GERT-JAAP POLINDER

New models and applications for railway timetabling

New Models and Applications for Railway Timetabling

New Models and Applications for Railway Timetabling

Nieuwe modellen en toepassingen voor spoorwegdienstregelingen.

Thesis

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

Prof.dr. R.C.M.E. Engels

and in accordance with the decision of the Doctorate Board.

The public defence shall be held on

Friday 18 December 2020 at 9:30 hours

by

Gerrit Jacob Polinder
born in Zwolle, the Netherlands.

Erasmus University Rotterdam

Doctoral committee

Doctoral dissertation supervisor:
Prof.dr. D. Huisman

Co-supervisor:
Dr. M.E. Schmidt

Other members:
Prof.dr. R.M.P. Goverde
Prof.dr. R.A. Zuidwijk
Dr. V. Cacchiani

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: www.erim.eur.nl

ERIM Electronic Series Portal: repub.eur.nl

ERIM PhD Series in Research in Management, 514
ERIM reference number: EPS-2020-514-LIS
ISBN 978-90-5892-591-6

Cover image: ©Arthur Scheltes

Cover design: PanArt, www.panart.nl

This publication (cover and interior) is printed by Tuijtel on recycled paper, BalanceSilk®.
The ink used is produced from renewable resources and alcohol free fountain solution.
Certifications for the paper and the printing production process: Recycle, EU Ecolabel, FSC®, C007225.
More info: www.tuijtel.com

FSC www.fsc.org
RECYCLED
Paper made from recycled material
FSC® C007225

To my twin boys Jesse and Nathan.
Your laughter and love motivated me to complete this work.

Acknowledgements

As a student, I used the train on a daily basis, especially in the 'Randstad', one of the most busy sections of the Dutch railway network. The complexity of the network and how it is organized, scheduled and managed, fascinated me. I was therefore very pleased when I got the opportunity to do my graduation research at Netherlands Railways (NS), because I would be working on these networks myself. I enjoyed carrying out my research, but obtaining a PhD was something that I had never imagined. If the late Leo Kroon would not have asked me to become one of his PhD students, I probably would not be doing what I am doing right now: completing my dissertation. Several other people have contributed to me reaching this point, and I would like to take the opportunity to express my gratitude.

First of all, I am grateful to my daily supervisor Marie Schmidt. You were always open and available for discussions, had many interesting suggestions for further steps in our research and somehow managed to make sense of my often vague ideas. I particularly appreciate your thorough reviews of my written pieces of text. You managed to both see the broad lines in the paper as well as the details. You have been a great source of inspiration and motivation to write papers, also when I'd rather do some programming.

Secondly, I am very grateful to Dennis Huisman, who stepped in as my promotor when Leo unfortunately passed away in 2016. You always had the broad lines of our research in mind as well as the goal that we were aiming for. Your comments were very helpful to put our research in perspective and what the implications for the practice of railway operations were. Thank you for pushing me to keep on schedule and to submit my papers.

I want to thank Thomas Breugem, Twan Dollevoet, Gabór Maróti, Karen Aardal and Marco Molinaro as well for working together on some of the chapters of this dissertation.

I am grateful to Rob Zuidwijk, Rob Goverde and Valentina Cacchiani for being part of my inner committee, for evaluating this work and for providing helpful suggestions for improvement. Karen Aardal and Twan Dollevoet agreed to participate in the opposition during the defence, which I greatly appreciate.

The fact that Valentina agreed to be part of my inner committee is extra special for me, because we have actually worked together on one of the chapters in this dissertation. Thank you very much for hosting me at the oldest university of Europe in the beautiful city of Bologna. I really enjoyed my time in Italy and working with you was a true pleasure. I also want to thank your colleagues Silvano Martello, Michele Monaci and Enrico Malaguti for the lunches we enjoyed together.

I am grateful to my colleagues at RSM and ESE, for the nice time that I had there and for the 'broodjes hete kip' that we shared. I specifically want to mention my fellow railway colleagues: Thomas, Johann, Rowan, Rolf, Naut, Paul, Twan and Shadi. It was very interesting to discuss our research with fellow railway researchers and to improve the quality of our papers. I enjoyed the courses we took and the conferences we visited together.

A very important part of my life as a PhD-student was the department of Process quality and Innovation (π) of NS. It was a lot of fun to work at your office and to enjoy the 'taartmomenten' that Anneke gladly scheduled every now and then. Of course, the yearly AIVD Kerstpuzzel can not be left unmentioned. The many hours that we have spent by joining insights and solving these nice non-trivial challenges definitely were not a waste of time. For my research, I mainly worked together with Pieter-Jan, Joël and Gabór. Thank you for the many discussions we have had, you have answered a lot of practice-related questions for me. I specifically want to thank Gabór here. Your mind was always open for discussions and for answering questions. The answers that you gave generally extended far beyond an answer to the actual question, but nonetheless turned out to be very useful. You taught me about data structures and how to write code efficiently (number of bugs per minute?) and together we have 'verpespt' quite some models.

Although I enjoyed investigating railway planning problems, luckily there is more in life than work. I am very grateful to my family for supporting me on this journey and for making me focus on other things once in a while. I thank Arthur Scheltes for the nice cover he designed for this dissertation. My friends Harold de Jong and Cornel Kok agreed to be my paranymphs and I am honoured to have them standing next to me during the defence ceremony. Not only my supervisors have provided feedback on my work, I also want to mention Cornel, Dik, Harold and Annemarie for checking some parts of this manuscript, thank you! Cornel, thank you for the many rubber ducking sessions that we had. Finally, Chantine, Jesse and Nathan, you are the best that has ever happened to me. Thank you for your love and laughter, and for distracting me from work. My deepest words of gratitude are for Chantine. Thank you for your patience and support to finish this dissertation and most of all for being you.

Most important of all, I want to thank God for giving me wisdom and strength to do my work: SOLI DEO GLORIA.

Waarder, October 2020,
Gert-Jaap Polinder

Table of contents

Chapter 1

Introduction

On an average working day, approximately 750.000 passengers use the train in the Netherlands, mainly to travel to work or to school. These passengers are transported by over 5000 scheduled sprinter and Intercity services per day (NS, 2019), operated by the largest public transport railway operator in the Netherlands: NS (Nederlandse Spoorwegen, the Dutch name for Netherlands Railways). Next to NS, there are a number of other companies operating trains on the Dutch railway network, both for the transport of passengers and for freight, leading to an even more congested network. Designing a timetable, i.e., scheduling all these services, such that they can be operated safely and efficiently in such a crowded network is an extremely challenging problem and covers months up to years of preparations and planning. This is one of the reasons that Decision Support Systems have been developed at NS to support the design of a timetable (Hooghiemstra et al., 1999; Kroon et al., 2009; Schrijver and Steenbeek, 1993). However, due to the strong increase of railway services on the network, several systems can no longer be used to compute a timetable.

In this thesis, we develop new models and algorithms to support the design of a railway timetable. Some of our methods are designed for a period long before the actual operation of the timetable will take place. They can be used to develop completely new timetables. It now becomes easier to select possible scenarios to investigate further, thus speeding up the timetable design process. Two other methods are designed for a period relatively close to the actual operation. The first method aims at finding a timetable when some constraints that the timetable has to satisfy are in themselves in contradiction. The second method aims at computing a delay-resistant timetable.

Railway timetabling is an academically very challenging problem. This is motivated by the large amount of literature focussing on this problem that has been produced in the last decades, see for example Cacchiani and Toth (2012), Hansen and Pachl (2014), Caimi et al. (2017) and Borndörfer et al. (2018) for overviews. However, more research is needed to design new models and algorithms for railway timetabling, as is indicated by the aforementioned references. Next to this, the increased utilisation of the networks lead to new challenges that have to be overcome. In this thesis, the models and algorithms are evaluated on the railway network in the Netherlands, although they are not limited to use only in the Netherlands. Instead, many other countries have a very crowded railway network, as is clearly shown by Boston Consulting Group (2017). They also face the issue of Decision Support Systems that can no longer be used, for example in Germany: Deutschland-TAKT (2019), Großmann et al. (2015).

1.1 Railway Planning Problems

In order to understand the context in which a railway timetable is designed, it is necessary to know what is considered as input to the timetabling problem. Furthermore, it is important to know how the other railway planning problems are related to the timetabling problem, as timetabling is not the only planning problem that has to be solved in order to provide a good service to the passengers.

The main railway planning problems are line planning, timetabling, rolling stock scheduling and crew planning. There are several ways in which these different planning problems can be classified. A common classification is to consider the planning horizon of the different planning problems, and in that case a strategic, a tactical and an operational planning phase can be identified (Abbink, 2014; Huisman et al., 2005). The strategic planning phase encompasses a time horizon of two to ten years before the actual operation. The tactical phase covers the period of one to two years before operation, while the operational phase covers the period of a few weeks up to months before the actual operation. Real time planning can be considered as well, but this mainly covers the management of the daily processes and not so much planning in advance. An overview of how we position the different planning problems according to the planning horizon is shown in Figure 1.1. A solution of one problem serves as input for the next problem, i.e., a timetable serves as input for the rolling stock scheduling problem. Note that, although the planning problems are often considered in this order (Borndörfer et al., 2018), it is not the only way in which they can be considered. An alternative can be found in Schöbel (2017). We now briefly describe the content of each of the aforementioned problems.

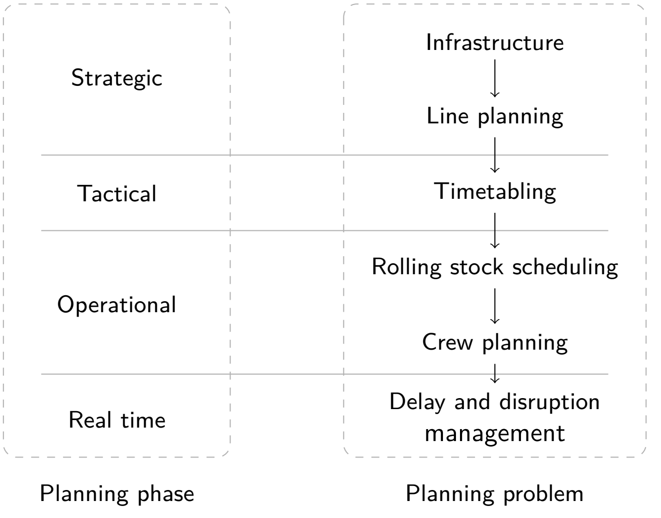


Figure 1.1: Railway planning problems

Railway planning starts with the design of or the extensions to an infrastructural network. New infrastructure increases the network capacity and hence influences the services provided. Infrastructure is generally considered on a macroscopic, a mesoscopic or a microscopic level (Goverde et al., 2016). The macroscopic level contains the least amount of detail, the microscopic level the most. Macroscopic level infrastructure is generally used a long time before the actual operation. Closer to the operation, the mesoscopic and microscopic level are more relevant.

