number of elements. In order to reduce the problem size, we also show that for each q_i, a smaller dominating set $\mathcal{C}_i$ can be defined.

Lemma 4.2 *Let* $\mathcal{C} := \bigcup_{i \in \mathcal{E}^s_{arr}} \{w_i, h_i, h_{i'}\}$ *be the set of all given wish and planned arrival and departure times. Then there exists an optimal solution* (q, y) *to (PA) with* $q_i \in \mathcal{C}_i := \mathcal{C} \cap [w_i, h_i]$ *for all* $i \in \mathcal{E}^s_{arr}$.

Proof Let (q, y) be a feasible solution to (PA). Clearly, $w_i \leq q_i \leq h_i$ for all i. Furthermore, with p the platform track for which $y_{ip} = 1$, $q_i \geq \max\{h_{j'} : y_{jp} = 1 \text{ and } h_{j'} \leq q_i\}$. Now assume that $q_i \notin \mathcal{C}$ for some $i \in \mathcal{E}^s_{\mathrm{arr}}$. Let p be the platform track with $y_{ip} = 1$. Define

$$\tilde{q}_i := \max\left\{w_i, \max\{h_{j'} : y_{jp} = 1 \text{ and } h_{j'} \leq q_i\}\right\}. \tag{4.16}$$

Then $\tilde{q}_i \in [w_i, h_i]$ and for all j condition (4.15) is still satisfied. Hence, replacing q_i by $\tilde{q}_i$ is a feasible solution to (PA) with better objective value and with $\tilde{q}_i \in \mathcal{C}_i$. Doing this for all values q shows the result. □

Now assume that some values $q_i \in \mathcal{C}_i$, $i \in \mathcal{E}^s_{\mathrm{arr}}$ are given. How can we check whether q is feasible? This means that we have to check whether there is a platform assignment y such that (4.15) is satisfied. To this end we transform our problem into a coloring problem in the graph $G(q) = (\mathcal{E}^s_{\mathrm{arr}}, E)$. For every $i \in \mathcal{E}^s_{\mathrm{arr}}$ there exists a node. We add an edge $\{i, j\}$ between two nodes if $(q_i, h_{i'}) \cap (q_j, h_{j'}) \neq \emptyset$, *i.e.*, if the two corresponding trains cannot be assigned to the same platform track. In order to find out whether there is a feasible platform assignment for q we thus have to find out whether $G(q)$ is P-colorable. Note that by construction this graph is an interval graph and thus perfect (see *e.g.* Schrijver (2003)). Thus $\chi(G(q)) = \omega(G(q))$ with $\chi(G(q))$ denoting the chromatic number of $G(q)$ and $\omega(G(q))$ the number of nodes in the biggest clique of $G(q)$. We hence have to check whether the number of nodes in the biggest clique in $G(q)$ is not greater than P.

Let us order the values in $\mathcal{C} = \{q^1, \ldots, q^{|\mathcal{C}|}\}$ in increasing order and let us define intervals $I_l := (q^l, q^{l+1})$ for $l = 1, \ldots |\mathcal{C}| - 1$. For a given q we define a matrix $A(q) = (a_{li})$ with $|\mathcal{C}| - 1$ rows and $|\mathcal{E}^s_{\mathrm{arr}}|$ columns and entries

$$a_{li} = \begin{cases} 1 & \text{if } (q_i, h_{i'}) \cap I_l \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{4.17}$$

Then we can determine the chromatic number of $G(q)$ as follows.

Lemma 4.3

$$\omega(G(q)) = \max_{l=1,\ldots,|\mathcal{C}|-1} \sum_{i \in \mathcal{E}^s_{arr}} a_{li}.$$

Proof Due to Lemma 4.2 we can assume that all values of q_i are in $\mathcal{C}$, hence there is an edge between i and j in $G(q)$ if and only if there exists an interval I_l such that $a_{li} = a_{lj} = 1$. Now let $\mathcal{E}' \subseteq \mathcal{E}^s_{\mathrm{arr}}$. As $G(q)$ is an interval graph, $\mathcal{E}'$ is a clique in $G(q)$ if and only if there exists *one* interval I_l such that $a_{li} = 1$ for all $i \in \mathcal{E}'$. $\square$

Now we can finally rewrite (PA) as an integer program in which we look for a choice of q-values from the set $\mathcal{C}$ checking feasibility by Lemma 4.3. Denote by q_i^k the entries of the set $\mathcal{C}_i = \{q_i^1, q_i^2, \ldots, q_i^{|\mathcal{C}_i|}\}$. Then for every arrival event i and every candidate $q_i^k \in \mathcal{C}_i$ we define the variable

$$\eta_i^k = \begin{cases} 1 & \text{if candidate } q_i^k \in \mathcal{C}_i \text{ is chosen,} \\ 0 & \text{otherwise.} \end{cases}$$

These are the variables of our integer program. In order to directly see properties of the resulting constraint matrix, we order our variables such that all variables η_i^k having the same index i are grouped together. We need to extend the matrix defined in (4.17) to all possible choices of q. To this end, we define for every (i, k) a column with

$$\tilde{a}_{lik} = \begin{cases} 1 & \text{if } (q_i^k, h_{i'}) \cap I_l \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

Doing this for all $i = 1, \dots, |\mathcal{E}^s_{\text{arr}}|$ we obtain a matrix $\tilde{A} = (\tilde{a}_{lik})$ with $|\mathcal{C}| - 1$ rows and $\sum_{i \in \mathcal{E}^s_{\text{arr}}} |\mathcal{C}_i|$ columns. Note that $q^k_i = q^{k'}_j$ with $q^k_i \in \mathcal{C}_i$, $q^{k'}_j \in \mathcal{C}_j$ is possible but would lead to two (maybe different) columns in $\tilde{A}$.
(PA) can hence be rewritten as

$$\min \sum_{i \in \mathcal{E}^s_{\text{arr}}} c_i \sum_{k=1}^{|\mathcal{C}_i|} q^k_i \eta^k_i \tag{4.18}$$

such that

$$\sum_{k=1}^{|\mathcal{C}_i|} \eta^k_i = 1 \qquad \forall i \in \mathcal{E}^s_{\text{arr}}, \tag{4.19}$$

$$\sum_{i \in \mathcal{E}^s_{\text{arr}}} \sum_{k=1}^{|\mathcal{C}_i|} \tilde{a}_{lik} \eta^k_i \leq P \qquad l = 1, \dots, |\mathcal{C}| - 1, \tag{4.20}$$

$$\eta^k_i \in \{0, 1\} \qquad \forall i \in \mathcal{E}^s_{\text{arr}},\ \forall k \in \mathcal{C}_i. \tag{4.21}$$

Lemma 4.4 *The constraint matrix A' defined by the inequalities (4.19)-(4.20) is totally unimodular.*

Proof To avoid notational confusion, for the matrix A' we use u as an index for the rows and v as an index for the columns. We show that A' is totally unimodular by showing that every subset U of rows of A' can be partitioned into two sets U_1, U_2 with $U_1 \cap U_2 = \emptyset$, $U_1 \cup U_2 = U$ and $\sum_{u \in U_1} a'_{uv} - \sum_{u \in U_2} a'_{uv} \in \{-1, 0, 1\}$ for all columns v (see for example Schrijver (2003)).
The columns of A' are associated with the variables of our integer program. For every $i = 1, \dots, |\mathcal{E}^s_{\text{arr}}|$ we denote by $C(i)$ the indices v of the columns of A' associated with a variable η^k_i for some k.
The rows represent the constraints. The first rows $u = 1, \dots, |\mathcal{E}^s_{\text{arr}}|$ contain the constraints that for every $i \in \mathcal{E}^s_{\text{arr}}$, exactly one variable η^k_i is set to 1. We thus have

$$a'_{uv} = \begin{cases} 1 & \text{if the column } v \text{ belongs to variable } \eta^k_u \text{ for a } k, \textit{i.e.}, \text{ if } v \in C(u), \\ 0 & \text{otherwise.} \end{cases}$$

for $u = 1, \dots, |\mathcal{E}^s_{\text{arr}}|$, *i.e.*, for u corresponding to a constraint for an $i \in \mathcal{E}^s_{\text{arr}}$.
Starting from row $|\mathcal{E}^s_{\text{arr}}| + 1$, the matrix A' consists of the matrix $\tilde{A}$. We notice that $\tilde{A}$ has the column-wise consecutive ones property. Furthermore, we note that every column $v \in C(i)$ of A' has its last 1-entry in the row that represents the constraint for the interval

with end point h_i.
Let U be an index set of rows of A' and $U^A = U \setminus \{1, \ldots, |\mathcal{E}^s_{\text{arr}}|\}$, that is the part of the chosen set of rows that is contained in $\tilde{A}$. We alternatingly assign the rows in U^A to two sets U^A_1 and U^A_2. Then for every $i \in \mathcal{E}^s_{\text{arr}}$ either

$$\left\{ \sum_{u \in U^A_1} a'_{uv} - \sum_{u \in U^A_2} a'_{uv} : v \in C(i) \right\} \subseteq \{-1, 0\} \tag{4.22}$$

$$\text{or} \left\{ \sum_{u \in U^A_1} a'_{uv} - \sum_{u \in U^A_2} a'_{uv} : v \in C(i) \right\} \subseteq \{1, 0\} \tag{4.23}$$

because of the consecutive ones property and because for a given i, the last entry of column v is in the same row for all $v \in C(i)$.
We set $U_1 := U^A_1$ and $U_2 := U^A_2$ and add the indices of the first $|\mathcal{E}^s_{\text{arr}}|$ rows in the following way to these sets: If for row u (4.22) holds, we assign the u-th row to U_1, if (4.23) holds we assign it to U_2. We obtain

$$\left\{ \sum_{u \in U_1} a'_{uv} - \sum_{u \in U_2} a'_{uv} : v \in C(i) \right\} \subseteq \{1, 0\} \text{ for all } i \in \{1, \ldots, |\mathcal{E}^s_{\text{arr}}|\} \text{ with (4.22)}$$

$$\left\{ \sum_{u \in U_1} a'_{uv} - \sum_{u \in U_2} a'_{uv} : v \in C(i) \right\} \subseteq \{-1, 0\} \text{ for all } i \in \{1, \ldots, |\mathcal{E}^s_{\text{arr}}|\} \text{ with (4.23).}$$

This proves total unimodularity. □

Corollary 4.5 *(PA) can be solved by linear programming.*

We conclude that the problem in Step 3 of the iterative algorithm can be solved by linear programming. This completes the description of the iterative algorithm. An alternative way to solve Step 3 of the iterative algorithm is by modeling (PA) as a network flow problem. This approach has been proposed by Kroon (1990) in the context of aircraft maintenance. Note that the network flow formulation allows for a solution approach that does not need an LP solver.

4.5 Computational results

We have performed a computational study to test whether it is important to consider the capacity within stations explicitly and to compare the different approaches presented in this chapter. We first describe the cases that were used in this study. Then we show that a dynamic platform assignment significantly improves a static one. Finally, we evaluate the performance of the iterative heuristic.

4.5.1 Cases

In our numerical experiments we consider the railway system in the Randstad, which is the mid-Western part of the Netherlands. Figure 4.1 gives a schematic representation of the railway network in this region. The dots in this figure indicate a station where long-distance trains stop. The stations where only regional trains stop are not depicted. A line indicates that there is a direct link between two stations. For each link, there are two or four long-distance trains and two regional trains per hour. It can be seen in the picture that the railway network contains direct links between many of the stations. As a consequence, the infrastructure is heavily utilized, especially in the stations.

We have generated four cases, that vary in the size of the network and the type of trains that are included. The first case considers only the stations in the network that are indicated by a black dot in Figure 4.1 and includes only the long-distance trains. Case B considers the same network, but includes both long-distance and regional trains. The third and fourth case include all stations that are indicated by a black or a white dot. Again, Case C considers only the long-distance trains, while Case D includes the regional trains, too. These cases resemble those that are used in Chapter 2.

We obtained the timetable and detailed information on the passenger demand from Netherlands Railways. The passenger figures are not the real numbers, but have been scaled for secrecy. For each pair of stations in the network, we were given the average number of passengers who want to travel between these stations on a regular day. From these origin-destination figures we obtained the average number of passengers w_i who arrive at their destination station with arrival event $i \in \mathcal{E}_{\text{arr}}$ and the number of passengers w_a who use transfer $a \in \mathcal{A}_{\text{change}}$.

In order to evaluate the performance of our delay management models, we have simulated for each case 100 delay scenarios. These scenarios were constructed as follows. Each driving and dwell activity has a probability of 10% to be delayed. If the activity is delayed, the size of the delay is a uniformly distributed random variable between 1 and 10 minutes. Note that delays on activities are additive: If two consecutive driving activities are delayed, the delay of the train is at least the sum of the two delays. We did not include delays at events.

Table 4.1 gives an impression of the sizes of the instances. For each of the four cases we

			Size of the program			Static model		Dynamic model	
Case	Stations	Trains	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{A}_{\text{head}}\rvert$	$\lvert\mathcal{A}_{\text{plat}}\rvert$	Bin.	Con.	Bin.	Con.
A	10	82	344	1508	6962	1998	5470	9597	33924
B	34	193	1374	3432	28830	8045	21568	37972	175614
C	16	119	576	2296	11688	3345	8895	15913	55301
D	82	275	2804	5848	35138	12877	36510	48841	231101

Table 4.1: Some characteristics of the cases and the resulting integer programs. Bin. and Con. give the number of binary variables and constraints in the integer program, respectively.

report the number of stations and trains in the railway network. Besides, we report the number of events $|\mathcal{E}|$ and headway activities $|\mathcal{A}_{\text{head}}|$ in the resulting event-activity network. The column $|\mathcal{A}_{\text{plat}}|$ gives the number of platform track activities. Recall that there is a *pair* of platform track activities for each pair of trains (t_1, t_2) that dwell at a common station. $|\mathcal{A}_{\text{plat}}|$ is therefore *twice* the number of times that Constraints (4.12)-(4.14) are added to formulation (4.1)-(4.10). Cases B and D consider all trains in a large part of the network. Cases of these sizes arise in practical applications. Comparing Cases A and B, one sees that the number of trains is increased roughly by 50%. The number of nodes in the event-activity network is about 4 times as large. The reason is that the regional trains stop at far more stations than the long-distance trains. As priority decisions are only necessary at the larger stations, where overtaking can take place, the number of headway activities is related to the number of trains.

For each instance we have two different models to obtain a solution to the delay management problem with station capacities. Our first model fixes the platform assignment as planned. According to Lemma 4.1, this leads to a delay management problem with headway activities only. According to Section 4.3.3, we refer to this model as the static model. In the second, dynamic model, we allow the platform tracks to be rescheduled dynamically as introduced in Section 4.3.2. In Table 4.1, we also list the number of binary variables and constraints in the resulting integer programs. One sees immediately that the number of binary variables and constraints is much larger for the dynamic model. As a consequence, the dynamic model is expected to be computationally much harder to solve.

	Objective value				Optimality gap			
Case	A	B	C	D	A	B	C	D
Static model	192291	1007849	595904	3635493	0%	0%	0%	0.3%
Dynamic model	192046	982983	585987	3572731	0%	0.1%	0%	1.3%

Table 4.2: The objective value and the optimality gap for the static and dynamic delay management model

Case	A	B	C	D
Improvement	0.1%	2.5%	1.7%	1.7%
Platform track changes	167	342	269	380

Table 4.3: The improvement of the dynamic model with respect to the static model and the number of platform track changes in the solution of the dynamic model

4.5.2 Static and dynamic platform assignments

We have used Cplex 12.2 on an Intel Core i5-2410M with 4 GB of RAM to solve the integer programs from Section 4.3. The maximal computation time was set to 20 minutes for each individual delay scenario. Such times are too long for practical purposes, but allow us to find solutions that are close to optimal. The objective value for a case is computed as the average total delay over all scenarios. For each case, we compare the solutions that are obtained with a static and with a dynamic platform assignment.

In Table 4.2, the objective values are presented for both solution approaches. In Table 4.3, the relative improvement of the dynamic model is given, as well as the number of platform track changes in the solutions of the dynamic model. The results show clearly that rescheduling the platform assignment dynamically reduces the delay for the passengers. For Case A, the reduction is negligible, the average delay is reduced only by 0.1%. For Case B, the delay is reduced by 2.5%. This is a significant improvement over the static model. In the Cases C and D, the reduction is 1.7%. The optimality gap in the first three cases can be neglected. In Case D, the optimality gap for the dynamic model is 1.3%. The improvement of the dynamic model over the static model could therefore be larger than 1.7 %.

Besides the delay, platform track changes are also inconvenient for the passengers. The static model does not allow to reschedule the platform assignment, so in the static solutions there are no platform track changes. On the contrary, in the dynamic model a complete new platform assignment is determined. As we do not penalize changes in the platform assignment, the dynamic model introduced hundreds of platform track changes

Case	A	B	C	D
Static model	1.0 (1.6)	62.9 (1162)	2.0 (3.5)	765.6 (1204)
Dynamic model	3.1 (5.4)	608.7 (1353)	13.9 (112.7)	1408 (1478)

Table 4.4: The average running times and between brackets the maximal running times in seconds

for every case.

As solutions to the delay management problem should be readily available, solution methods should be able to solve the delay management problem within a short computation time. In Table 4.4, the average and maximal running times are given for each case. Recall that we have set the maximal running time to 20 minutes for each delay scenario. For Cases A and C, which include only the long-distance trains, both models can be solved very fast. Both the average and maximal running time are less than two minutes. Such running times are acceptable in practice.

When the regional trains are included, the running times increase significantly. In order to speed up the solution process for Cases B and D, we first computed a solution to the static delay management model in which the order of trains in the stations and on the tracks is fixed. This results in an integer program that is much easier and can be solved within seconds. The solution to this model is also feasible for the dynamic model and can be used to decrease the solution times. In a similar fashion, a solution from the static model can be used when solving the dynamic model. When solving the dynamic model, we first ran the algorithm for the static model for 5 minutes. This explains why the maximal running times for the dynamic model are larger than 20 minutes.

For Case B, the static model can be solved within one minute on average. The dynamic model needs 10 minutes on average. For both models, some delay scenarios need much more computation time. For Case D, the static model can be solved within 12 minutes on average, while the dynamic model takes the full computation time of 25 minutes. In a real-time setting, such computation times are too long.

4.5.3 Performance of the iterative heuristic

In the previous section we have seen how a dynamic platform assignment can reduce the delay for the passengers. However, for larger cases, solving the dynamic model takes too much time. In these situations, the iterative heuristic from Section 4.4 can be applied to improve on the static solutions while still keeping the computation times within limits.

In Table 4.5, the results for the iterative heuristic are given. For all cases, the iterative

Case	A	B	C	D
Absolute objective	192088	992757	587307	3594245
Relative objective	100.0 %	101.0 %	100.2 %	100.6 %
Number of platform track changes	0.32	17.5	3.2	21.0
Number of iterations	2.06	4.14	2.72	4.81
Total solution time	2.2	174.1	4.76	1198

Table 4.5: The absolute objective value, the relative objective values with respect to the dynamic model, the number of platform track changes in the solutions and some characteristics of the iterative solution process

heuristic finds solutions that are at most 1.0% worse than the dynamic model. For the cases that consider only the long-distance trains, the iterative solutions are very close to the optimal ones. It is interesting to see that the iterative algorithm changes the platform assignment less often than the dynamic model. The iterative algorithm only schedules a train at a different platform track if it looks promising to do so. The slight decrease in quality is thus compensated by a platform assignment that looks much more like the original one.

Considering the computation times, the iterative approach solves the problem much faster than the dynamic model. For Cases A and C, the iterative algorithm can be executed within seconds. For Case B, the total running time of the iterative heuristic is on average less than 3 minutes. This is a reduction of 70% with respect to solving the dynamic model. A solution time of 3 minutes is acceptable in practice. For Case D, solving the static model to optimality already takes a longer time. We therefore allowed only 5 minutes of computation time per iteration. As a consequence, the first solution in the iterative procedure is slightly worse than the solution of the static model. The average total running time for Case D equals 20 minutes and is thus smaller than that of the dynamic model. It is, however, still too long for practical applications. We conclude that for cases with only long distance trains, both the performance and the running time of the dynamic model and the iterative algorithm are comparable. For Case B, solutions with slightly worse quality are obtained within much less computation time. For this case, the iterative algorithm finds good solutions within computation times that are acceptable in practice. Finally, for Case D, solutions are found that have a good quality, but the iterative algorithm needs too much time to find these solutions.

In Figure 4.7, we have plotted the progress of the iterative method. On the horizontal axis are the iterations, while the objective value is shown on the vertical axis. In order to show the progress for all cases in one figure, we have normalized the objective value.

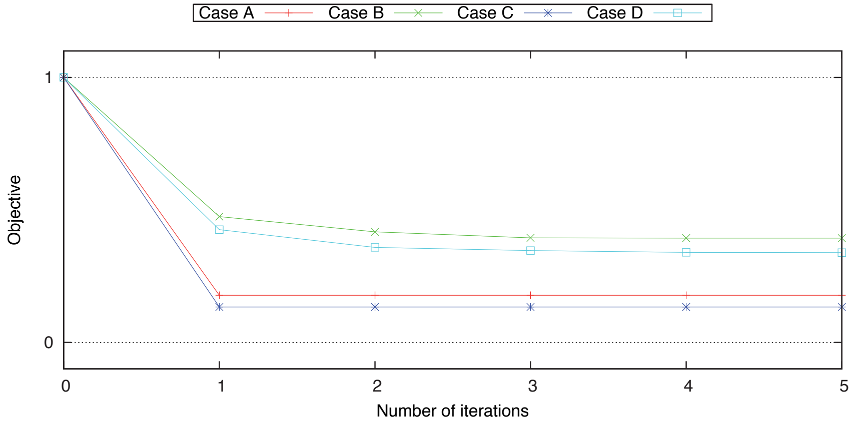

Figure 4.7: The progress of the iterative solution

The objective value of the first iteration is equal to that of the static model. This value is indicated by the upper line, labeled with 1. A lower bound on the objective value from the iterative approach is given by the solutions of the dynamic model. This value is depicted by the lower line, labeled with 0.

For Cases A and C, the iterative algorithm improves the solution from the static model only in the first iteration. The solution that is then obtained is very close to the optimal solution. For Cases B and D, the biggest improvement is found in the first iteration. The improvement in the second iteration is much smaller. From then on, the solution does not change much. This suggests that one could also run only one iteration of the iterative algorithm, in order to improve over the static solution within a short computation time. For Case D, it takes on average 10 minutes to perform one iteration. In practical applications, computation times of 10 minutes are acceptable.

4.5.4 Quality of the wait-depart decisions

The dynamic model optimizes the wait-depart decisions, the priority decisions and the platform assignment simultaneously. The original aim of delay management is, however, to decide on the wait-depart decisions only. In this section we will compare the quality of the wait-depart decisions from the dynamic model to those that are obtained with other delay management models from literature. If an optimization algorithm is available to determine the priority decisions and the platform assignment, the results in this section prescribe which delay management model should complement the train scheduling algorithm and optimize the wait-depart decisions.

Case	A	B	C	D
No-wait policy	120.1	140.4	124.5	134.4
DM without capacity constraints	102.6	108.7	101.7	108.7
DM with tracks priorities only	100.0	100.0	100.0	100.3
Dynamic model	100.0	100.0	100.0	100.0

Table 4.6: Comparing the dynamic model to other delay management models

In order to compare the quality of the different delay management models, we first decide on the wait-depart decisions with the various models. Then we fix the variables for the wait-depart decisions in the dynamic model in order to obtain the priority decisions and the platform assignment. We emphasize that we use the dynamic model only to compute the priority decisions and the platform assignment; the wait-depart decisions have been fixed by the model that we are interested in. We thus employ the dynamic model as a very basic train scheduling algorithm.

The first model that we consider implements a no-wait policy. With a no-wait policy, trains never wait for delayed trains. We use this policy as a benchmark for the other policies. The second model is a delay management model without any capacity considerations. This model is given by (4.1)-(4.4) and (4.8)-(4.9). The third model is a delay management model with priority decisions for tracks only. This model is presented in Section 4.3.1. Finally, the fourth model is the dynamic model that is introduced in Section 4.3.2.

In Table 4.6, the objective values are given for the four models. We have normalized the objective value and set the objective value of the dynamic model to 100. We see in the table that a no-wait policy performs very badly. The total delay is up to 40% higher than in the dynamic model. A delay management model without capacity considerations performs better. For Cases A and C, the delay is increased with 2%, while an increase of about 9% is observed for Cases B and D. Finally, the model that incorporates capacity constraints for tracks only performs very well. The maximal increase in delay is 0.3%. For the first three cases, the increase is even negligible.

We conclude that the model that includes headway constraints on the tracks only finds wait-depart decisions that are very close to optimal. Furthermore, the model without station capacities is much simpler and solving it requires less computation time. When deciding on the wait-depart decisions, one thus need not consider the capacity within the stations explicitly.

4.6 Analyzing the trade off between passenger delays and platform track changes

In Section 4.5.2 we have seen that allowing a re-assignment of trains to platforms yields a significant improvement with respect to the passengers' delay, compared to the model where the platform assignment is fixed. On the one hand, this certainly increases the passengers' comfort. However, on the other hand, the improvement with respect to the delay is accompanied by many platform track changes with respect to the announced timetable. These platform track changes are certainly not convenient for the passengers. In that respect, the solutions from the iterative algorithm, with a worse objective value but less platform track changes, might be preferable in practice. Delay management with station capacities could hence be considered a bi-objective problem with the two objectives of minimizing

1. the passengers' delay, and
2. the number of changes in the platform assignment.

In Section 4.6.1 we discuss how this extension of our model can be integrated in the proposed solution approaches. Computational results are presented in Section 4.6.2.

4.6.1 Theoretical Modification of the models

We first consider the exact model for delay management with station capacities which is provided by the assignment-based integer programming formulation. We describe how a restriction on the number of platform track changes can be easily included.
Since in this formulation the information about the platforms where an event $i \in \mathcal{E}$ takes place is already coded in the variables y_{ip}, a restriction on the number of platform track changes can be modeled easily by adding *one* additional constraint. To this end, let C denote the maximal number of platform track changes and let $P(i)$ denote the set of *good platforms for event* i. If we want to count all platform track changes, $P(i)$ consists only of the platform track p_i where event i was scheduled initially. However, $P(i)$ could as well contain more platforms that are easily reached from p_i, *e.g.*, the track on the opposite side of the platform.
We add the following constraint to the integer program described in Section 4.3.2.

$$\sum_{i \in \mathcal{E}_{\text{dep}}} \sum_{p \notin P(i)} d_{ip} y_{ip} \leq C. \tag{4.24}$$

Here, d_{ip} is a weight that represents the importance of scheduling the departure event i on a good platform track. For example, d_{ip} could represent the number of passengers that have to move to another platform if the platform track is changed. Since y_{ip} takes the value 1 if event i is scheduled on platform track p and 0 otherwise, this constraint determines the weighted number of departure events which do not take place on a good platform track and restricts this number to C. This method is an example of the ϵ-constraint approach. If $d_{ip} = 1$ for all $i \in \mathcal{E}_{\text{dep}}$ and $p \notin P(i)$ and 0 otherwise, the summation in the left hand side just counts the number of departure events that are not scheduled at a good platform. This allows us to find all non-dominated solutions, as the number of platform track changes is discrete.

It is also possible to take into account the number of platform track changes in the iterative approach. When searching for a new platform assignment in the third step of the iterative algorithm, instead of minimizing the potential departure times only, we could additionally consider the number of platform track changes. However, since in the linear programming formulation (4.18)-(4.21) for the third step of the iterative approach we do not explicitly define an assignment to the platforms, for every candidate q_i^k we replace the variable η_i^k by a set of $|P_s|$ variables

$$(\eta_i^k)_p = \begin{cases} 1 & \text{if candidate } q_i^k \in \mathcal{C}_i \text{ is chosen and event } i \text{ takes place at platform track } p, \\ 0 & \text{otherwise.} \end{cases}$$

Then constraints (4.19-4.21) can be rewritten as

$$\sum_{p \in P_s} \sum_{k=1}^{|\mathcal{C}_i|} (\eta_i^k)_p = 1 \qquad \forall i \in \mathcal{E}_{\text{arr}}^s, \tag{4.25}$$

$$\sum_{i \in \mathcal{E}_{\text{arr}}^s} \sum_{k=1}^{|\mathcal{C}_i|} a_{il}^k (\eta_i^k)_p \leq 1 \qquad \forall l \in \{1, \ldots, |\mathcal{C}| - 1\},\ \forall p \in P_s \tag{4.26}$$

$$(\eta_i^k)_p \in \{0, 1\} \qquad \forall i \in \mathcal{E}_{\text{arr}}^s,\ \forall k \in \mathcal{C}_i,\ \forall p \in P_s. \tag{4.27}$$

In order to incorporate our second objective of minimizing the number of platform track changes, we could again add a constraint that restricts the number of platform track changes at each station. However, this requires a distribution of the C allowed platform track changes to the stations. Hence we include the minimization of the number of platform track changes in the objective function (4.28), considering a weighted sum of both objectives:

$$\min \sum_{i \in \mathcal{E}^s_{\text{arr}}} c_i \sum_{p \in P_s} \sum_{k=1}^{|\mathcal{C}_i|} q_i^k (\eta_i^k)_p + \lambda \sum_{i \in \mathcal{E}_{\text{dep}}} \sum_{k=1}^{|\mathcal{C}_i|} \sum_{p \notin P(i)} (d_i^k)_p (\eta_i^k)_p \tag{4.28}$$

where $(d_i^k)_p$ is set to 1 if we just want to penalize platform track changes but could also be modified to represent passenger weights or to penalize only the platform track changes of non-delayed trains. The parameter λ can be used to control the influence of the different objective functions on the new platform assignment.
It should be noted that the integer program (4.25)-(4.28) is not totally unimodular. However, it is shown for similar formulations that most optimal solutions will be integral (see Kroon (1990)). This suggests that the problem can be solved with an IP solver.

4.6.2 Computational Results

We have used the cases from Section 4.5.1 to compare the solution approaches that balance the delay on the one hand and the number of platform track changes on the other. For Cases A and C, we considered all 100 delay scenarios. For Cases B and D, we restricted ourselves to 30 delay scenarios to reduce the amount of computation time needed. We first present the results for the exact solution approach and then consider the iterative procedure.

In Figure 4.8, we have plotted the average delay for all passengers as a function of the

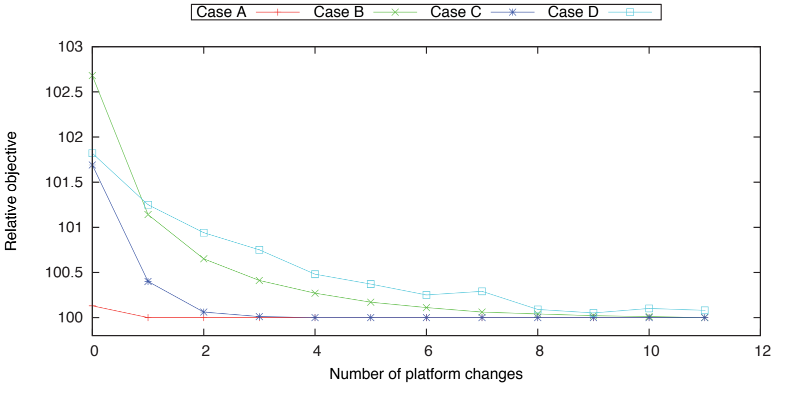

Figure 4.8: The average objective value as a function of the number of platform track changes that are allowed in the dynamic model.

maximal number of platform track changes. For Cases A and C, the optimal solution can be obtained with at most 3 platform track changes for all delay scenarios. By allowing only 1 platform track change, the solutions for Case A are optimal. For Case C, the optimal solution with at most one platform track change are within 0.4% of optimality on average. For Case B, 9 platform track changes are needed to obtain the optimal solution. Again, more than half of the delay reduction can be obtained with only one platform track change. For Case D, the results look somewhat different. We again find the optimal solution with only 9 platform track changes. However, for Case D, the delay reduction with only one platform track change is relatively small. Furthermore, we observe that the solution with at most 10 platform track changes has a worse objective value than the solution with at most 9 platform track changes. Recall from Section 4.5.2 that the dynamic model cannot be solved to optimality for Case D. When we limit the number of platform track changes, the average optimality gap equals 0.5%. This explains the small increase in objective value when increasing the number of platform track changes from 9 to 10. We think the aberrant progress for Case D is also caused by our inability to solve the integer program to optimality.

In general, in order to find the optimal solution, much less platform track changes are needed than were found by the dynamic model. Furthermore, the graph shows that a large part of the delay reduction can be obtained with only a small number of platform track changes.

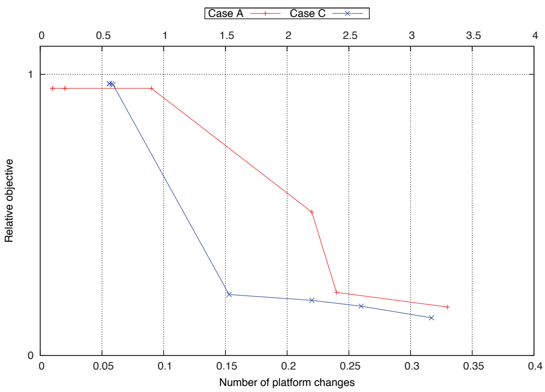

Figure 4.9: The average objective value and number of platform track changes for various values for the parameter λ. The lower x-axis corresponds to Case A; the upper x-axis to Case C

In Figure 4.9, we have plotted the results for the iterative algorithm for Cases A and C. We have executed the iterative algorithm for each value of the parameter $\lambda \in \{0.5, 1.5, 2.5, 4.5, 9.5, 19.5, 39.5

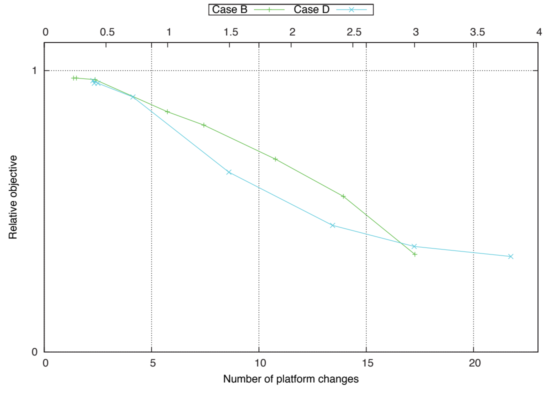

Figure 4.10: The average objective value and number of platform track changes for various values for the parameter λ for Cases B and D

For each value of λ, we ran the iterative algorithm and computed both the total passenger delay and the number of platform track changes for each delay scenario. The total delay is normalized in the same way as in Figure 4.7. The averages of the total delay and platform track changes are plotted in the figure. Similarly, in Figure 4.10, the results are depicted for Cases B and D.

We see in these figures that the number of platform track changes can be reduced by incorporating them in our iterative algorithm. For low values of the parameter λ, we obtain the same objective value as with the original iterative algorithm, but find less platform track changes. When the value of λ is increased, the number of platform track changes is reduced. However, this comes at the cost of more passenger delay.

For Case A, we find three interesting classes of solutions. The first class, found with $\lambda \geq 4.5$, resembles the static solution. Solutions in the second class, obtained for $\lambda \leq 1.5$, have an objective value that is comparable to that of the dynamic model, but introduce only few platform track changes. For $\lambda = 2.5$, a solution is found that balances the objective and the number of platform track changes. Note that for all values of λ, the average number of platform track changes is smaller than 1. This indicates that for most delay scenarios, there are no platform track changes at all. For Case C, only two classes of solutions are distinguished. For $\lambda \geq 9.5$, solutions are found with the same objective as the static approach. For $\lambda \leq 4.5$, we find an objective value that is close to the optimal objective, but the number of platform track changes is reduced significantly. For Cases B and D, the progress is more gradual. For all intermediate values of λ, a solution is found with a unique balance between the objective and the number of platform track changes.

Comparing both solution approaches for the bi-objective problem, we see that the exact algorithm finds solutions with less platform track changes. In terms of quality, the exact approach is thus preferable.
For Cases A and C, both the exact method and the iterative approach solve the instances within one minute. For Case B, the running time for the exact algorithm is in the order of 10 minutes, while the iterative approach can solve the model within 3 minutes. For practical applications, when computational time is scarce, it is thus better to apply the iterative algorithm. Finally, for Case D, both methods require 20 minutes of computation time. Such running times are too long for practical applications.