The line planning problem is the problem of selecting a set of train lines that have to be operated on the infrastructure network. Each train line consists of a route through the network, a list of stations where the train stops and a frequency at which the line has to be operated. Overviews of approaches to solve the line planning problem are provided by Kepaptsoglou and Karlaftis (2009) and Schöbel (2012). After the line planning problem, a timetable is generated based on the line plan. More details about timetabling are given in Section 1.2.

Having a timetable, physical train units must be assigned to the trips that have to be operated. This is called the rolling stock scheduling. The objective here is to schedule the train units in such a way that capacity is sufficient to accomodate the passengers, while keeping the operators costs low, as rolling stock is one of the most expensive parts of the operations in the rail sector. Examples of creating the schedules of rolling stock in rail networks can for example be found in Abbink et al. (2004), Fioole et al. (2006), Maróti (2006) and Lin and Kwan (2014).

After the rolling stock has been scheduled, the crew is scheduled. Crew scheduling is the problem of assigning tasks to personnel. The main restrictions that must be satisfied here vary per country and railway company. In the Netherlands, important restrictions are those imposed in the collective labour agreement. Furthermore, a nice division of work has to be made. To date, crew planning is solved by a two-phase approach: first crew scheduling, and then crew rostering (Abbink, 2014). In crew scheduling, the days of work have to be constructed, and in crew rostering the constructed duties have to be assigned to the crew members (Breugem, 2020). Approaches to integrate the two phases can be found in Breugem (2020).

Finally, everyday operations never go as planned. Therefore, adjustments must be made on a daily basis as soon as disturbances occur. The timetable has to be updated, and rolling-stock and crew must be rescheduled. Approaches to do so can be found in e.g. Veelenturf (2014). To some extent, disturbances and disruptions can be anticipated, and it can be attempted to build schedules in a way that makes them resilient towards these (cf. Cacchiani et al., 2008a; Lusby et al., 2018)

Railway planning is traditionally performed sequentially: First a line plan is determined, then a timetable is designed, followed by scheduling the rolling stock and the crew, i.e., the order that is displayed in Figure 1.1. However, this does lead to suboptimal solutions. Or, even worse, a solution for one planning problem can turn out to be infeasible for the next problem. As an example, it can occur that in the line planning phase a line plan is determined that can never be scheduled in the timetabling phase, as there are too many trains sharing a part of the infrastructure.

One option to avoid these 'mis-connections' between planning problems is to integrate them into one problem. In the literature, there are several attempts integrating two or more planning problems. For example, Schöbel (2015) integrate line planning and timetabling, Lübbecke et al. (2018a) integrate line planning, timetabling and rolling stock scheduling, and Huisman (2004) shows how the scheduling of personnel and rolling stock can be integrated, which is done especially at bus companies. As each of the aforementioned planning problems is already challenging to solve for practical cases, the integration of multiple planning problems leads to even more challenging problems to be solved, although in theory overall better solutions can be obtained. An interesting overview of solutions methods to various planning problems, the integration of them, and a framework for iterative solving these problems in different orders is given by Schöbel (2017).

1.2 Timetabling at Netherlands Railways

In timetabling, the task is to schedule a set of events, such that a set of restrictions is satisfied. As input, a line plan is given, stating which trains must be scheduled, and the output is a timetable. In macroscopic timetabling, each train line generates a set of events, corresponding to the arrivals and departures of this train at the stations it is visiting. For each of these events, a point in time has to be determined at which this event takes place.

Two main variants of timetables can be distinguished, namely periodic and aperiodic timetables. Whereas in an aperiodic timetable the timetable of every hour and day can be different, a periodic timetable has a certain regularity. That is, the timetable for a base period, generally one hour, is repeated multiple times, e.g., for a whole day. In the off-peak hours, some trains can be deleted and in the nights only a small number of trains are scheduled. Using a periodic timetable is often a design principle. Hybrid combinations of periodic and aperiodic timetables can be used as well (Robenek et al., 2017). The advantage of a periodic timetable is that the timetable is relatively easy to remember and only a timetable for one base period has to be designed. Periodic timetables are common in many European countries, also this thesis focusses solely on periodic timetabling.

A timetable has to satisfy many restrictions, also referred to as *activities* or *constraints*. If a timetable exists satisfying all constraints, this timetable is called *feasible*, if not, it is called an *infeasible* timetable. According to Caimi et al. (2017), a model that is commonly used to generate a periodic timetable is the Periodic Event Scheduling Problem (PESP) as introduced in Serafini and Ukovich (1989). In this model, all activities are of a specific form and restrict the time difference between pairs of events to be in a given (periodic) interval. The PESP framework can be used to model practically all constraints that a timetable has to satisfy (Liebchen and Möhring, 2007). Examples include driving activities (restricting the time difference between a departure and the next arrival), dwell activities (restricting the time a train dwells at a station), transfer activities (to guarantee a good transfer time from one train to another) and safety activities (to guarantee a safe operation of a timetable). As an example, a driving activity can be used to restrict the time difference between the departure of an Intercity train from Rotterdam Central Station and the next departure at Rotterdam Alexander to be at least 8 minutes, and at most 10 minutes. More details about the PESP and its modelling framework are provided in Chapter 2. Although finding a feasible solution to a PESP problem is not an easy task, many techniques are available to find such a feasible schedule for practical cases, often in a short time (cf. Caimi et al., 2017).

The task of finding a timetable that is not only feasible but optimal with respect to some objective function is much more challenging. An objective function assigns a value to a timetable, based on characteristics of the timetable. The task then is to find the best timetable according to these objective values.

There are several options for an objective function. First of all, we have the minimisation of travel time of passengers in the timetable. Such an objective function is used in Chapters 3 and 4 of this thesis. Travel time is not the only aspect of a timetable that passengers prefer to have, they also like a timetable to be resilient, i.e., to 'absorb' delays that occur in everyday practice. A method to achieve this is to add time supplements to the trips of the trains, such that a small timebuffer is created and delays do not propagate throughout the network. Overviews on how resilient timetables can be created are for example provided by Goerigk and Schöbel (2010), Cacchiani and Toth (2012) and Lusby et al. (2018). In Chapter 6 of this thesis, an approach is given for dealing with periodically reoccurring disturbances.

A third option for an objective function can be used when no feasible timetable exists satisfying all the constraints. Then a timetable is aimed for that satisfies as many constraints as possible, or that modifies the constraints as little as possible, such that they allow for a feasible solutions to exist. This is done in Chapter 5 of this thesis.

More (mathematical) details about the timetabling model, including a motivation of the difficulties, solution techniques and a practical example, are provided in Chapter 2.

1.3 Contributions

The contributions of this thesis are threefold. First of all, we propose several novel optimization problems that are aimed at improving the timetable design. Solving these problems leads to better motivated decision making and insights into the timetable structures. In Chapter 3, an optimization problem is proposed for the strategic planning phase, to compute what a timetable would ideally look like from a passengers perspective. To the best of our knowledge, we are the first in approaching this problem in such a systematic way. It is useful for determining new ways of scheduling the trains and identifying important transfer and synchronisation options. The same model is used in Chapter 4, but now more restrictions have to be incorporated, making it a more challenging problem. Chapter 5 discusses the problem of finding a feasible timetable, when the constraints are such that a feasible solution does not exist. This involves updating the constraints as little as possible. Chapter 6 discusses the problem of computing a periodic timetable taking into account periodic disturbances. Handling the so-called 'robustness' already in the design of a timetable has not received much attention in the literature, especially not when the disturbances are assumed to be periodic. Most existing approaches start from a given timetable, and try to make that timetable robust, whereas we incorporate the robustness-concept immediately in the first step of the timetable-design.

Secondly, for each of the proposed problems, we developed a solution approach. In Chapter 3, a heuristic method is developed, in order to compute good solutions even for networks of a national scale, to create relevant insights. This heuristic is also applied in Chapter 4. Furthermore, in this chapter an iterative framework is proposed to combine the method of Chapter 3 with an adapted version of another existing method and to improve the obtained solutions. Chapter 5 proposes a heuristic method to iteratively search for conflicts in the network and to resolve them as efficiently as possible. Hereby we balance the quality of the solutions and the time needed to compute them. In Chapter 6, we use techniques from the literature on robust optimization, in order to compute a robust timetable. More specifically, we use a linear decision rule, reformulation techniques and cutting-plane methods to find solutions.

Finally, we evaluate all the proposed optimization problems and our approaches on real world data from NS. Because of this, the implications for the practice of designing timetables can be evaluated. Decisions that are made regarding the design of networks can now be better motivated and the throughput time of generating a new railway timetable can be reduced.

1.4 Thesis Outline

The main topic of this thesis is the design of models and algorithms for periodic timetabling. Although timetabling is assigned to the tactical level (Figure 1.1), we also consider it in other planning phases. Figure 1.2 provides a schematic overview where each of the chapters in this thesis is positioned according to its objective and planning phase.

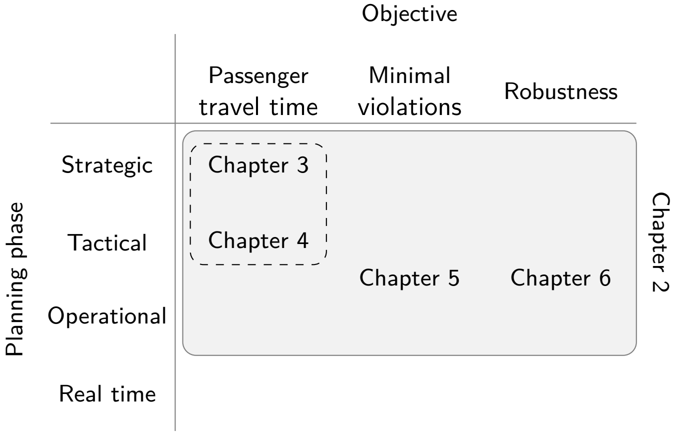


Figure 1.2: Schematic overview of the chapters in this thesis.

Chapter 3 considers passenger oriented timetabling in the strategic planning phase, because it is aimed for designing timetables a long time before the actual operation, where infrastructure constraints are not important. Chapter 4 has the same objective, but now in the tactical planning phase, here infrastructure restrictions play an important role. Chapters 5 and 6 can both be considered in the tactical and operational planning phases and are therefore positioned in the middle of these phases. The reason for this is that they are mainly used when a large part of the timetable already is constructed, and only smaller changes are made. In Chapter 5, the objective is to find a timetable that violates the restrictions as litte as possible. In Chapter 6, we compute a timetable that is robust against periodic disturbances.

Each of the aforementioned chapters builds upon the Periodic Event Scheduling Problem (PESP), which is explained in Chapter 2. This problem can be considered with practically any objective function, and in the strategic to operational planning phases. In real time (re)scheduling, the periodicity is less important and other methods can be more suitable, so PESP is not so relevant there.

Each chapter can be read as a separate entity. However, in order to understand the mathematical models proposed in the chapters, understanding PESP is very important. This is the underlying timetabling model that reappears in every chapter, so it is recommended to read Chapter 2 first. Next, as already indicated in Figure 1.2, Chapters 3 and 4 are closely related: Chapter 4 builds upon Chapter 3, so it is recommended to read these chapters in that order. Chapters 5 and 6 can both easily be read as independent chapters.

In the following, we briefly summarize Chapters 3–6. As each of these chapters is a modification of papers (about to be) submitted to academic journals, we provide the current status of each publication. Chapter 2 is a general overview about timetabling models that is mainly based on Peeters (2003). The work in Chapter 3 and 5 has been carried out independently under close supervision of the mentioned co-authors. Chapter 4 is done together with Valentina Cacchiani and Chapter 6 with Thomas Breugem, both under close supervision of the other mentioned co-authors.

Chapter 3: *G.J. Polinder, M.E. Schmidt, and D. Huisman: "Timetabling for strategic passenger railway planning", currently in second round of review at Transportation Research Part B: Methodological. This paper has been awarded the third price at the INFORMS RAS 2019 student paper competition and ranked third at the selection of best papers of Rail Nörrkoping 2019.*

Timetables are normally designed in the tactical planning phase (see Figure 1.1), taking into account a given line plan, safety restrictions arising from infrastructural constraints, as well as regularity requirements and bounds on transfer times. In this chapter, however, we propose a timetabling approach that is aimed at decision making in the strategic planning phase, to determine an outline of a timetable that is good from a passengers perspective. Instead of including explicit synchronization constraints between train runs (as most timetabling models do), we include the adaption time (waiting time at the station of origin) in the objective function to ensure regular connections between passengers' origins and destinations. We model the problem as a mixed integer quadratic program and linearise it. Furthermore we propose a heuristic to generate good starting solutions. We illustrate the trade-offs between dwell times and regularity of trains in two case studies based on the Dutch railway network.

Chapter 4: *G.J. Polinder, V. Cacchiani, M.E. Schmidt, and D. Huisman: "An iterative heuristic for passenger-centric train timetabling with integrated adaption times", currently under review at Transportation Research Part B: Methodological.*

In this chapter, we aim at constructing a timetable that minimizes average perceived passenger travel time, which, in addition to the in-train and transfer times, includes the adaption time (waiting time at the station of origin). Adaption time minimization allows us to avoid strict frequency regularity constraints and, at the same time, to ensure regular connections between passengers' origins and destinations. Besides considering safety restrictions (i.e., headway times, overtaking and crossing constraints), passenger routing, based on origin-destination demand pairs, must be taken into account when building the timetable.

This problem can be modelled as an extension of a Periodic Event Scheduling Problem (PESP) formulation, but cannot be directly solved by a general-purpose solver for our real-size instances. In this chapter, we propose a heuristic approach consisting of two phases that are executed iteratively. First, we solve a simplified model, and determine an ideal timetable that minimizes the average perceived passenger travel time but neglects safety restrictions. Then, a Lagrangian-based heuristic modifies train departure and arrival times as little as possible, in order to obtain a timetable that is feasible with respect to safety constraints. The obtained timetable is then evaluated to compute the resulting average perceived passenger travel time, and a feedback is sent to the Lagrangian-based heuristic so as to possibly improve the obtained timetable from the passenger perspective, while still respecting safety constraints. We have tested the proposed iterative heuristic approach on real-life instances of Netherlands Railways, showing that it converges to a feasible timetable very close to the ideal one.

Chapter 5: *G.J. Polinder, L.G. Kroon, K.I. Aardal, M.E. Schmidt, and M. Molinaro: "Resolving infeasibilities in railway timetabling instances", in preparation for journal submission. This is an extension of Polinder (2015).*

One of the key assumptions of timetabling algorithms is that a solution exists that meets the pre-specified constraints, like driving times, transfer constraints and headway constraints. If this assumption is satisfied, in most cases a timetable can be found rapidly. Nowadays, railways are being used more intensively, which leads to a higher utilization of the network. Due to this increased utilisation, capacity conflicts occur, so that no feasible solution to the timetabling models can be found, without making subtle but non-trivial changes to the initial input. Resolving these conflicts is essential for railway companies with high utilization of infrastructure. In this chapter, we consider infeasible timetabling instances together with a list of allowed modifications of the constraints. We iteratively identify local conflicts in these instances and resolve them by adapting some of the constraints, until there are no more conflicts. The adaptations of the constraints are changes in the right-hand sides that we try to make as small as possible but that resolve the infeasibility. We empirically show that our method can be improved by enriching the initial minimal conflicts found with more timetabling constraints. In order to keep the problems tractable, an iterative procedure is used to find solutions to subproblems corresponding to conflicts in the complete timetabling instance. In a case study on instances from the Dutch railway network, we show that these instances can be made feasible within a few minutes.

Chapter 6: *G.J. Polinder, T. Breugem, T. Dollevoet, and G. Maróti: "An adjustable robust optimization approach for periodic timetabling", published in Transportation Research Part B: Methodological.*

In this chapter, we consider the Robust Periodic Timetabling Problem (RPTP), the problem of designing a periodic timetable that can easily be adjusted in case of small periodic disturbances. We develop a solution method for a parametrized class of uncertainty regions. This class relates closely to uncertainty regions known in the robust optimization literature, and naturally defines a metric for the robustness of the timetable. The proposed solution method combines a linear decision rule with well-known reformulation techniques and cutting-plane methods. We show that the RPTP can be solved for practical-sized instances by applying the solution method to practical cases of Netherlands Railways (NS). In particular, we show that the trade-off between the efficiency and robustness of a timetable can be analysed using our solution method.

Chapter 2

Periodic Event Scheduling Problem

The underlying theme of this thesis are models and algorithms for periodic railway timetabling in various stages of railway planning. Regardless of the stage in which timetabling is considered, the basis of all timetabling models in this thesis is the *Periodic Event Scheduling Problem (PESP)* as introduced by Serafini and Ukovich (1989). Notation and details of this model are discussed in Section 2.1. Section 2.2 discusses several solution techniques to solve PESP. A practical example is provided in Section 2.3. A more extensive description of PESP and several of its properties can be found in Peeters (2003).

2.1 Definition

The timetabling problem is traditionally preceded by the line planning problem in railway planning, see also Section 1.1. In the line planning problem, a set of train lines $\mathcal{L}$ is determined that have to be scheduled on the infrastructure network. This set of lines is considered as input to the timetabling problem. A train line $\ell \in \mathcal{L}$ is a combination of a train type (e.g. Intercity of local train), a route, a list of stations where the train stops and a frequency at which the line is to be operated per cycle period. We assume that each train line is operated in both directions.

There are generally three levels of detail at which an infrastructure network can be considered: the macroscopic, mesoscopic and microscopic level (Goverde et al., 2016). In (tactical) timetabling, the infrastructure is often considered on the macroscopic level, that is, as railway stations with a number of tracks connecting them. Further details like block sections and signalling systems are not important in the tactical planning stage and can be included in a later planning stage (Radtke, 2014, Chapter 3.4).

In the case of a macroscopic network, an *event-activity network* $G = (V, A)$ can be generated based on a line plan. In this network, the nodes represent the set of departure and arrival events V, corresponding to the departures and arrivals of the trains at the various stations in the network. The events are periodically reoccurring with a cycle period T. For the experiments presented in this thesis, we take $T = 60$, that is, we aim at a cyclic timetable with a period of one hour, and events are scheduled in full minutes. The nodes are linked to each other by arcs, representing activities A. Each activity is a relation between a pair of events, with lower and upper bounds on their time difference. They can cover a wide range of restrictions, see for example Odijk (1996), Peeters (2003). Examples of the most commonly used activities include the following:

Drive activity This models the (minimum and possibly maximum) driving time between two stations, based on the distance between them and the maximum speed of the train. As an example, a drive activity can restrict the time difference between the departure of an Intercity train from Rotterdam Central Station and the next arrival at Rotterdam Alexander to be at least 8 minutes.

Dwell activity This restricts the time difference between the arrival of a train at a station and the time it departs. For small stations, this dwell time is mostly 0 or 1 minute, for larger stations this is usually four minutes.

Transfer activity In order to guarantee a good connection for passengers from one train to another, transfer activities can be used to limit the time between the arrival of one train at a station, and the departure of another train at the same station. This is the time that passengers who arrive with the first train have to alight from their train, change platforms, and board the second train.

Turnaround activity If a train has served a certain line and has reached its terminal station, the same train is often used to serve the line in the other direction as well. The minimum time restriction on this activity is at least the time that is needed for the driver to get to the other side of the train. Often, an upper bound is provided as well, in order to avoid long turnaround times, as a heuristic measure aimed at reducing the number of vehicles needed to operate the timetable and thus reduce rolling stock costs.

Safety activity A safety activity is used in order to guarantee a minimum headway time between trains, to prevent collisions on single track areas, and to prevent overtakings. For the interested reader, Section 2.A provides more details on the modelling of various satefy activities.