4.7 Conclusion and Further Research

In this chapter, we introduced a delay management model that incorporates the limited capacity of railway stations. Two models are presented. The first model fixes the assignment of trains to platforms and reduces to a delay management model with headway constraints. In the second model it is allowed to reschedule the platform assignment. In a computational study, we show that the delay for the passengers can be reduced when the platform assignment is rescheduled dynamically.
As solutions to the delay management problem should be available within a very short computation time, we also proposed an iterative solution method for the delay management problem with station capacities. This heuristic iterates between solving a delay management problem with a given platform assignment and optimizing the platform assignment given the timetable and wait-depart decisions. We show that for each station separately, an optimal platform assignment can be found in polynomial time. Computational tests show that the iterative heuristic can be applied to improve on a solution that is obtained by the static delay management model, especially for cases that include regional trains.
A drawback of the dynamic model is that it reschedules a lot of trains to other platforms in order to reduce the total delay. Although delays are a source of frustration for the passengers, many platform track changes are frustrating, too. Furthermore, these track changes put pressure on the dispatching organization of the railway operator. In our view, delay management with station capacities should therefore be viewed as a bi-objective optimization problem. We show that much of the delay reduction can be obtained by allowing only a few platform track changes. To resolve the remaining delays, many platform track changes are required. We therefore propose to limit the number of platform track changes that are allowed, in order to balance the delay for the passengers on the one hand and

the number of platform track changes on the other.
We distinguish two directions for future research. The first direction searches for faster solution methods. Although our model can solve real-world instances within computation times that are allowed in practice, solution methods for even larger instances might be required. Furthermore, we approximate the delay for passengers who miss a connection by the cycle time T. In reality, these passengers will probably select an alternative route. To cope with this problem, our model can be used in an iterative solution approach as proposed in Chapter 3, but then it should be solved several times. In such a setting, faster solution methods are necessary. We think that further attempts to solve delay management with station capacities heuristically could be developed that make use of relaxation-based solution approaches.
The second direction applies our methods to the timetabling problem. Station capacities are also an important issue in timetabling. Therefore it would be interesting to apply our exact and heuristic solution methods to the timetabling problem with station capacities and to compare them to existing solution approaches.

Chapter 5

An iterative optimization framework for delay management and train scheduling

5.1 Introduction

Most regular train passengers will recognize the frustration of missing a connecting train when their feeder train arrives at the transfer station with a small delay. In low-frequency railway systems, missing a connection can have a severe impact on the travel time of the passengers, even if the delay of the incoming train is only small. In such cases, an alternative would be to delay the departure of the connecting train, so that passengers from the delayed train can transfer to the connecting one. If a train waits for passengers from a delayed feeder train, the punctuality will be reduced; if it does not wait, passengers need to wait for the next service to their destination. Determining whether a train should wait for a delayed feeder train or should depart on time is the subject of Delay Management (DM). Netherlands Railways, the largest passenger operator in the Netherlands, endorses the importance of a reliable railway system and has recently introduced the passenger punctuality as a new performance indicator. The passenger punctuality measures the ratio of passengers who arrive at their destination with a delay smaller than a certain threshold value.

We propose in this chapter an innovative approach that computes a connection plan that solves the DM problem, and considers the limited capacity of the railway infrastructure. This latter is modeled as the Train Scheduling (TS) problem at a microscopic level, i.e. modeling the status of the signals and safety system. In our optimization framework the DM solution and TS solution iteratively set boundary conditions for each other. By coupling the two models, a solution is found that is locally feasible. Furthermore, by iteratively solving the DM and TS problems, delays for trains and passengers are reduced. The objective is multi-fold: (i) the computation of a feasible train schedule inside the stations, (ii) the minimization of train delays in station areas, (iii) the minimization of travel times for passenger flows at the network level.

We now review the main contributions on the DM and TS problems. In Schöbel (2001), a first integer programming formulation for the DM problem is given. This formulation is further developed in De Giovanni et al. (2008) and Schöbel (2007). In these models, it is assumed that passengers will wait for one cycle time whenever they miss a connection. In Chapter 2, we relaxed this assumption and assumed that passengers take the fastest route to their destination. We presented an integer programming formulation that allows for passenger rerouting and show that the delay is reduced significantly with respect to earlier models. In Chapter 3, we developed several heuristics to solve the DM problem with passenger rerouting.

Other extensions of the classical DM model incorporate the limited infrastructure capacity. Schöbel (2009) proposes to apply headway constraints to model the limited capacity on the tracks. An integer programming formulation that includes these headway constraints and several computational tests are given in Schachtebeck and Schöbel (2010). In Dollevoet et al. (2011), a first attempt to take the limited number of platforms in a station into account is presented. This approach was further developed in Chapter 4.

The DM models described so far are all macroscopic. The detailed characteristics of the railway infrastructure are abstracted to make sure that large parts of the network can be considered at once. Such a global scope is necessary for DM, as the passengers travel through large parts of the network. However, as a consequence, some of the complications arising from the infrastructure layout cannot be taken into account.

On the contrary, the train scheduling (TS) problem is to compute precisely the effects of delay propagation and the adjustments of train speed profiles at a microscopic level, by considering the capacity of the infrastructure and the behavior of the signaling system. This requires the definition of a microscopic scheduling problem, in which detailed information about the tracks and the switches is taken into account. This way, all characteristics of the infrastructure can be modeled.

Simulation models (see e.g. Hansen and Pachl (2008) for an overview) proved to be a suitable tool to represent the dynamics of train operations, but they are still limited especially when large stations and heavy traffic are considered, and are based on myopic rules that might result in large delays.

Concerning the optimization models for the TS problem, Törnquist (2012) resorts on heuristic procedures for computing schedules in a short time, compatible with operations. To this end, microscopic detail is considered for the most complex stations. Studies on a test case in Sweden report that for a time horizon of traffic prediction of 90 minutes, a feasible schedule is found within 30 seconds, even for instances where commercial optimization suites fail in finding a solution.

A fully microscopic model is used in Corman et al. (2011) to model train traffic over a complex and dense area of the Dutch railway network, with up to hundreds of trains. A truncated branch and bound procedure (D'Ariano et al., 2007) is used that achieves very often optimal solutions, substantially reducing delay propagation, compared to practice or simple dispatching rules. Building on that result, a bi-level programming is introduced in Corman et al. (2012) that allows control over very large instances, divided into many local areas. A coordination level is in charge of defining constraints at the border of the local areas to ensure a feasible global solution. The bi-level formulation allows to check feasibility and optimality at local and global network levels, leading to a branch and bound procedure that achieves quickly a good solution for up to one hour of traffic prediction.

Only recently, the DM problem has attracted some attention in the train scheduling literature. In Corman et al. (2010b), a bi-objective TS model is developed that minimizes the delay of the trains on the one hand and the number of missed connections on the other hand. However, as only the connections and trains within a station area are considered, the global behavior of the passenger flows cannot fully be captured.

For the existing TS approaches, the size of instances solvable within a real-time computation horizon is still smaller (in time horizon or geographical extent) compared to the macroscopic DM models. Moreover, typical objectives of TS models regard the reduction of (possibly weighted) delays and delay propagation, and generally exclude passenger flows. Inclusion of continuous passenger flows would increase further the complexity by taking into account multiple objectives.

In this chapter, we present an iterative optimization framework based on DM and TS models. It closes the gap between the theory on DM on the one hand, and on TS on the other hand. In doing so, the global scope of DM is combined with the high level of detail from TS. This way, we can model both the passenger flows at a network level and the detailed infrastructure locally at the stations. To the best of our knowledge, this is the first attempt to consider both levels in an integrated approach.

In a macroscopic DM model, we first determine which connections to maintain and derive the departure and arrival times of trains at the stations. Given the connections that should be maintained, these departure and arrival times are then validated using a microscopic TS model. Given the outcome of this microscopic validation, the process is repeated until a feasible solution is found. Doing so, we find solutions to the DM problem that respect the limited capacity of the station infrastructure, even for some of the most complicated and densely occupied stations in The Netherlands.

The remainder of this chapter is organized as follows. First, Section 5.2 describes the macroscopic DM model and Section 5.3 the microscopic TS model. Section 5.4 gives an illustrative example for both models. Then, Section 5.5 shows how these models are coupled in our iterative optimization framework. Section 5.6 reports the experimental setup to evaluate the framework. Section 5.7 concludes with remarks on the framework and on the computational results. Further research directions are also outlined for practical applications of the proposed methodology.

5.2 Delay management model

The central question of the DM models is which connections to maintain in case the railway system faces delays. It is assumed that the original timetable and the passenger demand are known. The passenger data is represented as a set of origin-destination pairs (OD-pairs) $\mathcal{P}$. Each OD-pair $p \in \mathcal{P}$ represents a group of n_p passengers who want to travel from a common origin station to a destination station at a specified time. Given a set of initial delays, the aim is to determine for each connection whether it should be maintained or not. Besides, a so-called disposition timetable is determined that prescribes the expected departure and arrival times of the trains at each station. Finally, for each OD-pair we determine a *passenger route* that connects their origin and destination, possibly including transfers at intermediate stations.

The DM problem is commonly modeled with an event-activity network. In this directed graph, the nodes correspond to the *events* that have to be scheduled and are denoted by $\mathcal{E}$. We distinguish between departure events $\mathcal{E}_{\text{dep}}$ and arrival events $\mathcal{E}_{\text{arr}}$, that correspond to the departure from and the arrival at a station, respectively. For each event $e \in \mathcal{E}$, we denote the time when the event is planned to take place by π_e. The variables π thus denote the timetable as it was planned to be operated. For each event $e \in \mathcal{E}$, the initial delay is denoted by d_e.

The arcs in the graph, denoted by $\mathcal{A}$, represent precedence constraints (or *activities*) between these events and ensure that a minimal time between the events is respected. We distinguish between driving arcs, waiting arcs and changing arcs. Driving arcs in $\mathcal{A}_{\text{drive}}$ connect a train's departure from one station to its arrival at the next station. Waiting arcs connect the arrival of a train at a station to its departure from that same station and make sure that time is available for the passengers to get off and on the train. We denote the set of waiting arcs by $\mathcal{A}_{\text{wait}}$. Finally, changing arcs, contained in $\mathcal{A}_{\text{change}}$, allow passengers to transfer from one train to another. Driving and dwell arcs correspond to operational constraints that cannot be neglected. On the contrary, transfer arcs model possible transfers for the passengers. In case of delays, the railway operator can decide to not maintain a transfer. For each activity $a \in \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{change}}$, we denote the minimal time required for that activity by L_a.

In order to compute the delay for the passengers correctly, We proposed to determine a passenger route for each OD-pair explicitly in Chapter 2. In order to do so, the event-activity network is extended with auxiliary events and activities. For each OD-pair $p \in \mathcal{P}$, both an origin event $Org(p)$ and a destination event $Dest(p)$ are added to the event-activity network. These auxiliary events act as a source and a sink in a unit flow problem. The origin event is connected to the departure events from the station where the passengers in p want to start their trip. Similarly, all arrivals at the destination station are connected to the destination event. The set of origin and destination arcs for OD-pair $p \in \mathcal{P}$ are denoted by $\mathcal{A}_{\text{origin}}(p)$ and $\mathcal{A}_{\text{destination}}(p)$, respectively. For notational convenience, we define

$$\mathcal{A}(p) = \mathcal{A}_{\text{drive}} \cup \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{change}} \cup \mathcal{A}_{\text{origin}}(p) \cup \mathcal{A}_{\text{destination}}(p).$$

In the extended event-activity network, a possible passenger route corresponds to a unit flow from the origin event to the destination event.

We are now ready to present an integer programming formulation for the DM problem with passenger rerouting. The main decision is which connections to maintain. We therefore introduce a binary decision variable

$$z_a = \begin{cases} 1 & \text{if connection } a \text{ is maintained,} \\ 0 & \text{otherwise.} \end{cases}$$

For each event $e \in \mathcal{E}$, we determine the actual time x_e that is in general equal to the planned time π_e plus a delay; the set x_e thus defines the disposition timetable.

For each OD-pair $p \in \mathcal{P}$, we determine a passenger route through the event-activity network. This corresponds to determining a unit flow from the origin event $Org(p)$ to the destination event $Dest(p)$ in the event-activity network. To model this, we introduce for each activity $a \in \mathcal{A}(p)$ a binary decision variable

$$q_{ap} = \begin{cases} 1 & \text{if OD-pair } p \text{ uses activity } a, \\ 0 & \text{otherwise.} \end{cases}$$

For each OD-pair $p \in \mathcal{P}$, we introduce an auxiliary variable T_p that represents the arrival time for passengers in OD-pair p.
The integer program then reads as follows (see Chapter 2).

$$\min \sum_{p \in \mathcal{P}} n_p T_p \tag{5.1}$$

such that

$$\begin{aligned}
x_e &\geq \pi_e + d_e, && \forall e \in \mathcal{E}, && (5.2)\\
x_e &\geq x_{e'} + L_a, && \forall a = (e', e) \in \mathcal{A}_{\text{wait}} \cup \mathcal{A}_{\text{drive}}, && (5.3)\\
M(1 - z_a) + x_e &\geq x_{e'} + L_a, && \forall a = (e', e) \in \mathcal{A}_{\text{change}}, && (5.4)\\
q_{ap} &\leq z_a, && \forall p \in \mathcal{P}, a \in \mathcal{A}_{\text{change}}, && (5.5)\\
1 &= \sum_{a \in \delta^{\text{out}}(Org(p))} q_{ap}, && \forall p \in \mathcal{P}, && (5.6)\\
\sum_{a \in \delta^{\text{in}}(e) \cap \mathcal{A}(p)} q_{ap} &= \sum_{a \in \delta^{\text{out}}(e) \cap \mathcal{A}(p)} q_{ap}, && \forall p \in \mathcal{P}, e \in \mathcal{E}, && (5.7)\\
\sum_{a \in \delta^{\text{in}}(Dest(p))} q_{ap} &= 1, && \forall p \in \mathcal{P}, && (5.8)\\
T_p &\geq x_e - M(1 - q_{ap}), && \forall a = (e, Dest(p)) \in \mathcal{A}_{\text{destination}}, && (5.9)\\
x_e &\in \mathbb{N}, && \forall e \in \mathcal{E}, && (5.10)\\
z_a &\in \{0, 1\}, && \forall a \in \mathcal{A}_{\text{change}}, && (5.11)\\
q_{ap} &\in \{0, 1\}, && \forall p \in \mathcal{P}, a \in \mathcal{A}(p), && (5.12)\\
T_p &\in \mathbb{N}, && \forall p \in \mathcal{P}. && (5.13)
\end{aligned}$$

The objective function (5.1) is to minimize the weighted sum of the passengers' arrival times. The planned arrival times are fixed, so this is equivalent to minimizing the average or total passenger delay. Constraints (5.2) incorporate the initial delays and make sure that no train departs earlier than planned. Constraints (5.3) propagate the delay along driving and waiting activities. For maintained connections, Constraints (5.4) propagate the delay from the arriving to the departing train. Constraints (5.5) make sure that a connection can only be used by passenger if it is maintained. Constraints (5.6)-(5.8) determine a unit flow from the origin event $Org(p)$ to the destination event $Dest(p)$, for each OD-pair $p \in \mathcal{P}$. Here $\delta^{\text{in}}(e)$ and $\delta^{\text{out}}(e)$ denote the set of arcs into and out of node $e \in \mathcal{E}$, respectively. Finally, Constraints (5.9) linearize the arrival times of the passengers. In Constraints (5.4) and (5.9), the parameter M is a sufficiently large number. We refer to Chapter 2 for more details on the integer programming formulation.

5.3 Train scheduling model

Given the actual train delays, the train scheduling problem is to compute a new feasible schedule compatible with the status of the network, with the signaling system, and the dynamics of trains. Potential conflicts between train paths are detected by a conflict detection procedure for a given period of traffic prediction. In case of fixed block signaling, tracks are divided into block sections; each block section cannot host two trains at the same time. A potential conflict occurs whenever two or more trains require the same block section and a decision on the train order has to be taken. The train that will traverse the block section as second will be held outside the block section by the signaling system. In fact, while this train approaches the occupied block section, first a yellow signal will be shown, prescribing to slow down to an approaching speed (e.g. 40 km/h); and finally the signal just before the block section will show a red signal that prescribes a complete stop before the block section, as long as the preceding train has not exited the block section and a minimum setup time has elapsed. A set of ordering decisions might furthermore result in a deadlock. A deadlock is the situation in which a set of trains is mutually waiting for a train in the set to move, and no movement for the trains in the set is possible.

To model those situations, a microscopic model is required, that has a precision of seconds in modeling the travel times and considers train movements at the level of block sections. This is the level of detail required to model properly the triggers of the safety system and represent the signal aspects of the signaling system. The final outcome is a detailed schedule of train movements, without deadlock situations, where all potential conflicts have been solved. In this way, precise times can be predicted and delays are estimated accurately.

We use a job shop scheduling model of the TS problem that can be represented as an event-activity network with additional constraints. Mascis and Pacciarelli (2002) show that this so-called alternative graph is a suitable model for the job shop scheduling problem with additional constraints, such as blocking, also occurring in the railway context. The main value of this formulation is the detailed representation of the train traffic, the network topology and the signaling system.

This formulation requires that a sequence of successive block sections is defined for each train. The time required by each train to traverse each block section can be computed in advance, except for a possible additional waiting time between operations in order to solve train conflicts. In the alternative graph model, this results in a chain of *operations* (passage of a train on a block section, modeled by nodes $n \in N$) and associated precedence constraints (modeled by fixed arcs in Fix), similarly to the event-activity network of the DM problem.
For every potential conflict, a passing order must be defined between the trains, which is modeled in the graph by introducing a suitable pair of alternative arcs (in the set Alt) for each pair of trains traversing a block section, that define each of the two possible orders between the trains. Those arcs result in minimum headways between different trains, according to the signaling system.
A deadlock-free and conflict-free schedule is finally obtained by selecting one alternative arc from each pair, and updating the speed profile of the trains to the actual aspects of the signaling system (see Corman et al. (2011)). Formally, the TS problem corresponds to a particular *disjunctive program*, i.e., a linear program with logical conditions involving operation "or" ($\vee$, disjunction), as follows.

$$\min t_n - t_0 \tag{5.14}$$

such that

$$t_j - t_i \geq w_{ij}, \qquad (i,j) \in Fix, \tag{5.15}$$

$$(t_j - t_{\sigma(i)} \geq w_{\sigma(i)j}) \vee (t_i - t_{\sigma(j)} \geq w_{\sigma(j)i}), \qquad ((\sigma(i), j), (\sigma(j), i)) \in Alt. \tag{5.16}$$

In Problem (5.14)-(5.16), a variable t_i, for $i = 1, \ldots, n-1$, is the start time of operation i and corresponds to the entrance time of a train in the associated block section, similar to x_e in the DM model. We use $\sigma(i)$ to refer to the successor of operation i on the route followed by a particular train, i.e. the operation on the block section after i. Moreover, operation 0 is a dummy operation that precedes all the other operations, to give a common temporal reference; and operation n is a dummy operation that follows all the other operations, and is used to keep track of delays, as explained later. In the scheduling model, all t_i are expressed in seconds, while the precision of x_e in the DM model is in minutes.
Fixed constraints in Fix are a general family of constraints associated to characteristic processes of railway operations, as follows.

- Running constraints naturally define a chain of driving operations between operation i of a train, and its successor $\sigma(i)$ on the path followed by the train. For such driving process, we consider precedence relations of the form $t_{\sigma(i)} \geq t_i + w_{i\sigma(i)}$, where $w_{i\sigma(i)} > 0$ is the time required to traverse the block section associated to that operation, at its actual speed profile.

- Dwell constraints at a station model the boarding and alighting of passengers, where $w_{i\sigma(i)}$ is the minimal time required between the arrival operation and the departure operation of the same train.

- Release constraints of the form $t_i - t_0 \geq w_{0i}$ relate to operation 0 and represent minimal start time for operation i, i.e. model the entrance time of a train into the area. This is analogous to the π_e in the DM model.

- Due date constraints of the form $t_n - t_i \geq w_{in}$ relate to operation n and represent a due date for operation i. Such constraints are used to compute the delay associated to train traffic.

- Connection constraints, as defined in the DM problem, fix the departure time of a connected train to be larger than the arrival of a feeder train, plus a given minimum connecting time. These constraints are the changing constraints specified by the DM problem (the variables z_a). Such connections are normally associated to an arrival event of a train at a station platform, and a successive departure of another train at another platform of the same station.

Differently, the set *Alt* is disjunctive, i.e., is composed of pairs of alternative constraints, each of them representing an ordering decision between trains. For each pair i and j of operations associated with the entrance of two trains in the same block section, we introduce the disjunction $(t_j - t_{\sigma(i)} \geq w_{\sigma(i)j}) \vee (t_i - t_{\sigma(j)} \geq w_{\sigma(j)i})$, where $w_{\sigma(i)j} > 0$ and $w_{\sigma(j)i} > 0$ are the minimum headway times. Those headway times are a function of a variety of factors, such as the length of the block section, the speed profile of the train, the driver behavior, and the length of the train, as specified by the blocking time theory (see e.g. Hansen and Pachl (2008)). Finally, running and headway times are a function of the speed profiles of trains, that again depend on the ordering decisions taken. The solutions computed are fully compliant with the operational rules, the dynamics of trains, and the actual signal aspects shown.

A TS solution corresponds to fixing the start time of each operation. The schedule is feasible if it satisfies all conjunctions in Fix and exactly one constraint for each disjunctive pair in *Alt*, and does not result in positive length cycles. Due to the structure of the arcs

(i, n), the (positive) train delay can be computed at a set of relevant points (scheduled stops and the exit of the network). It is interesting to consider the consecutive delay only, i.e. the delay introduced when solving conflicts in the dispatching area, caused by the propagation of the initial delays of late trains to the other trains in the railway area. The objective function of the TS problem is the minimization of the maximum consecutive delay, that corresponds to the length of the longest path between the dummy nodes 0 and n, i.e. $t_n - t_0$.

5.4 Illustrative example

Figure 5.1 gives an illustrative example of the two models of Section 5.2 and 5.3. In the top part of Figure 5.1, two trains V and T are running on a line connecting station P with station Q. Train T stops at both stations, while train V stops only at station Q; thus, at this latter station there is a possibility to enforce a connection between the two trains. The dotted line defines the station area, i.e., a region in which switches connect different tracks, that merge and cross each other. In fact, train T follows the lower path in the network, while train V follows the upper path in the network; both are using the block section b just before station Q. To ensure minimum train separation and safe movements over the network, the fixed-block signaling system is used.
The middle part of Figure 5.1 refers to the macroscopic model used for the DM problem. Events are represented as nodes, and activities in the set $\mathcal{A}$ as arcs connecting them; train V is represented as the upper chain of events (including arrival event $A3$ and departure event $D3$), and train T as the lower chain (including $A1$, $D1$, $A2$, $D2$). More in detail, the graph shows a $Wait$ activity at station P for train T, a *Drive* activity between station P and station Q for train T, and waiting activities at station Q for both trains (reported as $Wait3$ and $Wait2$). A connection activity in $\mathcal{A}_{\text{change}}$ is also considered, resulting in the arc labeled *Connection* $3 \to 2$.
The bottom part of Figure 5.1 considers instead the microscopic model as used for the TS problem, only for the area around station Q. The trains considered in the example define two chain of nodes and arcs (again, the upper chain for train V and the lower one for train T), plus the two dummy nodes 0 and n. Successive nodes of each train are connected by arcs representing *Run* activities, plus the two *Dwell* activities at station Q. The ordering decision on the block section b is modeled by a pair of alternative, dotted arcs, representing the two possible orders between trains. The same connection constraint (*Connection* $3 \to 2$) as in the DM model is included, constraining train T not to leave station Q before train V has arrived and a minimal time has passed. There are four

release arcs, that connect dummy node 0 to the first node of a train, representing the entrance time of the train in the area considered, and to the departure from the stations, modeling the published departure time. Finally, two due date arcs connect the exit from the area considered to dummy node n.

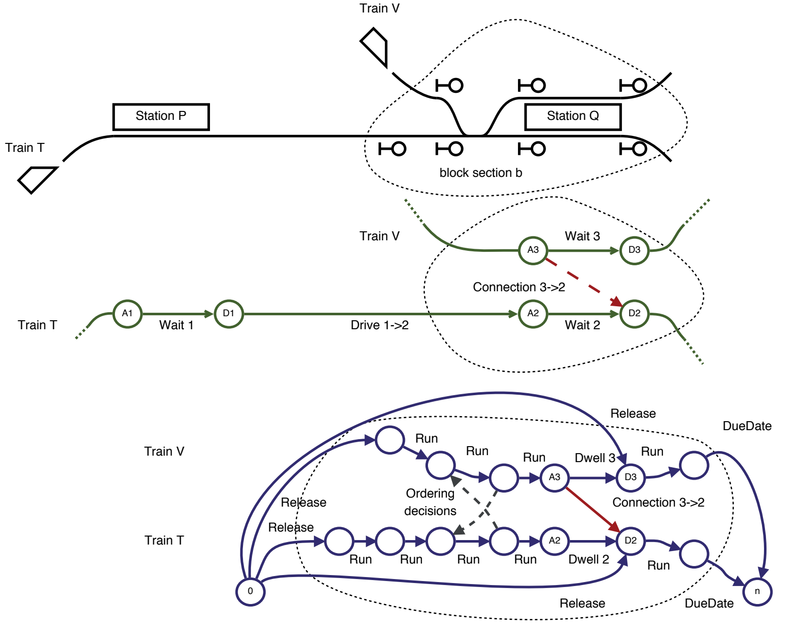

Figure 5.1: (top) Network of the illustrative example; (center) Macroscopic model used in the DM problem; (bottom) Microscopic model used for the area of station Q, in the microscopic model for the TS problem

5.5 Iterative DM and TS optimization approach

The previous sections presented the DM and TS models individually. We now introduce the optimization framework that iterates between solving a macroscopic DM problem on the one hand and a microscopic TS problem on the other. We will first give a general overview of the combined system and then an example is presented.

A schematic outline of our optimization framework is presented in Figure 5.2. The original timetable and the passenger demands are used as input for the algorithm. The passenger demand is given as a set of OD-pairs $p \in \mathcal{P}$, each of them representing n_p passengers

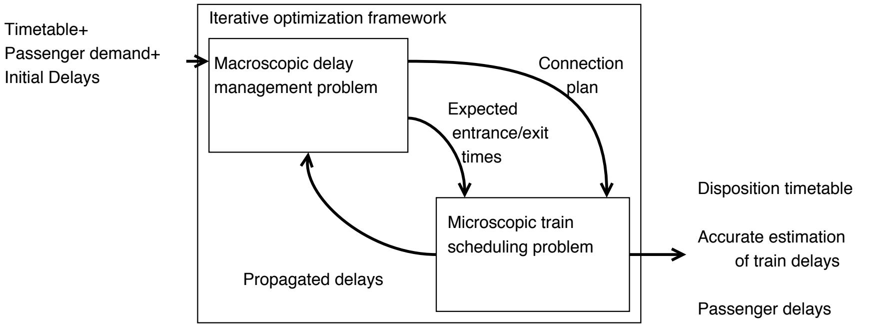

Figure 5.2: Schematic representation of the optimization framework.

who want to travel from an origin to a destination at a specified time. The timetable prescribes for each arrival and departure at a station at what time and at which platform it should take place. Furthermore, a set of initial delays is given. We assume that only the arrival events in the network have an initial delay. Equivalently, we assume that the initial delays are zero for all departure events.

The upper (macroscopic) part solves a DM problem to determine the connections to be maintained and computes a macroscopic disposition timetable. The DM solution minimizes the total delay for the passengers. In doing so, it allows the passengers to change their routes through the network.

The DM solution results in a set of passenger connections that should be maintained (i.e. a set of values for the variables z_a) and an expected macroscopic timetable (corresponding to a set of event times x_e for all events e). Those variables are used to define a TS problem. To this end, we focus on those stations in the network where the infrastructure capacity is a bottleneck, and the possibility of facing conflicts for the scarcely available infrastructure is the highest.

Each TS problem considers part of the railway network around a station, in order to represent most of the potential train conflicts. The release time for an arriving train $e \in \mathcal{E}_{\text{arr}}$ into the area is computed based on the expected arrival time x_e of that train in the DM solution, minus a fixed time τ_e that corresponds to the minimal running time between the entrance of the microscopic network and the arrival at the station platform. Similarly, we associate due dates to departing trains $e \in \mathcal{E}_{\text{dep}}$, based on x_e, the expected departure time computed by the DM solution, plus a time τ_e equal to the minimal running time from the platform until the exit of the microscopic network. The set of connections to be maintained, i.e. those for which $z_a = 1$, is also used in the TS problem. These

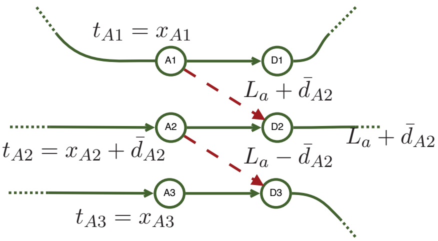

Figure 5.3: Part of the event-activity network within a station. $\bar{d}_{A2}$ is the extra delay computed by the TS model.

transfer activities are added as fixed arcs to the set Fix.
The solution to the TS problem is a set of starting times of all operations, that are feasible with regard to the signaling system and the dynamics of trains. In particular, the solution contains starting times for the arrival and departure events $e \in \mathcal{E}$, that are considered in the DM problem. We will denote these starting times by t_e for all $e \in \mathcal{E}$.
This updated plan of operations will in general have conflicts in the station area and propagate some of the delays. The actual arrival and departure times t are going to be different from those original times x considered in the DM model. We thus find additional delays $\bar{d}_e = t_e - x_e$ for each event $e \in \mathcal{E}$ that is considered in the DM problem of the next iteration. To take these deviations into account, we update the minimal duration of the process times L_a for activities $a \in \mathcal{A}_{\text{change}} \cup \mathcal{A}_{\text{drive}}$, while avoiding to explicitly fix variables in the DM model.
To explain how these additional delays are incorporated, consider a train that departs later from a station than it was expected in the previous iteration. In that case, more passengers are able to transfer to that departing train. Furthermore, the train will probably arrive later at the next station in the macroscopic network.
We explain how we incorporate these intuitive ideas using Figure 5.3 that refers to a DM model. Part of an event-activity network is shown, that contains the arrivals and departures of three trains (respectively, A1, D1; A2, D2; A3, D3). The diagonal lines connect events of different trains and represent possible transfers for the passengers. We assume that the solution computed by the TS model contains some propagated delays $\bar{d}_{A2}$ and $\bar{d}_{D2}$, i.e. the actual times t_{A2} and t_{D2} are different from the plan x_{A2} and x_{D2}, respectively. All other events e have $\bar{d}_e = 0$, i.e. they occur at their planned time x_e. There are two possible connections represented (between $A1$ and $D2$; and between $A2$ and $D3$). Recall that L_a denotes the minimal transfer time for a transfer activity $a = (e, e') \in \mathcal{A}_{\text{change}}$. This means that the connection is maintained if and only if $x_{e'} - x_e \geq L_a$.

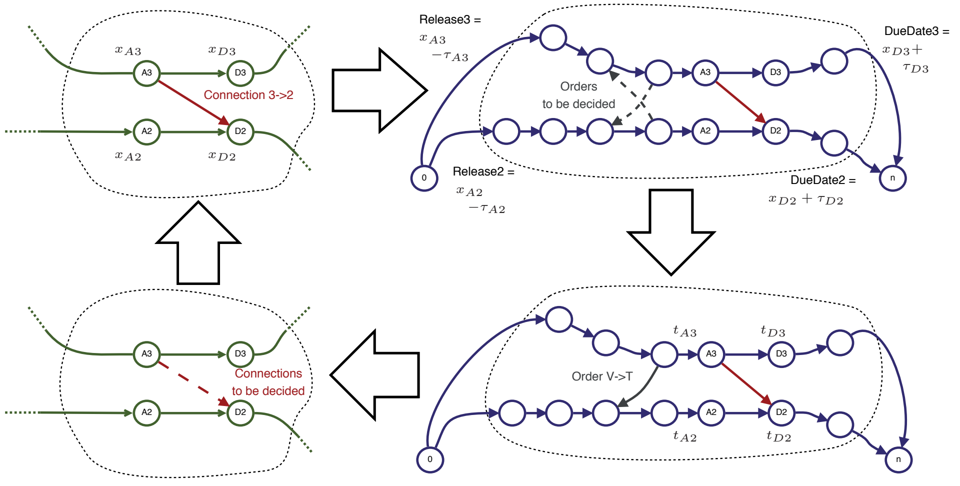

Figure 5.4: Iterative solution approach for the combined DM (left column) and TS (right column) problems. Iterations increase clockwise.

Our aim is now to anticipate the delays from the TS model in the DM model. In the microscopic timetable, the transfer time for passengers equals $t_{e'} - t_e$. Incorporating the delays from the TS model, we thus find that the connection is maintained, if and only if

$$L_a \le t_{e'} - t_e = x_{e'} + d_{e'} - x_e - d_e \quad \Leftrightarrow \quad L_a - d_{e'} + d_e \le x_{e'} - x_e.$$

This suggests to use $L_a - d_{e'} + d_e$ as the minimal transfer time in the next iteration. For the transfer to the delayed train (i.e. $A1 \to D2$), the transfer time is thus decreased by the propagation of delays, as more passengers will be able to transfer. For the transfer from the delayed train (i.e. $A2 \to D3$), the transfer time is increased by the amount of delay. Finally, for the driving activity that connects the departure $D2$ to the arrival at the next station, the minimal driving time L_a is increased with the amount of delay.

We next illustrate the steps graphically, referring to Figure 5.4. We start from the top-left of Figure 5.4, which shows a solution to the DM problem, corresponding to a decision to maintain connection $3 \to 2$, and a proposed disposition timetable computed at macroscopic level, that corresponds to expected arrival and departure times (respectively, x_{A3}, x_{D3}, x_{A2}, and x_{D2}) for the two trains of the example reported in Figure 5.1.

We use this solution to define a TS problem, in which only a station area is considered. This is reported in the top-right of Figure 5.4. The two trains enter the network at their release times ($Release3$ for the upper path corresponding to train V, and $Release2$ similarly for the lower path and train T), that are computed based on the expected arrival time (x_{A3} and x_{A2} respectively) and the fixed times τ_{A3} and τ_{A2} related to running between the entrance of the microscopic network and the arrival at the station platform. Similarly, due dates are computed based on the expected departure time (x_{D3} and x_{D2}) and fixed times τ_{D3} and τ_{D2} related to running time from the platform to the exit of the microscopic network.

The TS problem is to compute the times of each operation, and orders between trains on shared infrastructure elements, that are represented by alternative arcs. The connections defined by the DM solution are included in the TS problem as fixed arcs. A solution to the TS problem is shown in the bottom-right figure, showing the order $V \to T$ chosen (i.e. train V precedes train T on block section b). This defines a microscopically feasible arrival time of the trains at the platform (respectively t_{A3} and t_{A2}), and similarly feasible departure times from the platform (t_{D3} and t_{D2}, respectively).

We then use the microscopically feasible times of the TS solution to define a new instance of the DM problem in the bottom-left of Figure 5.4. In general there will be a difference between the actual times t and the expected times x that were considered at the previous iteration, as trains might face yellow or red signals to avoid potential conflicts. Those differences result in propagated delays, that define new process times for driving and changing activities. Based on these updated data, the DM solution might keep the same set of connections as in the iteration before, or choose for new ones. The resulting solution would be the one shown on the top-left of Figure 5.4, leading to another iteration.

5.6 Computational experiments

We assess the performance of our optimization framework using real-world instances from the Netherlands. Railway activities in the Netherlands are split between an infrastructure manager (ProRail) on the one hand and several railway operators on the other hand. We obtained detailed information on the infrastructure from ProRail and the timetable and passenger information from Netherlands Railways. Netherlands Railways is the largest passenger operator in the Netherlands and transports over a million passengers per day.

We now first describe the instances that we used to evaluate our optimization framework. Then we present the computational results. In all our experiments, our main objective will be to minimize the total passenger delay.

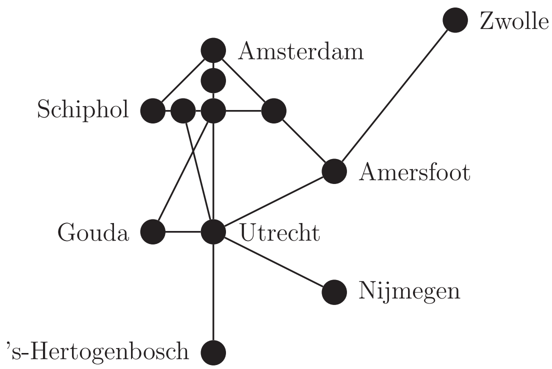

Figure 5.5: A large and busy part of the Dutch railway network

5.6.1 Instances

The instances consider the railway network that is depicted in Figure 5.5. This picture shows a dense part of the railway network that contains Utrecht Central Station, which is in the centre of the Netherlands. The dots in the picture represent larger stations, where passengers have the possibility to transfer from one train to another. Two stations are connected by a line if there is a direct train between them. On most lines, both long distance trains and regional trains are operated with a high frequency. The long distance trains stop at the stations in the picture only. On the contrary, regional trains stop on smaller stations along the line, too. In total, we consider 46 stations. Because there are both long distance trains and regional trains with a high frequency, the station infrastructure in major stations is utilized heavily, especially in Utrecht Central Station.
In order to assess the performance of the iterative approach, we generate a set of delay scenarios and solve the corresponding delay management problem with the proposed optimization framework. We generate two samples: one sample with small initial delays and one with large initial delays. Both samples contain ten scenarios. We have generated the delay scenarios as follows. In all scenarios, each arrival of a train at a station has a probability of 10% to be delayed. If an arrival is delayed, the initial delay is uniformly distributed between 1 and 5 minutes in the sample with small initial delays. Similarly, in the sample with large initial delays, the initial delay is uniformly distributed between 1 and 15 minutes.
In Table 5.1, we first present some characteristics of the resulting delay management problem. In total, 377 trains are considered. Together, these correspond to 1221 departure events, 1221 arrival events and 1221 driving arcs. Besides, there are 844 dwell arcs,

Characteristics of the macroscopic model	
Time horizon	4 hours
Stations	46
Trains	377
Train driving activities $\lvert\mathcal{A}_{\text{drive}}\rvert$	1221
Dwell activities $\lvert\mathcal{A}_{\text{wait}}\rvert$	844
Connections activities $\lvert\mathcal{A}_{\text{change}}\rvert$	9643
OD-pairs $\lvert\mathcal{P}\rvert$	7086

Table 5.1: Some characteristics of the delay management model

leading to 2065 operational activities. Furthermore, the network contains 9643 possible connections. We consider 7086 OD-pairs, of which 1732 have a direct train from their origin to their destination. This shows that 76% of the OD-pairs should transfer at least once. It turns out that OD-pairs with a direct trip attract much more passengers: Only 20% of the passengers in the railway network have to transfer.