Some timetabling restrictions cannot be included in PESP, like symmetry restrictions. In a perfectly symmetric timetable, there exists a time at which all trains meet a train of the same line in the other direction. As a consequence, transfer times in both directions are the same and hence trips between any two stations have the same duration in both directions. A complete overview of what can and what can not be included in PESP is given by Liebchen and Möhring (2007).

The Periodic Event Scheduling Problem can now formally be stated as:

Definition 2.1 (PESP). *Given a set V of events, a set $A \subseteq V \times V$ of activities, intervals $[\ell_{ij}, u_{ij}]$ for all $(i, j) \in A$ and a period length T, the Periodic Event Scheduling Problem is to find a feasible periodic schedule, that is, find event times $\pi : V \to \{0, 1, \dots, T-1\}$ satisfying*

$$(\pi_j - \pi_i - \ell_{ij} \text{ modulo } T) + \ell_{ij} \in [\ell_{ij}, u_{ij}] \qquad \forall\, (i, j) \in A. \tag{2.1}$$

Note that any PESP activity can be scaled in such a way that $0 \leq \ell_{ij} < T$ (cf. Peeters, 2003). Next to this, one may assume that for each activity we have $u_{ij} - \ell_{ij} < T$, otherwise the activity would be redundant, as all time differences are allowed.

The general form (2.1) can be used to express all common timetabling constraints as listed above. As an example of such an activity, consider a drive activity that states that the time difference between arrival event π_2 and departure event π_1 should be at least 15 minutes, and at most 18 minutes. This constraint can be written as

$$(\pi_2 - \pi_1 - 15 \text{ modulo } 60) + 15 \in [15, 18], \tag{2.2}$$

and is visualized in Figure 2.1.

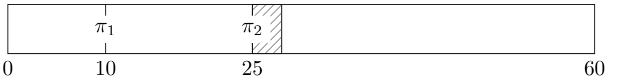


Figure 2.1: Example Periodic Constraint.

The highlighted area shows all feasible event times π_2, given the scheduled time $\pi_1 = 10$. For this activity, scheduling π_2 at minute 25 hence is a feasible solution.

2.1.1 Mixed Integer Programming Formulation

Equations (2.1) can be formulated as mixed-integer programming constraints by introducing a term Tp_{ij}, where p_{ij} is an integer variable, representing the modulo operator. T again is the cycle time. The result can then be written as

$$y_{ij} = \pi_j - \pi_i + Tp_{ij} \tag{2.3a}$$

$$\ell_{ij} \leq y_{ij} \leq u_{ij} \tag{2.3b}$$

$$p_{ij} \in \mathbb{Z}. \tag{2.3c}$$

The additionally introduced variable y_{ij} represents the *activity duration* for activity $(i,j) \in A$. As an example, if (i,j) is a transfer activity, y_{ij} denotes the duration of this transfer, i.e., how many minutes of transfer time are available.

In the above reformulation of a PESP-activity, no integrality restriction is set on the π-variable. Often integer values for π are desired as these variables represent the timetable itself. However, given a vector $p \in \mathbb{Z}^{|A|}$, the constraint matrix of (2.3) is totally unimodular, so a feasible solution exists with $\pi \in \{0, 1, \ldots, T-1\}^{|V|}$, assuming that $\ell, u \in \mathbb{Z}^{|A|}$ (Liebchen and Peeters, 2009).

A solution to the reformulated PESP activity as in (2.3) can easily be found when the values for the p-variables are known. In that case, the problem becomes a feasible differential problem, which is polynomially solvable (Rockafellar, 1998), while PESP is NP-complete (Serafini and Ukovich, 1989). Odijk et al. (2006) investigates the problem of finding several classes of timetables that are based on the different values the p-variables can have.

2.1.2 Cycle Periodicity Formulation

A formulation that is equivalent to (2.3) is the Cycle Periodicity Formulation (CPF) as introduced by Nachtigall (1999). This formulation is based on cycles in the (directed) graph representation of the PESP instance. Let $\mathcal{C}$ denote any cycle in the timetabling graph $G = (V, A)$. If we choose a direction in the cycle $\mathcal{C}$ in which the cycle is traversed, let $\mathcal{C}^+$ and $\mathcal{C}^-$ denote the set of arcs of this cycle that are traversed in the forward and backward direction, respectively. Starting at any node in this cycle, the cycle is traversed until we end up in the same node again. The sum of the activity durations in $\mathcal{C}^+$ minus the sum of the activity durations in $\mathcal{C}^-$ should be an integer multiple of T. This sum of the activity durations is denoted by $q_{\mathcal{C}}T$, where $q_{\mathcal{C}}$ is an integer variable. Giving this notation, the *Cycle Periodicity Formulation* is given as follows:

Definition 2.2 (CPF)**.** *Given a directed graph $G = (V, A)$ representation of a PESP instance as defined before, find y_{ij} for all $(i, j) \in A$ such that*

$$\sum_{(i,j)\in\mathcal{C}^+} y_{ij} - \sum_{(i,j)\in\mathcal{C}^-} y_{ij} = Tq_{\mathcal{C}} \qquad \forall\, \mathcal{C} \in \mathcal{B} \tag{2.4a}$$

$$\ell_{ij} \leq y_{ij} \leq u_{ij} \qquad \forall\, (i, j) \in A \tag{2.4b}$$

$$a_{\mathcal{C}} \leq q_{\mathcal{C}} \leq b_{\mathcal{C}} \qquad \forall\, \mathcal{C} \in \mathcal{B} \tag{2.4c}$$

$$y \in \mathbb{R}^{|A|}, \quad q \in \mathbb{Z}^{\kappa}, \tag{2.4d}$$

where $\mathcal{B}$ is the set of cycles in the graph, and $\kappa = |\mathcal{B}|$.

Observe that in (2.4c) we have introduced bounds on the $q_{\mathcal{C}}$ variables. These can be taken as $a_{\mathcal{C}} = -\infty$ and $b_{\mathcal{C}} = \infty$, however, tighter bounds can be computed based on the bounds of the activities that form the cycle as follows:

$$a_{\mathcal{C}} = \left\lceil \frac{1}{T} \left(\sum_{(i,j)\in\mathcal{C}^+} \ell_{ij} - \sum_{(i,j)\in\mathcal{C}^-} u_{ij} \right) \right\rceil, \tag{2.5a}$$

$$b_{\mathcal{C}} = \left\lfloor \frac{1}{T} \left(\sum_{(i,j)\in\mathcal{C}^+} u_{ij} - \sum_{(i,j)\in\mathcal{C}^-} \ell_{ij} \right) \right\rfloor. \tag{2.5b}$$

Although there is an exponential number of cycles in the graph, it is sufficient to require the cycle constraints (2.4a) and (2.4c) only for cycles in an integral cycle basis $\mathcal{B}$, i.e., a basis $\mathcal{B}$ such that every non-basis cycle is an integer linear combination of the cycles in $\mathcal{B}$ (Liebchen, 2003; Peeters, 2003). Such a cycle basis can for example be obtained by first finding a spanning tree in the graph. All the arcs that are not in the tree provide a cycle in the graph together with the tree-arcs. All these cycles together lead to an integral cycle basis of size $\kappa = |A| - |V| + 1$. Hence we need only a limited amount of cycles (Liebchen, 2003). If we define the width of a cycle as the possible values $q_{\mathcal{C}}$ can take, this is $b_{\mathcal{C}} - a_{\mathcal{C}} + 1$. The total width of a cycle basis can then be computed as

$$W(\mathcal{B}) = \prod_{\mathcal{C} \in \mathcal{B}} (b_{\mathcal{C}} - a_{\mathcal{C}} + 1). \tag{2.6}$$

If we find a basis $\mathcal{B}$ that reduces the value of $W(\mathcal{B})$ as much as possible, this leads to the smallest number of possible vectors $q \in \mathbb{Z}^{|\mathcal{B}|}$ and is a good candidate for a good cycle basis. However, to the best of our knowledge, it is still an open problem how to find a cycle basis that minimizes (2.6). An overview of the theory regarding cycle bases is given in Liebchen and Peeters (2009).

An advantage of the cycle periodicity formulation over the traditional PESP formulation is that CPF uses fewer integer variables: $|A| - |V| + 1$ (one for each cycle) in contrast to $|A|$ (one for each activity). Secondly, it uses equality constraints instead of inequality constraints. These two properties are generally beneficial when computing a solution by means of a branch-and-bound procedure. In general, using the CPF with a good cycle basis, feasible solutions are found sooner and the best bound is also improved more quickly when using an objective function. For a further discussion and comparison of different PESP formulations, see for example Liebchen et al. (2008).

2.1.3 Objective

Although PESP in itself is a feasibility problem, the formulation in (2.3) can be extended by including an objective function that can be optimized. Several types of objectives functions are possible, generally they are functions of the activity durations y. One example is the minimization of passenger travel time, which can be written as

$$\text{Minimize} \sum_{(i,j) \in A} w_{ij} y_{ij}, \tag{2.7}$$

where w_{ij} is a weight assigned to activity (i, j). These weights can for example represent the number of passengers using this activity. More sophisticated methods are possible to route the passengers in a timetabling model as we will see in Chapter 3 and 4, however, using the weights of activities is an approximation that is relatively easy to model and solve.

Another option for an objective function that is mentioned by Peeters (2003), is to minimize the initial constraint violations, in cases where the PESP instance does not allow for a feasible solution. Our approach to solve such a problem can be found in Chapter 5.

A last approach that we mention is to maximize the robustness of the timetable. For one approach, see Chapter 6, which also contains an overview of other approaches regarding robust timetabling. Examples of various objective functions applied to real life problems can for example be found in Caimi et al. (2017), Liebchen (2008), Liebchen and Peeters (2009), Nachtigall (1999), Peeters (2003).

2.2 Solution Techniques

In its original formulation, PESP is a feasibility problem and various approaches exist to solve it. One approach is to use an integer programming formulation, like (2.3) or (2.4). This is also used for computing timetables that are actually put into practice (Liebchen, 2008). A drawback of the integer programming approach is that is does not scale well: It is very challenging to find (good) solutions for large instances. Furthermore, because the constraints are 'big-M' constraints (the constraints either contain the term $p_{ij}T$ or $q_C T$), the LP-relaxation is usually very bad and proving optimality is a real challenge.