Considering the microscopic validation of the solution of the DM model, we focus on the bottleneck of Utrecht Central Station, that is the station in which the infrastructure is used most heavily. In fact, five main lines arrive and depart from the 14 platforms of Utrecht Central Station, passing through two large interlocking areas at the sides of the station with a total of about a hundred switches. The TS model refers to a railway network that includes the station area of Utrecht Central Station, and about 10 kilometers of the railway lines, as in Figure 5.6.

The network considered results in train scheduling problems with the characteristics reported in Table 5.2. On top of the main station of Utrecht, 10 more minor stations are considered along the lines. Compared to the DM problem, for the same time horizon, only the trains passing through the area are considered; anyway, the microscopic detail leads to more individual operations considered, with about 22 operations considered for each train, on average. The amount of ordering decisions increases polynomially with the amount of trains running on the block sections, resulting in more than 52,000 variables defining the order of trains.

5.6.2 Results for instances with small delays

Typical behavior of the iterative optimization framework for instances with small delays is presented in Figure 5.7. This figure shows the objective value in each iteration for a single case. Along the vertical axis is the total delay for the passengers in minutes. The solid line gives a lower bound on the optimal objective value, obtained by solving the

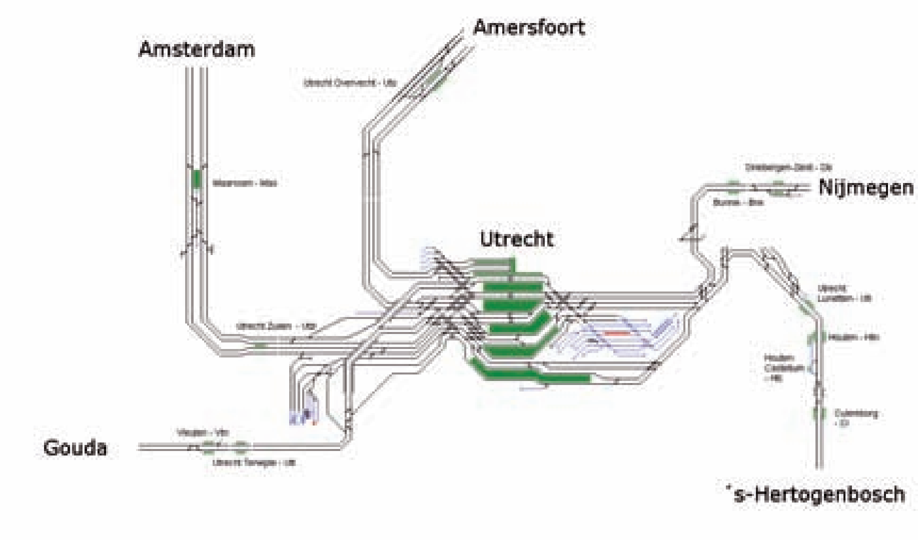

Figure 5.6: Microscopic detail in the area around Utrecht Central Station, and location of the borders.

Characteristics of the microscopic model	
Time horizon	4 hours
Stations	11
Block sections	531
Trains	257
Operations $\vert N\vert$	5681
Ordering decisions $\vert Alt\vert$	52600

Table 5.2: Some characteristics of the train scheduling model

delay management problem without considering the station capacity. The objective values from the DM model in each iteration are represented by asterisks and connected by the lower dashed line. Recall that the corresponding solutions are in general microscopically infeasible. In order to obtain feasible solutions, we apply the TS model, obtaining a set of consecutive delays for the trains in Utrecht Central Station. By propagating these delays through the network, we obtain a solution to the DM problem that is microscopically feasible. The objective value for this solution can be found by computing for each OD-pair the earliest arrival time and the corresponding delay. Adding these delays over all OD-pairs gives the objective value for this solution. These objective values are indicated

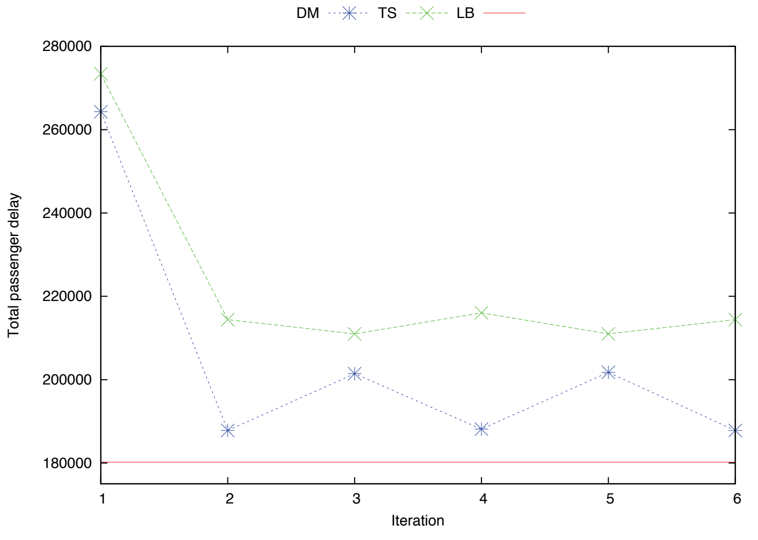

Figure 5.7: The total delay for the passengers in each of the iterations for a scenario with small delays

by the crosses in the figure.

We start the iterative approach with a solution in which no connections are maintained. In the second iteration, the possibility to maintain a connection is included and the solution value for the DM problem decreases significantly. However, the gap to the solution of the TS problem is rather large. In the next iteration, the consecutive delays found by the TS algorithm are anticipated, leading to a solution that is slightly better. From then onwards, the algorithm oscillates between two solutions.

The average objective values over 10 scenarios are presented in Table 5.3. In the second column we report the objective value that is found in a specific iteration. The iterative procedure does not improve the solution in every iteration. The third column therefore contains the best objective value that is obtained *until* that iteration. Finally, we present the best normalized passenger delays in the last column. As can be seen, in the second iteration the delay is reduced with 26% with respect to the first iteration. In the next iterations, the total passenger delay is reduced by another 1.0%. The best solution is found in the second iteration for 3 instances, in the third iteration for 5 instances and once in the fourth and sixth iteration.

Characteristics of the solution procedure for instances with small delays are reported in Table 5.4. Solutions to the DM problem can be found in 125 seconds on average. Solving

Iteration	Objective	Best objective	Best normalized objective
1	324734	324734	100
2	240836	240836	74.2
3	241027	238985	73.6
4	241205	238793	73.5
5	242023	238793	73.5
6	239075	237670	73.2

Table 5.3: The average objective value over 10 instances with small delays

Characteristics of the solution procedure	
Computation time for one DM iteration (seconds)	125
Computation time for one TS iteration (seconds)	240
Average consecutive train delay in Utrecht (seconds)	4.0
Gap between DM and TS solution (total passenger delay)	8%
Gap to the lower bound (total passenger delay)	15%
Difference between first and best solution (average train delay)	20%

Table 5.4: Some characteristics of the solution procedure for the instances with small delays

an instance of the TS problem takes on average 240 seconds of computation time. The resulting solutions have a maximum consecutive delay of 212 seconds and an average consecutive delay of 4 seconds. In the first iteration, the solution value after solving the TS problem is about 6% worse than the objective value from DM. In other iterations, the solution value is increased by about 9%. The gap between the final solution and the lower bound is on average 15%. Recall that the lower bound is obtained by solving the DM problem from Section 5.2 without considering the station capacity. We also compare the average train delay between the first iteration and the iteration in which the best solution is found. For instances with small delays, the average train delay is increased with 20%.

5.6.3 Results for instances with large delays

For the scenarios with larger delays, the algorithm behaves less consistently. In Figure 5.8, we show the solution values for an instance where the iterative approach improves over the first solution. Again, we start the process with a solution that maintains no connections. In the following three iterations, the solution value decreases. After that, worse solutions are found. Such behavior is observed for 40% of the scenarios.

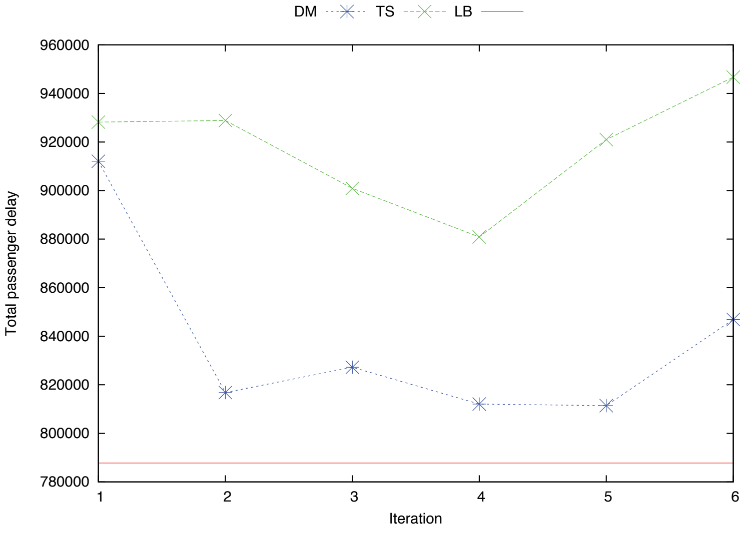

Figure 5.8: The total delay for the passengers for an instance with large delays where the iterative approach improves over the start solution

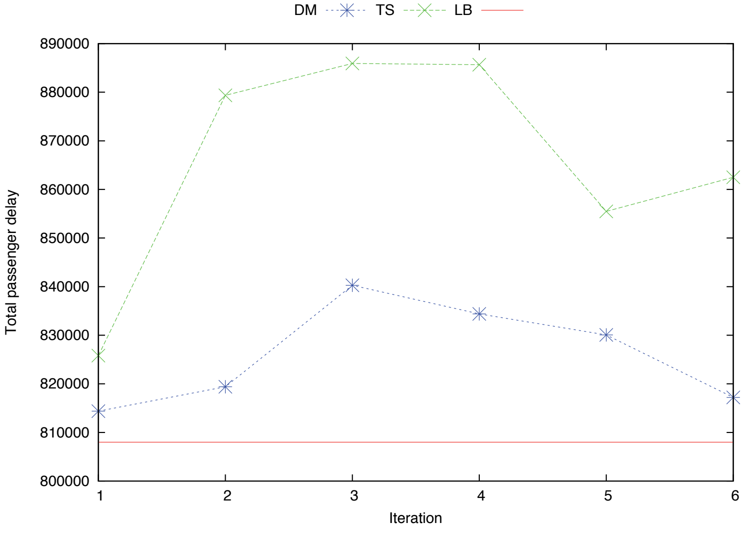

Figure 5.9: The total delay for the passengers for an instance with large delays where the iterative approach cannot improve over the start solution

Iteration	Objective	Best objective	Best normalized objective
1	900634	900634	100
2	903771	878029	97.5
3	907546	874899	97.1
4	892270	871588	96.8
5	897342	871588	96.8
6	909596	871588	96.8

Table 5.5: The average objective value over 10 instances with large delays

Characteristics of the solution procedure	
Computation time for one DM iteration (seconds)	-
Computation time for one TS iteration (seconds)	300
Average consecutive train delay in Utrecht (seconds)	8.7
Gap between DM and TS solution (total passenger delay)	10 %
Gap to the lower bound (total passenger delay)	13 %
Difference between first and best solution (average train delay)	21%

Table 5.6: Some characteristics of the solution procedure for instancs with large delays

For the other instances we find worse solutions after the first iteration. A typical example is given in Figure 5.9. The course of the solution values for these instances is very unstable. Furthermore, the iterative approach does not improve over the start solution.

In Table 5.5 we report the average objective values in each iteration. Only in the third and fourth iterations, the average objective value is better than that of the start solution. In the fourth column we report for each iteration the best relative solution value obtained until that iteration. Here we see that, on average, the solutions in the final iteration are 3.2 % better than the solutions found in the first iteration. The gap between the DM and TS solution is 4% in the first iteration and on average 11% in the other iterations.

In Table 5.6, some characteristics of the solution procedure are reported. Solving an instance of the train scheduling problem takes on average 300 seconds of computation time for large delays. The resulting solution has a maximum consecutive delay of 354 seconds and an average consecutive delay of 8.7 seconds. Solving the delay management problem to optimality takes much time. Therefore, we limit the time for the DM problem for each iteration to 20 minutes. Within this time, solutions are found that are close to optimal, with gaps smaller than 1% for all instances. Furthermore, the best solution is found within several minutes. Comparing the average train delay between the first and the best iteration, we observe an increase of 21%.

5.7 Conclusions

In this chapter we developed an iterative optimization framework for delay management and train scheduling. We propose a mechanism to incorporate consecutive delays from the train scheduling solution in the delay management problem. By combining the global scope of delay management and the local scope of train scheduling, we were able to find solutions to the delay management problem that respect the limited capacity of the station infrastructure. Besides these wait-depart decisions, the solution framework provides a feasible train schedule at the stations, where the infrastructure is used heavily. This train schedule allows for the precise evaluation of train delays, and thus also of the passenger delays.

We first consider scenarios with small initial delays. For those scenarios, our framework obtains a solution to the DM problem that is microscopically feasible. In the iterative optimization procedure, the delay for the passengers is reduced by 27% with respect to a naive approach where only one iteration is performed. For scenarios with larger delays, the behavior of the solution procedure is less consistent. However, we are able to compute and evaluate a solution to the delay management problem that is feasible at the station level.

Several directions for further research are available. First, the interaction between the models should be investigated in more detail for scenarios with large delays. Considering a more general feedback mechanism could potentially lead to better solutions. For example, one could define weights on the connections in the train scheduling model or penalize changes in the wait-depart decisions in the delay management model. Second, the iterative framework could be tested on a railway network with more bottlenecks. For each station where the infrastructure is scarce, a local scheduler could be applied to compute a feasible train schedule. As the updates from different station can be conflicting, these updates should be fully coordinated.

Chapter 6

Summary and Conclusions

In this thesis, we studied delay management and dispatching in railways. Delay management and dispatching deal with small delays that occur in the daily operations of a railway operator. When a delayed feeder train arrives at a station, delay management models decide whether connecting trains should either wait or depart on time. These wait-depart decisions should balance the delay for the transferring passengers on the one hand and the delay for passengers in the connecting train on the other. The aim is to minimize total passenger delay. Dispatching reschedules the infrastructure assignment of the trains in order to minimize the propagation of train delays. Especially in the areas around major stations, the available infrastructure is heavily utilized. Without proper rescheduling, a delay of one train may then propagate to many other trains. By changing the order in which trains enter and leave the station, the impact of such a train delay may be reduced.
In Chapters 2-4, we developed several models and solution methods for delay management. In Chapter 5 we make a first step towards integrating dispatching aspects into delay management. We now collect the main findings from these chapters and then suggest some directions for future research.

6.1 Main findings

In Chapter 2, we introduce *Delay Management with Rerouting of Passengers*. When a passenger misses a connection, he has to find an alternative way to reach his destination. Previous delay management models assume that passengers wait for a complete cycle time and take a similar train as the one they missed. In the Netherlands, this assumption is not realistic: On many lines there are more trains in each direction and most passengers will select the fastest of these alternatives to get to their destination. We therefore develop a delay management model that incorporates the routing decisions of the passengers. Assuming that all passengers select the fastest alternative, we show that the wait-depart decisions from our integrated model reduce the delays for the passengers with respect to a model without passenger rerouting. A drawback of the model with passenger rerouting is its complexity: The integer programs are much larger and therefore computationally harder to solve.

To cope with the complexity arising from the routing decisions, we develop heuristics for delay management with passenger rerouting in Chapter 3. Recall that models without rerouting generally assume that passengers wait for a complete cycle time when a transfer cannot be made. These models add a penalty of one cycle time to the total delay for each passenger that misses a connection. Instead of adding a fixed penalty that is equal to the cycle time, we propose a penalty that is different for each connection. The values of the penalties are updated in an iterative procedure. In a computational study, we show that this heuristic finds wait-depart decisions that lead to a delay very close to the minimal delay, in only a fraction of the time. Furthermore, we compare our models to the policies that are currently used in practice. These policies apply simple rules of thumb to decide whether a train should wait or not. We show that delay management models outperform such a rule-based policy.

In Chapter 4, we describe a delay management model that takes the limited capacity of the stations into account. Because the infrastructure at the railway stations is scarce, delays of connecting trains may propagate to other trains in the area. In order to take the effects of these secondary delays into account, we propose a delay management model that includes the stations' capacities. We model each station as a set of parallel tracks and apply headway constraints to make sure that two trains do not use the same platform track at the same time. Modeling the available infrastructure in this way allows us to reschedule the platform assignment, too. We perform computational tests to show that passenger delays can be reduced by allowing changes in the platform assignment. However, platform changes put pressure on the dispatching organization at the infrastructure manager and are annoying for the passengers. We therefore also introduce a bi-objective delay management model, which minimizes the passenger delay on the one hand, and the number of changes in the platform assignment on the other.

Integrating dispatching aspects into delay management models is the topic of Chapter 5. Microscopic train scheduling models capture all details of the railway infrastructure by modeling each junction and block signal individually. Doing so, these models are able to compute delay propagation with high accuracy. Because all details of the infrastructure are modeled explicitly, train scheduling models can only be applied to small parts of a

railway network. In contrast, delay management models should range over larger parts of the network, because passengers travel through the entire network. We propose to integrate train scheduling models into delay management, in order to combine the benefits of both. We developed a solution approach for delay management that iterates between a macroscopic delay management module and a microscopic train scheduling module. The delay management module makes the wait-depart decisions and determines a timetable, which is then validated locally at the stations by the train scheduling module. If the train scheduling module detects many secondary delays, these delays are fed back to the delay management module and the process is repeated. For large delays, the system behaves unstable, but for small source delays, this coupling is shown to be effective.
Summarizing, we develop delay management models that include passenger rerouting and incorporate the limited capacities of the stations. In a comparative study, we show that these models reduce passenger delays with respect to basic models and that the models outperform simple rules of thumb that are currently applied in practice. Finally, we conclude that by tuning the parameters of the basic models we find wait-depart decisions that are almost as good as those obtained by more complex models that incorporate passenger rerouting and the infrastructure capacity.

6.2 Recommendations

In this thesis, we have developed several extensions of the classical delay management models from Schöbel (2007) and Schachtebeck and Schöbel (2010). We have shown that the wait-depart decisions obtained with these extended models reduce the total delay for the passengers. A disadvantage of the extended models is their complexity: Solving the models to optimality takes too much time. We have therefore developed several heuristics, which apply solution methodologies for the classical models. In general, these heuristics perform very well. First, the quality of the solution is fairly good: The delay increase with respect to the optimal solutions is below 1%. Second, the computation times of the heuristics are much smaller than those of the exact methods. In practical applications, where decisions should be available within a short time, it is therefore better to apply the heuristics based on simpler models. The contribution of the exact methods in this setting is their ability to evaluate the performance of the heuristics: The solutions found with the heuristics can be compared to the optimal solutions. If the quality of the heuristics decreases, research should be devoted to developing better heuristics or faster optimal solution methods.
A first step towards the application of delay management models in practice would be

the validation of our results. Although numerous computational experiments support our findings, our models should be embedded in a simulation environment to decisively conclude that they can be applied to reduce passenger delays in reality. In such an environment, the wait-depart decisions can be re-optimized every time a new source delay appears. By precisely implementing the rules of thumb that are currently used, a fair comparison can be made between our model-based approach and the current practice.

If the validation proposed above supports our conclusion that the total delay for the passengers can be reduced by applying delay management models, we advise railway operators to experiment with these models in their dispatching centers. To this end, a conflict detection system should be installed that identifies connections that are not maintained in the current situation. Then, this system should be coupled to our delay management modules. If a connection conflict is detected, an optimization run is executed to determine whether connections should be maintained or not, given the real-time train delays. If it is decided that a train better waits for a delayed feeder train, a message is sent to the dispatcher with the number of involved passengers, the total delay reduction and the waiting time for the train. Based on this information, the dispatcher can then decide whether or not to implement the decision to maintain the connection.

6.3 Future Research

We see many possible directions for future research. First, our delay management model incorporating the station infrastructure views each station as a set of parallel tracks. However, large stations also contain huge interlocking areas with hundreds of switches. Our model neglects these areas, but in reality there may be many train conflicts that propagate delays to other trains. To model the propagation of delays due to the limited station capacity correctly, these interlocking areas could therefore be taken into account.

Second, the integration of train scheduling into delay management should be studied in more detail. Our iterative approach finds satisfactory solutions for small source delays, but is unable to solve instances with larger delays. By coupling the models more closely, for example by adapting the feedback loop from the train scheduling module to the delay management module, the iterative approach can be improved. Doing so, our approach can be adapted in such a way that instances with larger delays can be solved, too.

Finally, the online version of delay management should be studied. Our models assume that all source delays are known for a given period of time. Given these source delays, we decide which trains should wait for delayed feeder trains during the planning horizon. In real-world applications, the current delays of all trains are indeed known and some future delays may be anticipated. However, also new source delays will appear during the planning horizon under consideration. One way to deal with these source delays is to apply our model to find new wait-depart decisions every time a new source delay is identified. The performance of applying delay management models iteratively should be evaluated on real-world instances.

References

E. J. W. Abbink, B. W. V. van den Berg, L. G. Kroon, and M. Salomon. Allocation of Railway Rolling Stock for Passenger Trains. *Transportation Science*, 38:33–42, 2004.

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.

E. J. W. Abbink, J. van 't Wout, and D. Huisman. 07. Solving Large Scale Crew Scheduling Problems by using Iterative Partitioning. In C. Liebchen, R. K. Ahuja, and J. A. Mesa, editors, *7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'07)*, volume 7 of *OpenAccess Series in Informatics (OASIcs)*, pages 96–106. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2007.

E. J. W. Abbink, L. Albino, T. Dollevoet, D. Huisman, J. Roussado, and R. L. Saldanha. Solving large scale crew scheduling problems in practice. *Public Transport*, 3(2):149–164, 2011.

A. Berger, R. Hoffmann, U. Lorenz, and S. Stiller. Online railway delay management: Hardness, simulation and computation. *Simulation*, 87(7):616–629, 2011.

G. Caimi, F. Chudak, M. Fuchsberger, M. Laumanns, and R. Zenklusen. A New Resource-Constrained Multicommodity Flow Model for Conflict-Free Train Routing and Scheduling. *Transportation Science*, 45(2):212–227, 2011.

A. Caprara, M. Fischetti, and P. Toth. Modelling and Solving the Train Timetabling Problem. *Operations Research*, 50:851–861, 2002.

A. Caprara, L. Galli, and P. Toth. Solution of the train platforming problem. *Transportation Science*, 45(2):246–257, 2011.

C. Conte and A. Schöbel. Identifying dependencies among delays. In *proceedings of IAROR 2007*, 2007. ISBN 978-90-78271-02-4.

F. Corman. *Real-time Railway Traffic Management*. PhD thesis, Delft University of Technology, the Netherlands, 2010.

F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. A tabu search algorithm for rerouting trains during rail operations. *Transportation Research Part B: Methodological*, 44(1):175–192, 2010a.

F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Bi-objective conflict detection and resolution in railway traffic management. *Transportation Research Part C: Emerging Technologies*, 20(1):79–94, 2010b.

F. Corman, A. D'Ariano, M. Pranzo, and I. A. Hansen. Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area. *Transportation Planning and Technology*, 34(4):341–362, 2011.

F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Optimal inter-area coordination of train rescheduling decisions. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):71–88, 2012.

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

A. D'Ariano, D. Pacciarelli, and M. Pranzo. A Branch and Bound Algorithm for Scheduling Trains on Railway Networks. *European Journal of Operational Research*, 183(2): 643–657, 2007.

A. D'Ariano, F. Corman, D. Pacciarelli, and M. Pranzo. Reordering and local rerouting strategies to manage train traffic in real-time. *Transportation Science*, 42(2):405–419, 2008.

L. De Giovanni, G. Heilporn, and M. Labbé. Optimization models for the single delay management problem in public transportation. *European Journal of Operational Research*, 189(3):762–774, 2008.

R. de Vries, B. de Schutter, and B. de Moor. On max-algebraic models for transportation networks. In *Proceedings of the International Workshop on Discrete Event Systems*, pages 457–462, Cagliari, Italy, 1998.

T. Dollevoet and D. Huisman. Fast heuristics for delay management with passenger rerouting. Technical Report EI2011-35, Econometric Institute Report Series, 2011.

T. Dollevoet, M. Schmidt, and A. Schöbel. Delay management including capacities of stations. In A. Caprara and S. Kontogiannis, editors, *11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 20 of *OpenAccess Series in Informatics (OASIcs)*, pages 88–99. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2011.

T. Dollevoet, F. Corman, A. D'Ariano, and D. Huisman. An iterative optimization framework for delay management and train scheduling. Technical Report EI2012-10, Econometric Institute Report Series, 2012a.

T. Dollevoet, D. Huisman, M. Schmidt, and A. Schöbel. Delay management with rerouting of passengers. *Transportation Science*, 46(1):74–89, 2012b.

T. Dollevoet, D. Huisman, M. Schmidt, and A. Schöbel. Delay management including capacities of stations. Technical Report EI2012-22, Econometric Institute Report Series, 2012c.

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.

R. Freling, R. M. Lentink, L. G. Kroon, and D. Huisman. Shunting of Passenger Train Units in a Railway Station. *Transportation Science*, 39:261–272, 2005.

M. R. Garey and D. S. Johnson. *Computers and Intractability — A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, 1979.

M. Gatto. *On the Impact of Uncertainty on Some Optimization Problems: Combinatorial Aspects of Delay Management and Robust Online Scheduling*. PhD thesis, ETH Zürich, Switzerland, 2007.

M. Gatto, R. Jacob, L. Peeters, and A. Schöbel. The computational complexity of delay management. In *Graph-Theoretic Concepts in Computer Science: 31st International Workshop (WG 2005)*, volume 3787 of *Lecture Notes in Computer Science*, 2005.

M. Gatto, R. Jacob, L. Peeters, and P. Widmayer. On-line delay management on a single train line. In *Algorithmic Methods for Railway Optimization*, number 4359 in Lecture Notes in Computer Science, pages 306–320. Springer, 2007.

A. Ginkel and A. Schöbel. The Bicriteria Delay Management Problem in Public Transportation. *Tranportation Science*, 41(4):527–538, 2007.

R. M. P. Goverde. The max-plus algebra approach to railway timetable design. In *Computers in Railways VI: Proceedings of the 6th international conference on computer aided design, manufacture and operations in the railway and other advanced mass transit systems, Lisbon, 1998*, pages 339–350, 1998.

I. A. Hansen and J. Pachl, editors. *Railway Timetable and Traffic: Analysis, Modelling and Simulation*. Eurailpress, Hamburg, 2008.

J. S. Hooghiemstra, L. G. Kroon, M. A. Odijk, M. Salomon, and P. J. Zwaneveld. Decision support systems support the search for win-win solutions in railway network design. *Interfaces*, 29(2):15–32, 1999.

D. Huisman. A Column Generation Approach to Solve the Rail Crew Re-scheduling Problem. *European Journal of Operational Research*, 180:163–173, 2007.

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, 2005.

J. Jespersen-Groth, D. Potthoff, J. Clausen, D. Huisman, L. G. Kroon, G. Maróti, and M. Nielsen. Disruption Management in Passenger Railway Transportation. In R. Ahuja, R. Möhring, and C. Zaroliagis, editors, *Robust and Online Large-Scale Optimization*, volume 5868 of *Lecture Notes in Computer Science*, pages 399–421. Springer Berlin / Heidelberg, 2009.

N. Kliewer and L. Suhl. A note on the online nature of the railway delay management problem. *Networks*, 57(1):28–37, 2011.

L. G. Kroon. *Job scheduling and capacity planning in aircraft maintenance*. PhD thesis, Erasmus University Rotterdam, the Netherlands, 1990.

L. G. Kroon, H. E. Romeijn, and P. Zwaneveld. Routing trains through railway stations: complexity issues. *European Journal of Operational Research*, 98(3):485–498, 1997.

L. G. Kroon, D. Huisman, and G. Maróti. Optimisation Models for Railway Timetabling. In I. A. Hansen and J. Pachl, editors, *Railway Timetabling and Traffic*, pages 135–154. Eurailpress, 2008a.

L. G. Kroon, G. Maróti, M. Retel Helmrich, M. Vromans, and R. Dekker. Stochastic improvement of cyclic railway timetables. *Transportation Research Part B: Methodological*, 42:553–570, 2008b.

L. G. Kroon, D. Huisman, E. J. W. 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. G. Kroon, G. Maróti, and L. Nielsen. Rescheduling of railway rolling stock with dynamic passenger flows. Technical report, ERIM Report Series Research in Management, Erasmus University Rotterdam, 2010.

S. Krumke, C. Thielen, and C. Zeck. Extensions to online delay management on a single train line: new bounds for delay minimization and profit maximization. *Mathematical Methods of Operations Research*, 74(1):53–75, 2011.

R. M. Lentink. *Algorithmic Decision Support for Shunt Planning.* PhD thesis, Erasmus University Rotterdam, the Netherlands, 2006.

C. Liebchen. The first optimized railway timetable in practice. *Transportation Science*, 42:420–435, 2008.

R. Lusby, J. Larsen, M. Ehrgott, and D. Ryan. Railway track allocation: models and methods. *OR Spectrum*, 33(4):843–883, 2011.

G. Maróti. *Operations research models for railway rolling stock planning.* PhD thesis, Eindhoven University of Technology, the Netherlands, 2006.

A. Mascis and D. Pacciarelli. Job shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, 143(3):498–517, 2002.

D. Middelkoop and M. Bouwman. SIMONE: Large scale train network simulations. In B. Peters and J. Smith, editors, *Proceedings 2001 Winter Simulation Conference*, pages 1042–1047. Institute for Electrical and Electronic Engineers Press, Piscataway, NJ, 2001.

K. Nachtigall. *Periodic Network Optimization and Fixed Interval Timetables.* Deutsches Zentrum für Luft– und Raumfahrt, Institut für Flugführung, Braunschweig, 1998. Habilitationsschrift.

L. K. Nielsen. *Rolling stock rescheduling in passenger railways: Applications in short term planning and in disruption management.* PhD thesis, Erasmus University Rotterdam, the Netherlands, 2011.

L. K. Nielsen, L. G. Kroon, and G. Maróti. A rollign horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, 220: 469–509, 2012.

L. W. P. Peeters. *Cyclic Railway Timetable Optimization*. PhD thesis, Erasmus University Rotterdam, the Netherlands, 2003.

D. Potthoff. *Railway crew rescheduling: Novel approaches and extensions*. PhD thesis, Erasmus University Rotterdam, the Netherlands, 2010.

D. Potthoff, D. Huisman, and G. Desaulniers. Column generation with dynamic duty selection for railway crew rescheduling. *Transportation Science*, 44(4):493–505, 2010.

M. Schachtebeck. *Delay Management in Public Transportation: Capacities, Robustness, and Integration*. PhD thesis, Universität Göttingen, Germany, 2010.

M. Schachtebeck and A. Schöbel. To wait or not to wait and who goes first? Delay management with priority decisions. *Transportation Science*, 44(3):307–321, 2010.

M. Schmidt. *Integrating Routing Decisions in Network Problems*. PhD thesis, Universität Göttingen, Germany, 2012.

A. Schöbel. A model for the delay management problem based on mixed-integer programming. *Electronic Notes in Theoretical Computer Science*, 50(1):1–10, 2001.

A. Schöbel. *Optimization in public transportation. Stop location, delay management and tariff planning from a customer-oriented point of view*. Optimization and Its Applications. Springer, New York, 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*, number 4359 in Lecture Notes in Computer Science, pages 145–170. Springer, 2007.

A. Schöbel. Capacity constraints in delay management. *Public Transport*, 1(2):135–154, 2009.

A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin Heidelberg New York, 2003.

A. Schrijver and A. Steenbeek. Spoorweg Dienstregelontwikkeling (Railway Timetable Development). Technical report, Center for Mathematics and Computer Science, Amsterdam, 1993.

P. Serafini and W. Ukovich. A Mathematical Model for Periodic Event Scheduling Problems. *SIAM journal of Discrete Mathematics*, 2:550–581, 1989.

L. Suhl and T. Mellouli. Requirements for, and design of, an operations control system for railways. In *Computer-Aided Transit Scheduling*. Springer, 1999.

L. Suhl, T. Mellouli, C. Biederbick, and J. Goecke. Managing and preventing delays in railway traffic by simulation and optimization. In *Mathematical methods on Optimization in Transportation Systems*, pages 3–16. Kluwer, 2001.

J. Törnquist. Design of an effective algorithm for fast response to the re-scheduling of railway traffic during disturbances. *Transportation Research Part C: Emerging Technologies*, 20(1):62–78, 2012.

L. P. Veelenturf, D. Potthoff, D. Huisman, and L. G. Kroon. Column generation with dynamic duty selection for railway crew rescheduling. *Transportation Research Part C: Emerging Technologies*, 20:95–110, 2011.

L. P. Veelenturf, D. Potthoff, D. Huisman, L. G. Kroon, G. Maróti, and A. P. M. Wagelmans. A recoverable robust solution approach for real-time railway crew rescheduling. Technical report, Erasmus University Rotterdam, 2012.

M. J. C. M. Vromans. *Reliability of Railway Systems*. PhD thesis, Erasmus University Rotterdam, the Netherlands, 2005.

P. J. Zwaneveld, L. G. Kroon, and C. P. M. van Hoesel. Routing Trains through Railway Stations Based on a Node Packing Model. *European Journal of Operations Research*, 128:14–33, 2001.

Nederlandse Samenvatting (Summary in Dutch)

Tijdens mijn studietijd heb ik, samen met vele andere studenten en forenzen, erg veel met de trein gereisd. Een voor mij populaire treinreis begon in Rosmalen, waar ik in een stoptrein richting 's-Hertogenbosch stapte. In 's-Hertogenbosch stapte ik vervolgens over op een Intercity naar Utrecht en vanaf daar ging ik verder naar de Utrechtse Uithof of in de richting van Rotterdam. (Zie ook figuur 1.1 op bladzijde 2; de stoptrein via Rosmalen vertrekt vanuit Nijmegen.) Voor de overstap in 's-Hertogenbosch had ik 5 minuten en de trein naar Utrecht vertrok vanaf de overzijde van het perron waar de trein uit Rosmalen aankwam. Als de trein uit Rosmalen meer dan vijf minuten vertraging had vond ik dat niet zo erg, want een kwartier na mijn geplande trein naar Utrecht ging er alweer een volgende.
Op de terugreis herhaalden deze stappen zich in omgekeerde volgorde. Vanaf Utrecht nam ik een sneltrein naar 's-Hertogenbosch en daar stapte ik over op een stoptrein naar Rosmalen. Ook in deze richting was de geplande overstaptijd 5 minuten, maar in dit geval vertrok de stoptrein niet vanaf hetzelfde perron. Het kwam daardoor regelmatig voor dat ik de stoptrein net miste als de Intercity vertraging had. Als het spits was stormde ik dan in een enorme groep mensen richting de stoptrein en zag daar dat de deuren juist werden gesloten op het moment dat wij de roltrap afrenden. Dit was extra vervelend omdat de stoptrein maar twee keer per uur rijdt en ik dus een half uur moest wachten. Vaak vroeg ik me dan gefrustreerd af: "Waarom kan die stoptrein niet één minuut wachten op al deze reizigers?"