Where integer programming is a general approach that can be used to model many different problems, there exist more dedicated approaches to find solutions for PESP. Examples of these include constraint programming (Kroon et al., 2009; Schrijver and Steenbeek, 1993) or using a Satisfiability (SAT) solver after applying a polynomial transformation from PESP to SAT (Großmann et al., 2012; Kümmling et al., 2015a). These approaches generally do not consider an objective function. However, also when an objective function is considered, there are several dedicated approaches: using a modulo-simplex heuristic (Goerigk and Schöbel, 2013; Nachtigall and Opitz, 2008), a matching-approach (Pätzold and Schöbel, 2016), using a SAT approach (Matos et al., 2017), possibly combined with machine learning (Matos et al., 2020). If a feasible solution exists, this can often be found rapidly with the mentioned techniques.

2.3 Example

We now provide an example to demonstrate how, for a small network, a timetable can be found using the PESP formulation, with a cycle time of 60 minutes ($T = 60$). This example is based on an example used by Kroon (2015). Consider a network with stations $\mathcal{S} = \{S_1, S_2, \ldots, S_6\}$ and lines $\mathcal{L} = \{l_1, l_2, l_3\}$. All lines are operated in both directions. An overview of this network together with the routes of the lines is shown in Figure 2.2.

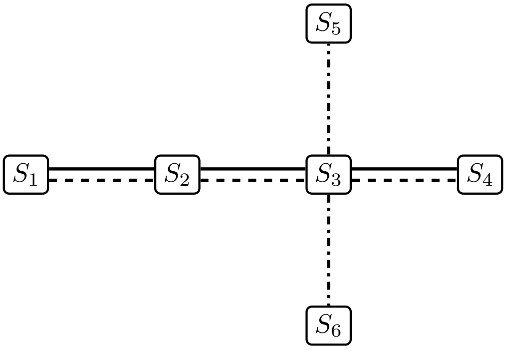


Figure 2.2: Example network

The solid line represents line l_1, which is an Intercity train line that travels between S_1 and S_4 and does not stop at S_2, only at the remaining stations. The dashed line represents line l_2, which is a local train line that has the same route as l_1, but has an additional stop in S_2. Finally, the dashdotted line is line l_3, which is an Intercity train line that travels between S_5 and S_6 and also stops at S_3. The trip times between the stations that the trains visit are shown in Table 2.1. In this network, connections have to be realized between two pairs of trains: from S_5 to S_2 (line l_3 to l_2) and vice versa, and from S_6 to S_4 (line l_3 to l_1) and vice versa. For the first connection, passengers need to change to a different platform, this connection should take between 5 and 7 minutes. The other connection between lines 1 and 3 is a cross-platform connection and should take between 2 and 3 minutes. Throughout the whole network, a headway time of at least 3 minutes is required between any two trains going in the same direction. Finally, we have the additional constraint that the train that operates a line in one direction also serves the line in the other direction, and a turn-around time of at least 7 minutes is needed at the end stations.

Combining all these restrictions leads to a PESP instance, in which 32 departure and arrival events have to be scheduled. These event times are restricted by 48

	$S_1 \longleftrightarrow S_2$	$S_2 \longleftrightarrow S_3$	$S_3 \longleftrightarrow S_4$
l_1	10 – 11	11 – 13	18 – 20
l_2	11 – 13	14 – 16	19 – 22

	$S_5 \longleftrightarrow S_3$	$S_3 \longleftrightarrow S_6$
l_3	20 – 22	31 – 33

Table 2.1: Trip times for the example instance

activities. These 48 activities cover 16 drive activities, 10 dwell activities, 12 headway activities, 6 activities ensuring the turn around time of the trains and 4 ensuring the connections. This leads to an event-activity network which can be pictured as a graph with 32 nodes and 48 arcs, as is done in Figure 2.3.

This figure displays the events that have to be scheduled as circles (nodes) and the activities as arcs in the graph. The nodes cover both arrivals and departures, the arrival nodes are shaded in Figure 2.3. The allowed time interval for each activity is shown next to it. On the left, the train lines are mentioned, which means that all nodes that are at the left of, for example, $l_1 \uparrow$ belong the line l_1 in the forward direction, i.e., driving from S_1 to S_4. $l_1 \downarrow$ shows the nodes for the same line in the backward direction. All nodes corresponding to one station are 'grouped' in shaded regions. At the top and bottom of the graph, the corresponding station is displayed.

In the graph, the drive and dwell activities are marked as solid arcs. Within the stations, the intervals are either $[0, 0]$, denoting that a dwell time of 0 minutes is allowed, i.e., the train does not stop, or $[1, 3]$. Safety activities are shown as dashed arcs. The interval for these activities is $[3, 57]$, which is the same as stating that the departures or arrivals can not be closer to each other than 3 minutes. There are 4 transfer activities in the network, denoted by dashdotted arcs. They impose a restriction on the time difference between an arrival and a departure of another train.

In this network, it would have been possible that passengers from S_6 do not transfer to l_1 but to l_2 instead, as both lines go to station S_4. Which of these options is better can only be determined once there is a timetable. It would have been possible to model a flexible connection, i.e., that one feeder train (line l_3) in this case, connects to one out of multiple other connection trains (lines l_1 or l_2 in this case). See Kroon et al. (2014) how this can be modelled. Finally, turnaround times for the trains are marked by densely dotted arcs. Note that they actually impose no restriction, as all time differences are allowed. It is possible to add a tighter upper bound, however,

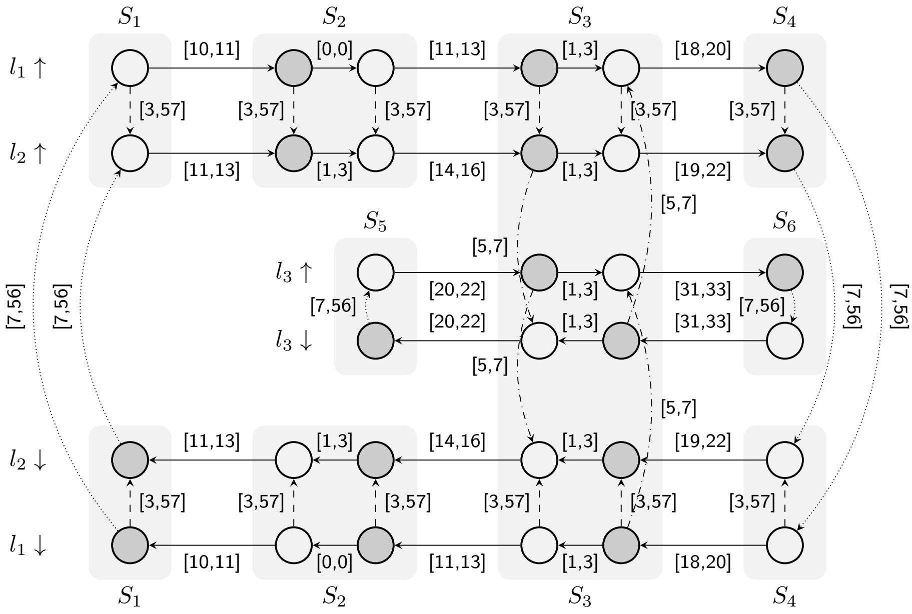


Figure 2.3: Network respresentation of the example instance

it is also possible to include the durations of these activities in an objective function which is to be minimized, in order to shorten the turn around times. Setting tight upper bounds for such activities can lead to infeasible timetabling instances as we will see in Chapter 6.

For this network, we compute a solution, in which we minimize the durations of the activities, each weighted by a certain weight factor. For drive and dwell activities, we set the weight to 1, for turn around activities, we set the weight to 0.9, to model that these are slightly less important. Finally, we set the weights of the other activities to zero. This leads to a solution that is displayed in Figure 2.4. The time at which an event takes place is shown in the nodes. Note that shifting each event by one minute again leads to a feasible timetable with the same quality, because all events are periodic events with a period of 60 minutes.

In Figure 2.4, several arcs are marked in bold. These arcs together form a spanning tree in the network. Adding any of the non-tree arcs to the spanning tree generates a cycle in the graph. These cycles together form a cycle basis for the Cycle Periodicity Formulation, see Section 2.1.2. In each of these cycles, the sum of the activity durations is a multiple of T, which was denoted by $q_C T$. Bounds on this variable

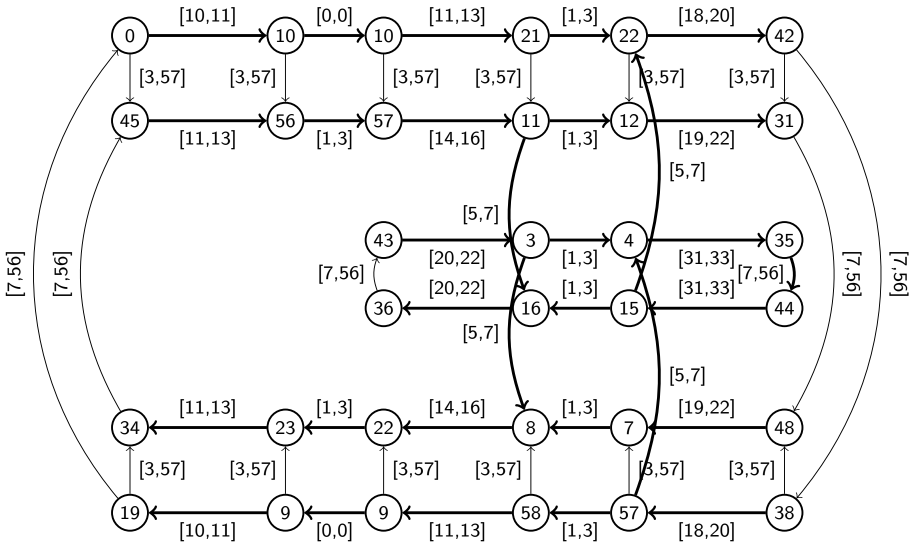


Figure 2.4: Solution for the example instance

$q_{\mathcal{C}}$ can be computed according to (2.5). As an example, consider the cycle formed by adding the arc between the two departure events of line l_1 and l_2 at station S_1, oriented in the direction of this added arc. Then the bounds can be computed as

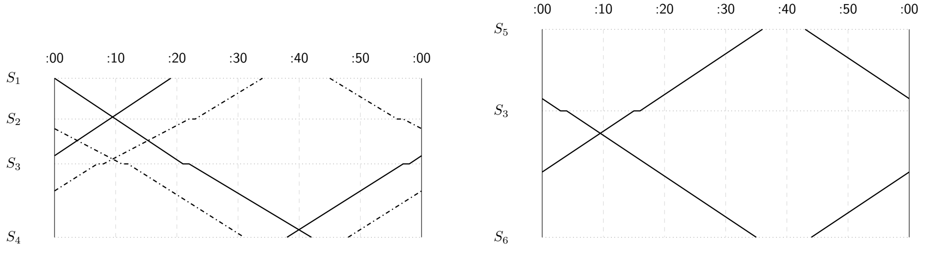
$$a_{\mathcal{C}} = \left\lceil \frac{1}{60}\left(3+11+1+14+5-3+5-3-13-0-11\right)\right\rceil \quad = \left\lceil \frac{9}{60} \right\rceil = 1 \quad (2.8a)$$

$$b_{\mathcal{C}} = \left\lfloor \frac{1}{60}\left(57+13+3+16+7-1+7-1-11-0-10\right)\right\rfloor \quad = \left\lfloor \frac{80}{60} \right\rfloor = 1 \quad (2.8b)$$

This shows that in this cycle, despite the fact that the sum of process times can vary between 9 and 80, there is only one possibility for the number of multiples of T. This can help in a branch and bound procedure to solve a PESP instance.