Bovenstaande vraag is de centrale vraag in dit proefschrift: Als een trein met vertraging aankomt op een station waar veel reizigers willen overstappen, is het dan beter om de volgende trein op tijd te laten vertrekken of kan beter gewacht worden op de vertraagde reizigers. Een wachtende trein zal met vertraging vertrekken en daarmee enerzijds de treinpunctualiteit verlagen. Anderzijds zal dit een positief effect hebben op de *reizigers-*

punctualiteit. De reizigerspunctualiteit meet het percentage reizigers dat met een vertraging van ten hoogste vijf minuten op de eindbestemming aankomt.
Het onderzoeksgebied dat bestudeert of treinen op overstappende passagiers zouden moeten wachten is een onderdeel van Mathematische Besliskunde en heet in het Engels “Delay Management” (door mij vertaald als “het besturen van aansluitingen” in de titel van dit proefschrift). De doelstelling is hierbij het minimaliseren van de totale vertraging van reizigers. In het voorbeeld hierboven worden al enige belangrijke aspecten van dit vraagstuk benoemd: (1) het aantal reizigers dat wil overstappen, (2) de tijd die reizigers moeten wachten op de volgende trein en (3) de vertrekvertraging die de wachtende trein oploopt. Twee andere aspecten die voor reizigers minder zichtbaar zijn, zijn (4) de aankomstvertraging van de wachtende trein op het volgende station en (5) de invloed die de vertraagde trein heeft op andere treinen die gebruik maken van hetzelfde spoornetwerk. Het kan bijvoorbeeld gebeuren dat een Intercity als gevolg van het wachten kort ná een stoptrein vertrekt in plaats van ervoor. Hierdoor zal de Intercity nog meer vertraging oplopen. Dit laatste is overigens de reden dat de stoptrein in ’s-Hertogenbosch uit het voorbeeld niet wacht: alle treinen die vanuit ’s-Hertogenbosch in noordelijke richting vertrekken gaan over hetzelfde spoor op de Diezebrug. Als de stoptrein zou wachten, zouden de Intercity’s naar Utrecht en Zwolle daardoor ook vertraging oplopen.
Er is veel onderzoek gedaan naar Delay Management in de spoorsector. In de literatuur vinden we voornamelijk modellen voor Delay Management die uitgaan van een *cyclische* dienstregeling. We noemen een dienstregeling cyclisch als deze zichzelf na een bepaalde tijd herhaalt. Als de herhalingstijd één uur bedraagt, zoals in Nederland, ziet de dienstregeling tussen 8 en 9 uur er hetzelfde uit als de dienstregeling tussen 9 en 10 uur. De modellen uit de literatuur nemen nu aan dat reizigers een vertraging oplopen van precies één uur wanneer ze een aansluiting missen. We noemen het model dat uitgaat van deze aanname het klassieke model. We zagen in het voorbeeld al dat deze aanname niet altijd geldig is. Reizigers die bijvoorbeeld de overstap op de Intercity naar Utrecht missen hoeven niet een uur te wachten, maar slechts 15 minuten. In hoofdstuk 2 van dit proefschrift wordt daarom een model voorgesteld dat de vertraging voor reizigers die een overstap missen niet benadert, maar exact uitrekent. Om dit te doen wordt in het model voor alle reizigers die een overstap missen een alternatieve route bepaald. Hierdoor kunnen de nadelige effecten van het missen van een overstap exact worden berekend, en kan precies worden bepaald welke treinen moeten wachten en welke treinen beter op tijd kunnen vertrekken. We laten in een simulatie zien dat we hiermee de totale vertraging voor reizigers met 8% kunnen verminderen. Voor deze simulatie, en alle andere die volgen, hebben we gebruik gemaakt van data die we van NS hebben gekregen.

Het grootste nadeel van ons model uit hoofdstuk 2 is de verhoogde complexiteit. Omdat we bij deze aanpak voor veel reizigers een alternatieve route moeten bepalen, duurt het veel langer om te berekenen welke aansluitingen moeten worden gehandhaafd. Als we ons model in de praktijk willen toepassen, kunnen we ons deze tijd niet veroorloven: als een trein binnen enkele minuten zal vertrekken kunnen de bijstuurders van een spoorwegvervoerder niet eerst een kwartier wachten tot er is uitgerekend of de trein zal wachten of niet. Daarom stellen we in hoofdstuk 3 een aantal heuristieken voor, die niet de absoluut minimale vertraging opleveren, maar wel proberen daarbij zeer dicht in de buurt te komen. Een voorbeeld is een iteratieve heuristiek die het originele model herhaaldelijk toepast met veranderende parameters. Door de parameters precies de goede waarden te geven, vinden we een oplossing die hoogstens 1% slechter is dan de optimale oplossing. De rekentijd die de heuristiek nodig heeft is in dit geval slechts enkele seconden. De heuristiek is dus in de praktijk wel bruikbaar en in onze experimenten hebben we laten zien dat de kwaliteit van de oplossing slechts marginaal wordt verlaagd.

Alle modellen die tot nu toe zijn beschreven negeren de effecten die vertraagde treinen hebben op de andere treinen op hetzelfde spoornetwerk. De modellen gaan er dus bijvoorbeeld van uit dat een Intercity die achter een stoptrein komt daardoor geen extra vertraging oploopt. In de praktijk zal dat natuurlijk wel gebeuren. Tegelijkertijd met het onderzoek dat in dit proefschrift wordt gepresenteerd, zijn er modellen ontwikkeld die de onderlinge invloed van treinen op de vrije baan meewegen. Met de vrije baan bedoelen we hier de stukken spoor tussen twee stations. In deze modellen wordt ervoor gezorgd dat een trein die eerder vertrekt van een bepaald station, ook eerder op het volgende station zal aankomen. Als een Intercity achter een stoptrein komt, zal deze dus tot het volgende station achter de stoptrein blijven, en hierdoor extra vertraging oplopen. In hoofdstuk 4 breiden we dit model uit door ook de beperkte capaciteit binnen de stations te modelleren. We vatten een station hierbij op als een aantal parallelle sporen. Treinen die gebruik maken van hetzelfde spoor in het station, kunnen elkaar niet inhalen. Als aan twee treinen verschillende sporen worden toegewezen, is het mogelijk dat de ene trein de andere in het station inhaalt. Een bijkomend voordeel van dit model is dat kan worden bepaald hoeveel treinen vanaf een ander perron vertrekken. Als een trein niet vanaf het geplande perron vertrekt, is dit zowel voor de bijstuurders als voor de reizigers vervelend. Met ons nieuwe model kan een afweging worden gemaakt tussen de totale vertraging voor reizigers enerzijds en het aantal wisselingen in de perrontoewijzing anderzijds.

Het model uit hoofdstuk 4 neemt wel het beperkte aantal sporen in een station mee, maar niet hoe de infrastructuur er rond het station precies uitziet. Net buiten de grote stations vinden we gebieden met fly-overs en tientallen wissels. Ook in deze gebieden zullen vertraagde treinen rode seinen veroorzaken voor andere treinen. Het is vooralsnog niet mogelijk om deze gebieden op een exacte manier in ons model mee te nemen. Daarom ontwikkelen we in hoofdstuk 5 een oplosmethode die itereert tussen een globale aanpak voor het delay management probleem en een lokale aanpak voor de precieze routering van treinen door de stations. Op deze manier kunnen we de oplossingen van ons delay management model evalueren door de precieze vertraging voor treinen en passagiers met een microscopisch model te bepalen. We tonen hierbij aan dat het globale delay management model de vertraging voor reizigers goed benadert. Daarnaast kunnen we met de iteratieve aanpak de globale oplossing uit de eerste iteratie verbeteren. Voor kleine vertragingen werkt deze aanpak goed en kunnen we de vertraging inderdaad verminderen. Als we de aanpak toepassen met grotere vertragingen werkt deze methodiek echter niet. Om ook met grotere vertragingen om te kunnen gaan, zullen we de iteratieve aanpak dus nog nader moeten onderzoeken.

Samenvattend hebben we in dit proefschrift de modellen voor Delay Management op een aantal manieren uitgebreid. We hebben algoritmen ontwikkeld om deze modellen op te lossen, waarmee de totale vertraging voor reizigers kan worden verminderd. Onze aanbeveling is nu om deze methoden in een simulatiemodel te testen en te vergelijken met de huidige aanpak van de vervoerders. Als uit de simulatie dezelfde conclusies volgen als uit onze testen, namelijk dat toepassing van de modellen leidt tot minder reizigersvertraging, kunnen de modellen worden ingebouwd in de bijsturingssystemen.

Curriculum Vitae

Twan Dollevoet (1983) holds master's degrees in Mathematics and in Physics from Utrecht University and in Econometrics and Management Science from Erasmus University Rotterdam. In 2008, he started his PhD research at the Erasmus School of Economics. His main research interest is in Operations Research, and in particular in applications of combinatorial optimization, such as railway scheduling, inventory control, and route planning for unmanned aerial vehicles. Also since 2008, Twan works as a researcher at the department Process quality & Innovation of Netherlands Railways. At Netherlands Railways, he mainly works on crew scheduling.

The research described in this thesis has been presented at various conferences, such as TRISTAN, EURO, the INFORMS Annual Meeting, IFORS, and CASPT. The second chapter of this thesis has been published in *Transportation Science*. For this chapter, Twan and Marie Schmidt were awarded the first prize in the 2010 Student Research Paper Contest in the Railroad Application Section of INFORMS. The prize was presented at the 2010 Annual Meeting in Austin, TX. Papers based on the other chapters have been submitted to *Transportation Science*, *Public Transport*, and *Flexible Services and Manufacturing*. Furthermore, a paper on crew scheduling has been published in *Public Transport*, and papers on mission planning and inventory control have been submitted to *Annals of Operations Research* and *IIE Transactions*, respectively.

From January 2013, Twan will work as an assistant professor at the Econometric Institute of Erasmus University Rotterdam. He will continue to work on real-time railway scheduling within the European project ON-TIME.

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT (ERIM)

ERIM PH.D. SERIES RESEARCH IN MANAGEMENT

The ERIM PhD Series contains PhD dissertations in the field of Research in Management defended at Erasmus University Rotterdam and supervised by senior researchers affiliated to the Erasmus Research Institute of Management (ERIM). All dissertations in the ERIM PhD Series are available in full text through the ERIM Electronic Series Portal: http://hdl.handle.net/1765/1
ERIM is the joint research institute of the Rotterdam School of Management (RSM) and the Erasmus School of Economics at the Erasmus University Rotterdam (EUR).

DISSERTATIONS LATEST FIVE YEARS

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

Agatz, N.A.H., *Demand Management in E-Fulfillment*. Promoter(s): Prof.dr.ir. J.A.E.E. van Nunen, EPS-2009-163-LIS, http://hdl.handle.net/1765/15425

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

Asperen, E. van, *Essays on Port, Container, and Bulk Chemical Logistics Optimization*. Promoter(s): Prof.dr.ir. R. Dekker, EPS-2009-181-LIS, http://hdl.handle.net/1765/17626

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

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

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

Bezemer, P.J., *Diffusion of Corporate Governance Beliefs: Board Independence and the Emergence of a Shareholder Value Orientation in the Netherlands.* Promoter(s): Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2009-192-STR, http://hdl.handle.net/1765/18458

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

Blitz, D.C., *Benchmarking Benchmarks,* Promoter(s): Prof.dr. A.G.Z. Kemna & Prof.dr. W.F.C. Verschoor, EPS-2011-225-F&A, http://hdl.handle.net/1765/22624

Boer-Sorbán, K., *Agent-Based Simulation of Financial Markets: A modular, Continuous-Time Approach,* Promoter(s): Prof.dr. A. de Bruin, EPS-2008-119-LIS, http://hdl.handle.net/1765/10870

Boon, C.T., *HRM and Fit: Survival of the Fittest!?,* Promoter(s): Prof.dr. J. Paauwe & Prof.dr. D.N. den Hartog, EPS-2008-129-ORG, http://hdl.handle.net/1765/12606

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

Braun, E., *City Marketing: Towards an Integrated Approach,* Promoter(s): Prof.dr. L. van den Berg, EPS-2008-142-MKT, http://hdl.handle.net/1765/13694

Brumme, W.-H., *Manufacturing Capability Switching in the High-Tech Electronics Technology Life Cycle,* Promoter(s): Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. L.N. Van Wassenhove, EPS-2008-126-LIS, http://hdl.handle.net/1765/12103

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

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

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

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

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

Chen, C.-M., *Evaluation and Design of Supply Chain Operations Using DEA*, Promoter(s): Prof.dr. J.A.E.E. van Nunen, EPS-2009-172-LIS, http://hdl.handle.net/1765/16181

Chen, H., *Individual Mobile Communication Services and Tariffs*, Promoter(s): Prof.dr. L.F.J.M. Pau, EPS-2008-123-LIS, http://hdl.handle.net/1765/11141

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

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

Derwall, J.M.M., *The Economic Virtues of SRI and CSR*, Promoter(s): Prof.dr. C.G. Koedijk, EPS-2007-101-F&A, http://hdl.handle.net/1765/8986

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

Diepen, M. van, *Dynamics and Competition in Charitable Giving*, Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2009-159-MKT, http://hdl.handle.net/1765/14526

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

Dietz, H.M.S., *Managing (Sales)People towards Performance: HR Strategy, Leadership & Teamwork*, Promoter(s): Prof.dr. G.W.J. Hendrikse, EPS-2009-168-ORG, http://hdl.handle.net/1765/16081

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

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

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

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

Eijk, A.R. van der, *Behind Networks: Knowledge Transfer, Favor Exchange and Performance*, Promoter(s): Prof.dr. S.L. van de Velde & Prof.dr.drs. W.A. Dolfsma, EPS-2009-161-LIS, 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*, Promoter(s): Prof.dr. C.B.M. van Riel, EPS-2008-117-ORG, http://hdl.handle.net/1765/10723

Erken, H.P.G., *Productivity, R&D and Entrepreneurship*, Promoter(s): Prof.dr. A.R. Thurik, EPS-2008-147-ORG, http://hdl.handle.net/1765/14004

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

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

Gertsen, H.F.M., *Riding a Tiger without Being Eaten: How Companies and Analysts Tame Financial Restatements and Influence Corporate Reputation*, Promoter(s): Prof.dr. C.B.M. van Riel, EPS-2009-171-ORG, http://hdl.handle.net/1765/16098

Gijsbers, G.W., *Agricultural Innovation in Asia: Drivers, Paradigms and Performance,* Promoter(s): Prof.dr. R.J.M. van Tulder, EPS-2009-156-ORG, http://hdl.handle.net/1765/14524

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

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

Gong, Y., *Stochastic Modelling and Analysis of Warehouse Operations*, Promoter(s): Prof.dr. M.B.M. de Koster & Prof.dr. S.L. van de Velde, EPS-2009-180-LIS, http://hdl.handle.net/1765/16724

Greeven, M.J., *Innovation in an Uncertain Institutional Environment: Private Software Entrepreneurs in Hangzhou, China*, Promoter(s): Prof.dr. B. Krug, EPS-2009-164-ORG, http://hdl.handle.net/1765/15426

Guenster, N.K., *Investment Strategies Based on Social Responsibility and Bubbles*, Promoter(s): Prof.dr. C.G. Koedijk, EPS-2008-175-F&A, http://hdl.handle.net/1765/16209

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

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*, Promoter(s): Prof.dr. S.B.M. van Riel, EPS-2008-122-ORG, http://hdl.handle.net/1765/10872

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

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

Hessels, S.J.A., *International Entrepreneurship: Value Creation Across National Borders*, Promoter(s): Prof.dr. A.R. Thurik, EPS-2008-144-ORG, http://hdl.handle.net/1765/13942

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

Hoedemaekers, C.M.W., *Performance, Pinned down: A Lacanian Analysis of Subjectivity at Work*, Promoter(s): Prof.dr. S. Magala & Prof.dr. D.H. den Hartog, EPS-2008-121-ORG, http://hdl.handle.net/1765/10871

Hoever, I.J., *Diversity and Creativity: In Search of Synergy*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2012-267-ORG, http://hdl.handle.net/1765/1

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

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

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

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

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

Jaspers, F.P.H., *Organizing Systemic Innovation*, Promoter(s): Prof.dr.ir. J.C.M. van den Ende, EPS-2009-160-ORG, http://hdl.handle.net/1765/14974

Jennen, M.G.J., *Empirical Essays on Office Market Dynamics*, Promoter(s): Prof.dr. C.G. Koedijk & Prof.dr. D. Brounen, EPS-2008-140-F&A, http://hdl.handle.net/1765/13692

Jiang, T., *Capital Structure Determinants and Governance Structure Variety in Franchising*, Promoter(s): Prof.dr. G. Hendrikse & Prof.dr. A. de Jong, EPS-2009-158-F&A, http://hdl.handle.net/1765/14975

Jiao, T., *Essays in Financial Accounting*, Promoter(s): Prof.dr. G.M.H. Mertens, EPS-2009-176-F&A, http://hdl.handle.net/1765/16097

Kaa, G. van, *Standard Battles for Complex Systems: Empirical Research on the Home Network,* Promoter(s): Prof.dr.ir. J. van den Ende & Prof.dr.ir. H.W.G.M. van Heck, EPS-2009-166-ORG, http://hdl.handle.net/1765/16011

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

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

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

Klein, M.H., *Poverty Alleviation through Sustainable Strategic Business Models: Essays on Poverty Alleviation as a Business Strategy,* Promoter(s): Prof.dr. H.R. Commandeur, EPS-2008-135-STR, http://hdl.handle.net/1765/13482

Krauth, E.I., *Real-Time Planning Support: A Task-Technology Fit Perspective,* Promoter(s): Prof.dr. S.L. van de Velde & Prof.dr. J. van Hillegersberg, EPS-2008-155-LIS, http://hdl.handle.net/1765/14521

Kwee, Z., *Investigating Three Key Principles of Sustained Strategic Renewal: A Longitudinal Study of Long-Lived Firms,* Promoter(s): Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2009-174-STR, http://hdl.handle.net/1765/16207

Lam, K.Y., *Reliability and Rankings*, Promoter(s): Prof.dr. P.H.B.F. Franses, EPS-2011-230-MKT, http://hdl.handle.net/1765/22977

Lander, M.W., *Profits or Professionalism? On Designing Professional Service Firms*, Promoter(s): Prof.dr. J. van Oosterhout & Prof.dr. P.P.M.A.R. Heugens, EPS-2012-253-ORG, http://hdl.handle.net/1765/30682

Langhe, B. de, *Contingencies: Learning Numerical and Emotional Associations in an Uncertain World*, Promoter(s): Prof.dr.ir. B. Wierenga & Prof.dr. S.M.J. van Osselaer, EPS-2011-236-MKT, http://hdl.handle.net/1765/23504

Larco Martinelli, J.A., *Incorporating Worker-Specific Factors in Operations Management Models,* Promoter(s): Prof.dr.ir. J. Dul & Prof.dr. M.B.M. de Koster, EPS-2010-217-LIS, http://hdl.handle.net/1765/21527

Li, T., *Informedness and Customer-Centric Revenue Management,* Promoter(s): Prof.dr. P.H.M. Vervest & Prof.dr.ir. H.W.G.M. van Heck, EPS-2009-146-LIS, http://hdl.handle.net/1765/14525

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

Lovric, M., *Behavioral Finance and Agent-Based Artificial Markets*, Promoter(s): Prof.dr. J. Spronk & Prof.dr.ir. U. Kaymak, EPS-2011-229-F&A, http://hdl.handle.net/1765/ 22814

Maas, A.A., van der, *Strategy Implementation in a Small Island Context: An Integrative Framework,* Promoter(s): Prof.dr. H.G. van Dissel, EPS-2008-127-LIS, http://hdl.handle.net/1765/12278

Maas, K.E.G., *Corporate Social Performance: From Output Measurement to Impact Measurement,* Promoter(s): Prof.dr. H.R. Commandeur, EPS-2009-182-STR, http://hdl.handle.net/1765/17627

Markwat, T.D., *Extreme Dependence in Asset Markets Around the Globe*, Promoter(s): Prof.dr. D.J.C. van Dijk, EPS-2011-227-F&A, http://hdl.handle.net/1765/22744

Mees, H., *Changing Fortunes: How China's Boom Caused the Financial Crisis*, Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2012-266-MKT, http://hdl.handle.net/1765/34930

Meuer, J., *Configurations of Inter-Firm Relations in Management Innovation: A Study in China's Biopharmaceutical Industry*, Promoter(s): Prof.dr. B. Krug, EPS-2011-228-ORG, http://hdl.handle.net/1765/22745

Mihalache, O.R., *Stimulating Firm Innovativeness: Probing the Interrelations between Managerial and Organizational Determinants*, Promoter(s): Prof.dr. J.J.P. Jansen, Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2012-260-S&E, http://hdl.handle.net/1765/32343

Moitra, D., *Globalization of R&D: Leveraging Offshoring for Innovative Capability and Organizational Flexibility,* Promoter(s): Prof.dr. K. Kumar, EPS-2008-150-LIS, http://hdl.handle.net/1765/14081

Moonen, J.M., *Multi-Agent Systems for Transportation Planning and Coordination*, Promoter(s): Prof.dr. J. van Hillegersberg & Prof.dr. S.L. van de Velde, EPS-2009-177-LIS, http://hdl.handle.net/1765/16208

Nalbantov G.I., *Essays on Some Recent Penalization Methods with Applications in Finance and Marketing*, Promoter(s): Prof. dr P.J.F. Groenen, EPS-2008-132-F&A, http://hdl.handle.net/1765/13319

Nederveen Pieterse, A., *Goal Orientation in Teams: The Role of Diversity*, Promoter(s): 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*, Promoter(s): Prof.dr. A. de Jong, EPS-2008-148-F&A, http://hdl.handle.net/1765/14005

Nielsen, L.K., *Rolling Stock Rescheduling in Passenger Railways: Applications in Short-term Planning and in Disruption Management*, Promoter(s): Prof.dr. L.G. Kroon, EPS-2011-224-LIS, http://hdl.handle.net/1765/22444

Niesten, E.M.M.I., *Regulation, Governance and Adaptation: Governance Transformations in the Dutch and French Liberalizing Electricity Industries*, Promoter(s): Prof.dr. A. Jolink & Prof.dr. J.P.M. Groenewegen, EPS-2009-170-ORG, http://hdl.handle.net/1765/16096

Nieuwenboer, N.A. den, *Seeing the Shadow of the Self*, Promoter(s): Prof.dr. S.P. Kaptein, EPS-2008-151-ORG, http://hdl.handle.net/1765/14223

Nijdam, M.H., *Leader Firms: The Value of Companies for the Competitiveness of the Rotterdam Seaport Cluster,* Promoter(s): Prof.dr. R.J.M. van Tulder, EPS-2010-216-ORG, http://hdl.handle.net/1765/21405

Noordegraaf-Eelens, L.H.J., *Contested Communication: A Critical Analysis of Central Bank Speech,* Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2010-209-MKT, http://hdl.handle.net/1765/21061

Nuijten, A.L.P., *Deaf Effect for Risk Warnings: A Causal Examination applied to Information Systems Projects*, Promoter(s): Prof.dr. G. van der Pijl & Prof.dr. H. Commandeur & Prof.dr. M. Keil, EPS-2012-263-S&E, http://hdl.handle.net/1765/34928

Nuijten, I., *Servant Leadership: Paradox or Diamond in the Rough? A Multidimensional Measure and Empirical Evidence*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2009-183-ORG, http://hdl.handle.net/1765/21405

Oosterhout, M., van, *Business Agility and Information Technology in Service Organizations,* Promoter(s): Prof,dr.ir. H.W.G.M. van Heck, EPS-2010-198-LIS, http://hdl.handle.net/1765/19805

Oostrum, J.M., van, *Applying Mathematical Models to Surgical Patient Planning*, Promoter(s): Prof.dr. A.P.M. Wagelmans, EPS-2009-179-LIS, http://hdl.handle.net/1765/16728

Osadchiy, S.E., *The Dynamics of Formal Organization: Essays on Bureaucracy and Formal Rules*, Promoter(s): Prof.dr. P.P.M.A.R. Heugens, EPS-2011-231-ORG, http://hdl.handle.net/1765/23250

Otgaar, A.H.J., *Industrial Tourism: Where the Public Meets the Private,* Promoter(s): Prof.dr. L. van den Berg, EPS-2010-219-ORG, http://hdl.handle.net/1765/21585

Ozdemir, M.N., *Project-level Governance, Monetary Incentives and Performance in Strategic R&D Alliances*, Promoter(s): Prof.dr.ir. J.C.M. van den Ende, EPS-2011-235-LIS, http://hdl.handle.net/1765/23550

Peers, Y., *Econometric Advances in Diffusion Models*, Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2011-251-MKT, http://hdl.handle.net/1765/ 30586

Pince, C., *Advances in Inventory Management: Dynamic Models,* Promoter(s): Prof.dr.ir. R. Dekker, EPS-2010-199-LIS, http://hdl.handle.net/1765/19867

Porras Prado, M., *The Long and Short Side of Real Estate, Real Estate Stocks, and Equity*, Promoter(s): Prof.dr. M.J.C.M. Verbeek, EPS-2012-254-F&A, http://hdl.handle.net/1765/30848

Potthoff, D., *Railway Crew Rescheduling: Novel Approaches and Extensions,* Promoter(s): Prof.dr. A.P.M. Wagelmans & Prof.dr. L.G. Kroon, EPS-2010-210-LIS, http://hdl.handle.net/1765/21084

Poruthiyil, P.V., *Steering Through: How Organizations Negotiate Permanent Uncertainty and Unresolvable Choices*, Promoter(s): Prof.dr. P.P.M.A.R. Heugens & Prof.dr. S. Magala, EPS-2011-245-ORG, http://hdl.handle.net/1765/26392

Pourakbar, M. *End-of-Life Inventory Decisions of Service Parts*, Promoter(s): Prof.dr.ir. R. Dekker, EPS-2011-249-LIS, http://hdl.handle.net/1765/30584

Prins, R., *Modeling Consumer Adoption and Usage of Value-Added Mobile Services*, Promoter(s): Prof.dr. Ph.H.B.F. Franses & Prof.dr. P.C. Verhoef, EPS-2008-128-MKT, http://hdl.handle.net/1765/12461

Quak, H.J., *Sustainability of Urban Freight Transport: Retail Distribution and Local Regulation in Cities*, Promoter(s): Prof.dr. M.B.M. de Koster, EPS-2008-124-LIS, http://hdl.handle.net/1765/11990

Quariguasi Frota Neto, J., *Eco-efficient Supply Chains for Electrical and Electronic Products*, Promoter(s): Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. H.W.G.M. van Heck, EPS-2008-152-LIS, http://hdl.handle.net/1765/14785

Radkevitch, U.L, *Online Reverse Auction for Procurement of Services*, Promoter(s): Prof.dr.ir. H.W.G.M. van Heck, EPS-2008-137-LIS, http://hdl.handle.net/1765/13497

Rijsenbilt, J.A., *CEO Narcissism; Measurement and Impact*, Promoter(s): Prof.dr. A.G.Z. Kemna & Prof.dr. H.R. Commandeur, EPS-2011-238-STR, http://hdl.handle.net/1765/ 23554

Roelofsen, E.M., *The Role of Analyst Conference Calls in Capital Markets*, Promoter(s): Prof.dr. G.M.H. Mertens & Prof.dr. L.G. van der Tas RA, EPS-2010-190-F&A, http://hdl.handle.net/1765/18013

Rook, L., *Imitation in Creative Task Performance*, Promoter(s): 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*, Promoter(s): Prof.dr. P.J.F. Groenen, EPS-2009-165-MKT, http://hdl.handle.net/1765/15536

Roza, M.W., *The Relationship between Offshoring Strategies and Firm Performance: Impact of Innovation, Absorptive Capacity and Firm Size,* Promoter(s): Prof.dr. H.W. Volberda & Prof.dr.ing. F.A.J. van den Bosch, EPS-2011-214-STR, http://hdl.handle.net/1765/22155

Rus, D., *The Dark Side of Leadership: Exploring the Psychology of Leader Self-serving Behavior*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2009-178-ORG, http://hdl.handle.net/1765/16726

Samii, R., *Leveraging Logistics Partnerships: Lessons from Humanitarian Organizations,* Promoter(s): Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. L.N. Van Wassenhove, EPS-2008-153-LIS, http://hdl.handle.net/1765/14519

Schaik, D. van, *M&A in Japan: An Analysis of Merger Waves and Hostile Takeovers,* Promoter(s): Prof.dr. J. Spronk & Prof.dr. J.P.M. Groenewegen, EPS-2008-141-F&A, http://hdl.handle.net/1765/13693

Schauten, M.B.J., *Valuation, Capital Structure Decisions and the Cost of Capital,* Promoter(s): Prof.dr. J. Spronk & Prof.dr. D. van Dijk, EPS-2008-134-F&A, http://hdl.handle.net/1765/13480

Schellekens, G.A.C., *Language Abstraction in Word of Mouth,* Promoter(s): Prof.dr.ir. A. Smidts, EPS-2010-218-MKT, ISBN: 978-90-5892-252-6, http://hdl.handle.net/1765/21580

Sotgiu, F., *Not All Promotions are Made Equal: From the Effects of a Price War to Cross-chain Cannibalization,* Promoter(s): Prof.dr. M.G. Dekimpe & Prof.dr.ir. B. Wierenga, EPS-2010-203-MKT, http://hdl.handle.net/1765/19714

Srour, F.J., *Dissecting Drayage: An Examination of Structure, Information, and Control in Drayage Operations*, Promoter(s): Prof.dr. S.L. van de Velde, EPS-2010-186-LIS, http://hdl.handle.net/1765/18231

Stam, D.A., *Managing Dreams and Ambitions: A Psychological Analysis of Vision Communication*, Promoter(s): 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,* Promoter(s): Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2008-145-STR, http://hdl.handle.net/1765/13943

Sweldens, S.T.L.R., *Evaluative Conditioning 2.0: Direct versus Associative Transfer of Affect to Brands,* Promoter(s): Prof.dr. S.M.J. van Osselaer, EPS-2009-167-MKT, http://hdl.handle.net/1765/16012

Szkudlarek, B.A.*, Spinning the Web of Reentry: [Re]connecting reentry training theory and practice*, Promoter(s): Prof.dr. S.J. Magala, EPS-2008-143-ORG, http://hdl.handle.net/1765/13695

Teixeira de Vasconcelos, M., *Agency Costs, Firm Value, and Corporate Investment*, Promoter(s): Prof.dr. P.G.J. Roosenboom, EPS-2012-265-F&A, http://hdl.handle.net/1765/37265

Tempelaar, M.P., *Organizing for Ambidexterity: Studies on the Pursuit of Exploration and Exploitation through Differentiation, Integration, Contextual and Individual Attributes,* Promoter(s): Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-191-STR, http://hdl.handle.net/1765/18457

Tiwari, V., *Transition Process and Performance in IT Outsourcing: Evidence from a Field Study and Laboratory Experiments,* Promoter(s): Prof.dr.ir. H.W.G.M. van Heck & Prof.dr. P.H.M. Vervest, EPS-2010-201-LIS, http://hdl.handle.net/1765/19868

Tröster, C., *Nationality Heterogeneity and Interpersonal Relationships at Work*, Promoter(s): Prof.dr. D.L. van Knippenberg, EPS-2011-233-ORG, http://hdl.handle.net/1765/23298

Tsekouras, D., *No Pain No Gain: The Beneficial Role of Consumer Effort in Decision Making*, Promoter(s): Prof.dr.ir. B.G.C. Dellaert, EPS-2012-268-MKT, http://hdl.handle.net/1765/1

Tuk, M.A., *Is Friendship Silent When Money Talks? How Consumers Respond to Word-of-Mouth Marketing*, Promoter(s): Prof.dr.ir. A. Smidts & Prof.dr. D.H.J. Wigboldus, EPS-2008-130-MKT, http://hdl.handle.net/1765/12702

Tzioti, S., *Let Me Give You a Piece of Advice: Empirical Papers about Advice Taking in Marketing,* Promoter(s): Prof.dr. S.M.J. van Osselaer & Prof.dr.ir. B. Wierenga, EPS-2010-211-MKT, hdl.handle.net/1765/21149

Vaccaro, I.G., *Management Innovation: Studies on the Role of Internal Change Agents,* Promoter(s): Prof.dr. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-212-STR, hdl.handle.net/1765/21150

Verheijen, H.J.J., *Vendor-Buyer Coordination in Supply Chains,* Promoter(s): Prof.dr.ir. J.A.E.E. van Nunen, EPS-2010-194-LIS, http://hdl.handle.net/1765/19594

Verwijmeren, P., *Empirical Essays on Debt, Equity, and Convertible Securities*, Promoter(s): Prof.dr. A. de Jong & Prof.dr. M.J.C.M. Verbeek, EPS-2009-154-F&A, http://hdl.handle.net/1765/14312

Vlam, A.J., *Customer First? The Relationship between Advisors and Consumers of Financial Products*, Promoter(s): Prof.dr. Ph.H.B.F. Franses, EPS-2011-250-MKT, http://hdl.handle.net/1765/30585

Waard, E.J. de, *Engaging Environmental Turbulence: Organizational Determinants for Repetitive Quick and Adequate Responses*, Promoter(s): Prof.dr. H.W. Volberda & Prof.dr. J. Soeters, EPS-2010-189-STR, http://hdl.handle.net/1765/18012

Wall, R.S., *Netscape: Cities and Global Corporate Networks,* Promoter(s): Prof.dr. G.A. van der Knaap, EPS-2009-169-ORG, http://hdl.handle.net/1765/16013

Waltman, L., *Computational and Game-Theoretic Approaches for Modeling Bounded Rationality*, Promoter(s): Prof.dr.ir. R. Dekker & Prof.dr.ir. U. Kaymak, EPS-2011-248-LIS, http://hdl.handle.net/1765/26564

Wang, Y., *Information Content of Mutual Fund Portfolio Disclosure*, Promoter(s): Prof.dr. M.J.C.M. Verbeek, EPS-2011-242-F&A, http://hdl.handle.net/1765/26066

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,* Promoter(s): Prof.dr. H.W. Volberda, EPS-2009-173-STR, http://hdl.handle.net/1765/16182

Wubben, M.J.J., *Social Functions of Emotions in Social Dilemmas*, Promoter(s): Prof.dr. D. De Cremer & Prof.dr. E. van Dijk, EPS-2009-187-ORG, http://hdl.handle.net/1765/18228

Xu, Y., *Empirical Essays on the Stock Returns, Risk Management, and Liquidity Creation of Banks*, Promoter(s): Prof.dr. M.J.C.M. Verbeek, EPS-2010-188-F&A, http://hdl.handle.net/1765/18125

Yang, J., *Towards the Restructuring and Co-ordination Mechanisms for the Architecture of Chinese Transport Logistics*, Promoter(s): Prof.dr. H.E. Harlambides, EPS-2009-157-LIS, http://hdl.handle.net/1765/14527

Yu, M., *Enhancing Warehouse Performance by Efficient Order Picking*, Promoter(s): Prof.dr. M.B.M. de Koster, EPS-2008-139-LIS, http://hdl.handle.net/1765/13691

Zhang, D., *Essays in Executive Compensation*, Promoter(s): Prof.dr. I. Dittmann, EPS-2012-261-F&A, http://hdl.handle.net/1765/32344

Zhang, X., *Scheduling with Time Lags,* Promoter(s): Prof.dr. S.L. van de Velde, EPS-2010-206-LIS, http://hdl.handle.net/1765/19928

Zhou, H., *Knowledge, Entrepreneurship and Performance: Evidence from Country-level and Firm-level Studies,* Promoter(s): Prof.dr. A.R. Thurik & Prof.dr. L.M. Uhlaner, EPS-2010-207-ORG, http://hdl.handle.net/1765/20634

Zwan, P.W. van der, *The Entrepreneurial Process: An International Analysis of Entry and Exit*, Promoter(s): Prof.dr. A.R. Thurik & Prof.dr. P.J.F. Groenen, EPS-2011-234-ORG, http://hdl.handle.net/1765/23422

Zwart, G.J. de, *Empirical Studies on Financial Markets: Private Equity, Corporate Bonds and Emerging Markets*, Promoter(s): Prof.dr. M.J.C.M. Verbeek & Prof.dr. D.J.C. van Dijk, EPS-2008-131-F&A, http://hdl.handle.net/1765/12703

Erasmus Research Institute of Management - ERiM

DELAY MANAGEMENT AND DISPATCHING IN RAILWAYS

Passenger railway transportation plays a crucial role in the mobility in Europe. Since the privatization of the railway sector in the 90s, passenger satisfaction has become an important performance indicator in this sector. A key aspect for passengers is the reliability of transfers between trains. When a train arrives at the station with a delay, passengers might miss their connection if the next train departs on time. These passengers then prefer the connecting train to wait, but this introduces delays for many other passengers. Delay Management is a field in railway operations that deals with this situation. It determines whether a connecting train should wait for the passengers that arrive with a delayed train or should depart on time.

In this thesis, we apply techniques from Operations Research to develop models and solution approaches for Delay Management. The objective in our models is the minimization of passenger delay. First, we extend the classical delay management model with passenger rerouting. This allows us to compute the exact delays for passengers. We develop an exact algorithm and several heuristics to solve this extension. Then, we incorporate the limited capacity of the stations in our models. Stations are the bottlenecks of the railway infrastructure, where delays of one train can easily propagate to other trains. When optimizing the wait-depart decisions, these secondary delays should be considered. We therefore develop an integrated model that includes headway constraints for trains on the same track in the station and an iterative approach that evaluates the timetable microscopically.

Design & layout: B&T Ontwerp en advies (www.b-en-t.nl) Print: Haveka (www.haveka.nl)

ERiM

The Erasmus Research Institute of Management (ERIM) is the Research School (Onderzoekschool) in the field of management of the Erasmus University Rotterdam. The founding participants of ERIM are the Rotterdam School of Management (RSM), and the Erasmus School of Economics (ESE). ERIM was founded in 1999 and is officially accredited by the Royal Netherlands Academy of Arts and Sciences (KNAW). The research undertaken by ERIM is focused on the management of the firm in its environment, its intra- and interfirm relations, and its business processes in their interdependent connections.

The objective of ERIM is to carry out first rate research in management, and to offer an advanced doctoral programme in Research in Management. Within ERIM, over three hundred senior researchers and PhD candidates are active in the different research programmes. From a variety of academic backgrounds and expertises, the ERIM community is united in striving for excellence and working at the forefront of creating new business knowledge.

ERIM PhD Series
Research in Management

Erasmus Research Institute of Management - ERiM
Rotterdam School of Management (RSM)
Erasmus School of Economics (ESE)
Erasmus University Rotterdam (EUR)
P.O. Box 1738, 3000 DR Rotterdam,
The Netherlands

Tel. +31 10 408 11 82
Fax +31 10 408 96 40
E-mail info@erim.eur.nl
Internet www.erim.eur.nl