Time-Space Diagram A graph that is often used to picture a timetable is a *Time-Space Diagram.* This is a graph with on one axis space/distance, and on the other axis time. Throughout this thesis, we plot the distance on the vertical axis. In a time-space diagram, a line corresponds to the timetable of one train. The Time-Space Diagrams corresponding to the solution shown in Figure 2.4 are pictured in Figure 2.5.

Figure 2.5a displays the Time-Space diagram for the route between stations S_1 and S_4. The stations are mentioned on the left of the figure. Time is shown on the



(a) $S_1 \longleftrightarrow S_4$ (b) $S_5 \longleftrightarrow S_6$

Figure 2.5: Time-Space diagrams displaying the solution in Figure 2.4

horizontal axis, one cycle time is displayed between :00 and :60. The solid line displays the timetable for line l_1. The corresponding train in the forward direction departs from S_1 at :00, passes S_2 without stopping and dwells for one minute at S_3, which is indicated by the small horizontal part of the line at S_3. Finally, it continues to S_4 to arrive there at :42. The train in the backward direction leaves S_4 at :38 and arrives at S_3 at :57. It leaves there at :58 and travels to S_2 and S_1. Here it crosses the cycle time and it goes back to the beginning of the cycle again. This is indicated by the solid line corresponding to this train stopping at the right of the figure, and re-entering the figure again on the left. The dashed line shows the timetable for line l_2. Figure 2.5b shows the Time-Space diagram for the route between stations S_5 and S_6.

The provided example clearly shows the use of several types of activities that can be considered in PESP instances. Furthermore, it shows how the instance can be represented by a timetabling graph, in which we can also relatively easy find an integer cycle basis. Finally, we showed how a simple objective function can be included to find solutions with specific properties.

Appendix

2.A Several Types of PESP Activities

When two trains share the same part of infrastructure, we introduce constraints to prevent them from using it simultaneously. Trains can be driving in the same direction or in opposite directions on that infrastructure. As mentioned before, we use a macroscopic modelling of the infrastructure network. Safety activities are used to separate pairs of trains in time. In this appendix, we describe the modelling of several safety activities. An assumption that is often made in timetabling is that trip times are fixed. This reduces the problem size and is especially usefull when modelling safety activities. We show how to deal with known and fixed trip times and how the bounds of many of the safety activities can depend on some trip time of a train on a part of the railway infrastructure.

For notational convenience, let h_a and t_a be the 'head' and 'tail' of an activity in the PESP-graph respectively, i.e. for $a = (i, j) \in A$ we have $h_a = j$ and $t_a = i$. When trip times are assumed to be fixed, one could think of the nodes in a PESP instance as a combination of a departure event, a trip time and an arrival event, aggregated into one contracted node. In this context, trip time activities link consecutive departures of a train line to each other. The bounds in these activities consist of the trip time from a departure to the next arrival, and the possible dwell time at the arrival station. If this dwell time should be between $\underline{d}$ and $\overline{d}$ and the trip time is denoted by r, the trip/dwell time activities are of the form

$$r + \underline{d} \leq \pi_j - \pi_i + Tp_{ij} \leq r + \overline{d}, \tag{2.9}$$

with i and j the two departure events. Other common activities that arise in timetabling are dwell time, connection and synchronisation activities. An overview on how to derive such PESP activities is shown in Peeters (2003). In practice, the majority of the activities however are on safety. In the remainder of this section, the safety activities are explained, as well as the way they depend on trip times.

2.A.1 Trains Tunning in the Same Direction

Multiple trains using the same piece of infrastructure are separated in time by minimum headway times in order to guarantee a safe operation. In reality, such headway restrictions are imposed at stations as well as in between stations at any point where a conflict could occur, e.g., at points where train paths cross or merge. For the sake of simplicity, we only refer to stations in the description here.

Suppose two trains share the same track between stations s and s' and assume that the trains must be separated in time upon arrival and departure by at least h minutes. The departure and arrival times of train i at station s are denoted by $\pi^s_{d_i}$ and $\pi^s_{a_i}$ respectively. The travel time between the stations for train i is denoted by r_i. If one train departs from station s, the other train cannot depart in the h minutes before or after this departure. This leads to

$$\pi^s_{d_2} - \pi^s_{d_1} \notin (-h, h). \tag{2.10}$$

Since the timetable is cyclic, we can use this to model the above as a PESP activity:

$$\pi^s_{d_2} - \pi^s_{d_1} \in [h, T-h]_T, \tag{2.11}$$

or equivalently by introducing the integer variable p:

$$h \leq \pi^s_{d_2} - \pi^s_{d_1} + pT \leq T - h. \tag{2.12}$$

This activity is required for each pair of trains departing from a station s. For safety upon arrival at the next station s', a similar activity holds: replace d_i by a_i ($i \in \{1, 2\}$) and s by s'.

In order to state this activity solely in departure events, note that $\pi^s_{d_i} + r_i = \pi^{s'}_{a_i}$ ($i \in \{1, 2\}$), i.e., arrival time equals the departure time plus trip time. Substituting this into the headway activity upon arrival leads to

$$h \leq \pi^s_{d_2} + r_2 - \pi^s_{d_1} - r_1 + Tp \leq T - h \tag{2.13}$$

Rewriting this by moving the trip times to the activity bounds gives

$$h + r_1 - r_2 \leq \pi^s_{d_1} - \pi^s_{d_2} + Tp \leq T - h + r_1 - r_2 \tag{2.14}$$

as the safety activity ensuring a correct headway time upon arrival. Clearly, these bounds depend on the trip times of the trains towards station s'.

2.A.2 Single Track Headways

On some parts of the rail network, tracks are used in both directions. A train in one direction can only enter this track once the train in the opposite direction has cleared the track. There are several constraints that guarantee safety here. These are also used if the train paths of one incoming and one outgoing train cross around a station, which occurs frequently at larger stations.

Suppose stations s and s' are given with a single track in between. Trains can only pass each other at the stations. Train i ($i \in \{1, 2\}$) drives between s to s' in r_i minutes. In Figure 2.6 a sketch is given in a time space diagram, with time on the horizontal axis and space on the vertical axis. Train 1 is shown twice, the second denotes means the same train in the next cycle period, as is indicated by the departure times next to the nodes. Trains must be separated in time. How large this headway time h is, can be different if it concerns the time between an incoming train and the next outgoing train (an *in-out* relation, indicated by 'io'), or the other way round (an *out-in* relation, indicated by 'oi').

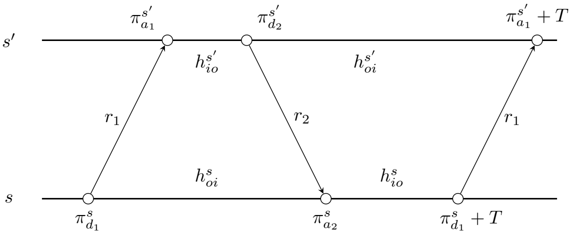


Figure 2.6: Single track time-space diagram

The activity to avoid a collision on such a single track is given by

$$h^{s'}_{io} + r_1 \leq \pi^{s'}_{d_2} - \pi^{s}_{d_1} + Tp \leq T - h^{s}_{io} - r_2. \tag{2.15}$$

The lower bound comes from the fact that $\pi^{s'}_{d_2} - \pi^{s'}_{a_1} \geq h^{s'}_{io}$, which implies

$$h^{s'}_{io} + r_1 \leq \pi^{s'}_{d_2} - \pi^{s}_{d_1}. \tag{2.16}$$

The upper bound comes from the fact that $\pi^{s}_{d_1} - \pi^{s}_{a_2} \geq h^{s}_{io}$, which implies

$$\pi^{s}_{d_1} - \pi^{s'}_{d_2} \geq h^{s}_{io} + r_2 \quad \Longrightarrow \quad \pi^{s'}_{d_2} - \pi^{s}_{d_1} \leq -h^{s}_{io} - r_2. \tag{2.17}$$

Combining this with (2.16) and using cyclicity, leads to (2.15).

Clearly, trip times are involved in the bounds. Headway time constraints upon arrival at stations can be stated without trip times being involved, by not contracting trip- and dwell times into one new activity, and hence keeping a node for the arrival times. For single-track constraints, this is not possible as is seen in the constraints stated above, safety can only be guaranteed using the actual trip times of the trains involved.

2.A.3 Crossing Train Paths

In (2.15) it is assumed that both trains share the whole single track part. It is possible that trains only use the same infrastructure close to a station and then go different ways. This basically means that no train can leave around the time an incoming train enters the station, when their routes cross. This restriction does not have to be as strict as the single-track restriction that prevents the trains meeting on the whole single track part.

The restriction dealing with the time between an incoming train (train 1) and the next outgoing train (train 2) at station s' is (see also Figure 2.6)

$$h_{io}^{s'} \leq \pi_{d_2}^{s'} - \pi_{a_1}^{s'} + pT \leq T - h_{oi}^{s'}, \tag{2.18}$$

which, using the fixed trip times, is equal to

$$h_{io}^{s'} + r_1 \leq \pi_{d_2}^{s'} - \pi_{d_1}^{s} + pT \leq T - h_{oi}^{s'} + r_1. \tag{2.19}$$

(2.19) considers an *in-out* relation. The *out-in* relation leads to the restriction

$$h_{oi}^{s'} \leq \pi_{a_1}^{s'} - \pi_{d_2}^{s'} + pT \leq T - h_{io}^{s'}, \tag{2.20}$$

which is the same as

$$h_{oi}^{s'} - r_1 \leq \pi_{d_1}^{s} - \pi_{d_2}^{s'} + pT \leq T - h_{io}^{s'} - r_1. \tag{2.21}$$

Chapter 3

Timetabling for Strategic Passenger Railway Planning*

3.1 Introduction

The public transportation planning process is traditionally subdivided into a number of steps which are assigned to either the strategic, the tactical, or the operational planning phase. According to Huisman et al. (2005), the strategic phase encompasses a time horizon of two to ten years before implementation and includes infrastructure decisions and line planning. The timetabling problem is often allocated to the tactical phase (approximately one year before implementation).

This chapter, however, focuses on *strategic timetabling*, i.e., the generating of a (preliminary) timetable already in the strategic planning phase. Strategic timetabling can be used to make strategic decisions with respect to timetables, like *"What should the headway times be between consecutive trains at a station?"* and *"Where should good transfer connections between trains be made?"*. Due to the location of strategic timetabling early in the planning horizon, it can also be used to evaluate line plans and to point to bottlenecks in the infrastructure, and thus to promising infrastructure investments as is supported by Odijk et al. (2006).

*This chapter, up to minor modifications, is a direct copy of G.J. Polinder, M. Schmidt, and D. Huisman (2020): *Timetabling for strategic passenger railway planning.*

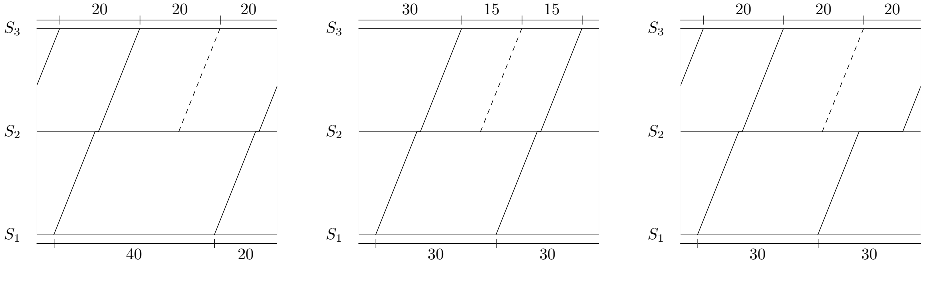
The value of strategic timetabling has been recognized in the practice of transportation planning. Following the example of Switzerland, the initiative *Deutschland-Takt* aims at establishing a so-called ‘integraler Taktfahrplan’ in Germany from the year 2030 on (Deutschland-TAKT, 2019). The transportation system should be redeveloped in such a way that connections between cities are served every 30 or 60 minutes, and that better transfer connections are provided. Reversing the current planning practice, the creation of a so-called ‘target timetable’ should precede and guide infrastructure investment decisions (e.g. in additional tracks between stations, or additional platforms at stations). In the Netherlands, a similar approach is used to evaluate infrastructure investments (Beter & Meer, 2014).

However, to the extent of our knowledge, models from academic research on timetabling as well as software tools for timetabling decision support are aimed at timetabling in the tactical (and operational) planning phase. Therefore, they focus on operational feasibility on a given infrastructure, and are rather restrictive in modelling of quality requirements. While these features are suitable for the more restrictive setting of tactical and operational planning (where changes in infrastructure and major changes in passenger behaviour are not desirable or possible), they are not appropriate to find new and innovative timetabling solutions as is desirable in the strategic planning phase.

In this chapter we aim to close this gap by presenting an optimisation approach to strategic timetabling. As common in railway timetabling, we aim at finding a periodic timetable, i.e., we require that the timetable follows a repeating pattern and hence the timetable of a base period is repeated throughout the day. Our objective is to find a periodic timetable that minimizes average perceived travel time for a given line plan. Different from most other timetabling models, we include adaption time in the perceived travel time. Assuming that the desired departure time of a passenger is known, the adaption time describes the time difference between this desired departure time and his actual departure time. Whereas for example Yan et al. (2019) penalizes regularity deviations to obtain a passenger oriented timetable, we explicitly allow for irregularity. Using the adaption time allows us to omit regularity constraints between runs of the same line (or runs of different lines that run in parallel for part of their route), which are otherwise often used to ensure low adaption times in an indirect way, simply by enforcing regular departures. Note that in case of dense networks and high frequencies, where OD-pairs are served by more than one line, the question which trains should be synchronized with each other becomes far from trivial to answer. In such situations, imposing regularity constraints may lead to sub-optimal solutions or even infeasibility of the timetabling problem (Chapter 5).

This is illustrated in the following example. Consider three stations S_1, S_2, S_3, and travel demand between all pairs of stations. Assume that the line plan prescribes a line from S_1 via S_2 to S_3 with a frequency of two trains per hour, and a line from S_2 to S_3 with a frequency of one train per hour. If we synchronize all trains between S_2 and S_3, the headway time between the trains on this part of the route will be 20 minutes, but from S_1 to S_2 the headway times are 20 minutes and 40 minutes. This is depicted in the time-space diagram in Figure 3.1a, where time and distance are shown on the horizontal and vertical axis, respectively. On the other hand, if we synchronize between S_1 and S_2, we have one headway of 30 minutes and two shorter headway times between S_2 and S_3 (Figure 3.1b). Perfect synchronization on both parts of the network is possible, but only if one of the trains from S_1 to S_2 waits an additional 10 minutes at S_2 (Figure 3.1c). Which of these solutions is best with respect to average perceived travel times depends on the size of the travel demand between the stations and the perceived value of adaption time compared to in-train time, as will be further illustrated in Section 3.3.4.

Our timetabling model allows us to find the best trade-off in case of different synchronization options for more complicated networks than the one sketched by explicitly including the adaption time into the perceived travel time, instead of deciding on where to impose regularity constraints heuristically before the optimisation.



(a) $S_2 \rightarrow S_3$ synchronised (b) $S_1 \rightarrow S_2$ synchronised (c) All synchronised

Figure 3.1: Time space diagrams for different synchronisation options

We define the *Strategic Passenger-Oriented Timetabling (SPOT)* problem as follows: Given a railway network consisting of stations and links connecting them, a line plan specifying lines routes and frequencies on the network, and an estimate of the hourly expected demand per origin-destination pair: find a timetable that minimizes average perceived passenger travel time under the assumption that passengers will choose the route with least perceived travel time. Hereby, perceived travel time is measured from the desired departure time on, that is, it includes adaption time.

Since we consider timetabling in the strategic planning phase, we cannot expect to have accurate demand information. In particular, the exact timing of travel requests is unknown. Therefore, we think that in this time frame it is appropriate to model passengers' desired departure times as evenly spread over the period and explicitly use this assumption in our mathematical program for the SPOT problem. Both the assumption that passengers indeed arrive randomly at the station and the assumption that they adapt to the communicated timetable to a large extent can be modelled by a parameter in our objective function which relates the perceived duration of waiting at the origin to the in-train time.

We model the SPOT problem as a quadratic mixed integer program that extends the traditional Periodic Event Scheduling Problem (PESP) model for periodic timetabling (Serafini and Ukovich, 1989). This model is explained in Chapter 2. We linearise the SPOT model and develop a heuristic to find a starting solution. We test our approach in two case studies based on the Dutch railway network.

Note that infrastructure constraints can be included in PESP (and thus also in our SPOT model) as headway constraints in a natural way (Liebchen and Möhring, 2007). However, due to the strategic perspective we take, we do not include them in our approach for two reasons: to be able to identify promising infrastructure investments, and to keep the model tractable. In later planning phases (tactical and operational planning), when the timetabling focus shifts towards macroscopic and later microscopic feasibility, such constraints can (and should) be added. Furthermore, in our model we omit upper bounds on transfer times and regularity constraints, since this would artificially restrict the solution space, and long transfers and irregular departure patterns are penalized in the objective function.

Our contribution in this chapter is fourfold. First of all, we formulate the *Strategic Passenger-Oriented Timetabling (SPOT)* problem for timetabling in the strategic planning phase. Secondly, we model this problem as a quadratic integer program that integrates timetabling with passenger routing (on perceived-travel-time-minimal routes) and linearise this formulation. Thirdly, we propose a heuristic to construct a starting solution, in order to find good solutions even for complex large instances. Fourthly, we test our model on two case studies based on the Dutch railway network, illustrating the trade-offs between the duration of dwell times and regularity of train service.

The remainder of this chapter is organized as follows. In Section 3.2, we describe literature that is related to our study. We discuss the SPOT problem in detail in Section 3.3. In Section 3.4 we propose a quadratic integer programming model for SPOT and linearize it. In Section 3.5 we describe how we solve it. In Section 3.6, we evaluate our solution approach and perform a case study on two practical instances from Netherlands Railways in Section 3.7. Finally, we conclude this study in Section 3.8.

3.2 Related Work

In this section, we give an overview on related research. A general overview on research related to periodic timetabling has been provided in Chapter 2. In Section 3.2.1 we describe attempts to timetabling in the strategic planning phase. Section 3.2.2 describes how passenger routing can be combined with periodic timetabling and how this is done in existing literature.

3.2.1 Strategic Timetabling

Decisions in the public transportation planning process are traditionally assigned to either the strategic, the tactical or the operational planning phase, according to their location in the planning horizon. E.g., according to Huisman et al. (2005), the strategic planning phase at the Dutch railway operator Netherlands Railways (NS) spans a period from 20 to 'few' years before the date of implementation and encompasses rolling stock management, crew management, and line planning (see also Chapter 1). Timetabling and rolling stock scheduling are allocated to the tactical planning phase, and operational and short-term planning include detailing timetable and rolling stock scheduling, as well as crew scheduling. However, finding a feasible and good timetable for a given line plan becomes increasingly difficult with an increasing utilization of the railway network. We therefore propose to include (a first attempt on) timetabling already in the strategic planning phase, to be able to use it as a tool to evaluate, and possibly reject proposed line plans. Several attempts have been made in the literature to *integrate* line planning and timetabling, see, e.g, in Burggraeve et al. (2017), Michaelis and Schöbel (2009), Schöbel (2017), Yan and Goverde (2019). We follow a different approach, regarding timetabling as a problem on its own (as opposed to integrating it with line planning), but taking into account typical characteristics of the strategic planning phase, like the lack of detailed passenger information and the possibility to extend infrastructure. While in the practice of public transportation planning, strategic attempts on timetabling are not uncommon (see, e.g., Deutschland-TAKT, 2019), research on railway planning normally allocates timetabling to the tactical phase. An omission of infrastructure constraints is not uncommon in railway timetabling, see e.g., Borndörfer et al. (2017), Pätzold et al. (2017), Robenek et al. (2017), Robenek et al. (2016), Schmidt and Schöbel (2015b), but other than omitting infrastructure constraints, the approaches cited here do not seem particularly tailored to strategic planning.

3.2.2 Timetabling and Passenger Routing

Timetables are evaluated according to different criteria. Following Cacchiani and Toth (2012), the most prominent are that the timetable should be (1) ‘efficient’ and (2) ‘robust’. Overviews on approaches to deal with robustness can be found in Cacchiani and Toth (2012), Lusby et al. (2018) and Chapter 6 of this thesis. Efficiency can be aimed both at costs and travel time aspects (or a trade-off of both). In the following we give an overview on how the literature addresses one aspect of ‘efficiency’, namely minimizing the passenger travel times, since this is also the objective we use in our model.

Early approaches to find timetables minimize passenger waiting times by assigning weights, modeling passenger numbers, to activities in the objective function (see the aforementioned references). These approaches, however, neglect that passengers choose their routes based on the timetable, which makes it difficult to assign a-priori weights to activities.

Thus, better results can be obtained when timetable and passenger routing are determined simultaneously. Several approaches have been published regarding an integrated approach, both in periodic and aperiodic settings. Schmidt and Schöbel (2015a) integrate passenger routing in aperiodic timetabling. Passenger demand is a priori assigned to a departure event, and passengers are routed from that point onwards. For periodic timetabling, a similar approach is taken by Borndörfer et al. (2017). In this approach, train capacities are used to determine frequencies of train lines. Furthermore, many performance criteria are introduced regarding several routing methods. A more recent approach where a viewpoint on applicability in practice is taken is by Schiewe and Schöbel (2018). The authors study the effect of including only a subset of the OD-pairs, in order to reduce the computational effort and to obtain good timetables in a short time. Hartleb and Schmidt (2019) also integrate timetabling with passenger routing, but here passengers are not routed along shortest paths, but a logit distribution is used. An alternative to integrating timetabling and passenger routing in one integer programming model, is to iterate timetabling and passenger routing. Kinder (2008), Lübbe (2009), Siebert (2011), Siebert and Goerigk (2013) determine passenger flows by routing passengers through the network on, for example, shortest paths with respect to the travel time. After this, the timetable is optimised with respect to these flows and passengers are rerouted, until a stopping criterium is reached.

The division of the public transportation planning into several sub problems (like line planning, timetabling, vehicle scheduling, etc.) is likely to lead to globally suboptimal solutions. Therefore, there are attempts to also integrate line planning and vehicle scheduling into timetabling with passenger routing (see, e.g., Lübbecke et al., 2018b; Schöbel, 2017). However, for real-life instances this leads to models that are hard to solve, and in these cases each sub problem is solved separately.

None of the aforementioned approaches considers adaption time, although a few approaches exist which explicitly consider this. Some of them focus on a single corridor and schedule the trains in such a way that the average adaption time is minimized (Barrena et al., 2014a, 2014b). In these papers, a mathematical programming model and a large-neighbourhood search algorithm to find good solutions is provided. A similar situation is considered in Zhu et al. (2017), where the authors consider a bi-level model. In the upper layer, a timetable is found based on passenger demand. In a lower layer, passenger arrival times are updated such that passengers arrive shortly before their train departs, in order to minimize adaption time. Another approach to solve timetabling with integrated passenger routing including adaption times is given in Gattermann et al. (2016), where the problem is modelled as a Satisfyability problem and solved with a dedicated solver. Here passengers are assigned to a time slice and a penalty is given if a passenger cannot depart within that time slice and has to adapt to a different slice. Borndörfer et al. (2017) and Hartleb et al. (2019) discuss and compare alternative evaluation functions for passenger-oriented timetabling. Wang et al. (2015) propose an approach to reduce the operational costs of trains which models demand as time-dependent and includes route choice. Finally, Yin et al. (2017) include passenger demand and adaption time minimization into an approach to optimise energy efficiency.

In this chapter, we integrate timetabling and passenger routing in a mathematical model and include the adaption time of passengers. By discarding the current infrastructural situation, solutions to our model can be used to support decision making in the strategic planning phase of railway planning.

3.3 Problem Definition

3.3.1 Periodic Timetabling

Following Chapter 2, a timetable is an assignment of points in time π_i to events i. In periodic timetabling, these events reoccur each base period with length T. The *periodic event-scheduling (PESP)* approach (Serafini and Ukovich, 1989) visualizes the set of periodic events and constraints on these events in a so-called *event-activity network* $G = (V, A)$. Each activity $(i, j) \in A$ represents a constraint and is formulated as a relation between events $i, j \in V$, stating that the time difference between these two events should be in a given (periodic) time interval, bounded by a lower bound ℓ_{ij} and an upper bound u_{ij}, where the word 'periodic' refers to the fact that event times are considered modulo T. This can be formalized by requiring that a timetable is a function $\pi : V \to \{0, \ldots, T-1\}$ on the departure and arrival events V such that

$$y_{ij}(\pi) = \pi_j - \pi_i + Tp_{ij} \in [\ell_{ij}, u_{ij}], \tag{3.1}$$

for all activities $(i, j) \in A$. p_{ij} is an integer variable, denoting the shift from one cycle period to the next, see also Chapter 2. We call $y_{ij}(\pi)$ the duration of activity (i, j) under timetable π.

As mentioned in Chapter 2, these activities can cover a variety of constraints that a timetable has to satisfy, e.g., drive and dwell activities. It is common in periodic timetabling to impose more constraints in the form of activities: firstly, a set of constraints that impose *infrastructure* constraints (trains that use a common part of the infrastructure must keep a safety distance), secondly a set of constraints that is aimed to increase the *quality* of a timetable from the passengers' perspective by enforcing, e.g., synchronization of trains of the same line (or of lines with a similar route), upper bounds on dwell times and coordination of transfers. Overviews on how to model a variety of timetabling constraints in a PESP framework can be found in Liebchen and Möhring (2007), Odijk (1996), Peeters (2003).

In our model, the quality of the timetable is controlled by the objective function 'total perceived travel time'. Therefore, the set of quality constraints is unnecessary in our model. For routing purposes, however, we do include transfer activities in our event-activity network, that connect the arrival event of a train at a station with the departure events of other trains at the same station. The bounds on these activities are set to $[\ell_{ij}, T + \ell_{ij} - 1]$, with ℓ_{ij} representing the minimum time needed for a transfer between arrival i and departure j. In this way, transfer activities do not impose any constraint on the feasibility of a timetable, but can be used in a convenient way to describe passenger routes and to compute their perceived travel time. This is detailed in Section 3.3.2.

Furthermore, in many cases, in the strategic phase of the timetabling process, infrastructure does not impose hard constraints on the timetable yet. Therefore, we propose to disregard (most of) the infrastructure constraints for timetabling in the strategic phase as well.

Note that without infrastructure constraints and quality constraints, the event-activity network consists of train paths, which are not connected to each other. Therefore, finding a *feasible* timetable on such an event-activity network is trivial - just choose an arbitrary time for each first event of a train and propagate it along the train path. In other words, the difficulty of solving PESP as a feasibility problem (see Section 3.2) can be attributed to the infrastructure and quality constraints.

It depends on the planning horizon and the budget for investments available, to which extent infrastructure availability imposes hard constraints on the timetable, or is subject to extension where needed. In this chapter, we neglect infrastructure constraints, due to the early planning phase in which we consider the timetabling problem. Adding all or some infrastructure constraints would pose no problem from a modelling perspective, but would make the problem computationally harder to solve.

In the same way, quality constraints may be added to our model if they are considered indispensable. We do believe, however, that in most cases the addition of quality constraints does unnecessarily constrain the set of feasible solutions, and that our objective function will find timetables of better quality if no quality constraints are imposed.

Note that, as common in the railway planning literature, we do not yet consider the rolling stock allocation to trips during timetabling, but postpone these questions to the *Rolling Stock Scheduling* problem which is solved in the tactical planning phase (see Chapter 1) and outside the scope of this chapter. In line with that, we do not consider constraints on the maximum number of passengers in a train, or operational costs, as these are largely determined by the rolling stock schedule.

3.3.2 Passenger Demand, Route Choice, and Perceived Travel Time

We consider passenger demand aggregated over all time periods, by assuming that we have an OD-matrix $\mathcal{OD}$, providing for each OD-pair $k \in \mathcal{OD}$ an estimate of the average number of passengers d_k that want to travel from the origin to the corresponding destination per time period. While demand may differ between different time periods, we assume that within the same time period, demand is uniformly distributed over time. For each OD-pair $k \in \mathcal{OD}$, we precompute the set of routes among which the passengers may choose. A route r is a directed path through the event-activity network. It consists of a sequence of trip, dwell, and transfer activities, so r is an (ordered) subset of A. The set of routes for OD-pair $k \in \mathcal{OD}$ is denoted by $\mathcal{R}^k$. The set of all routes is

$$\mathcal{R} = \bigcup_{k \in \mathcal{OD}} \mathcal{R}^k. \tag{3.2}$$

For a fixed timetable π, the activity duration y_{ij} of drive, dwell, and transfer activities (i, j) is given by (3.1) in dependence of the chosen timetable. The *pure* travel duration of a route r is simply the sum of all activity durations on the route. To obtain the *perceived* travel duration $Y_r(\pi)$ of route r for timetable π, we penalize each transfer by adding a transfer penalty γ_t, that is

$$Y_r(\pi) := \sum_{(i,j) \in r} y_{ij} + \gamma_t \cdot 1_t(i, j). \tag{3.3}$$

The function $1_t(i, j)$ is an indicator function, denoting whether activity $(i, j) \in A$ is a transfer activity or not.

However, the route choice of a passenger does not only depend on the perceived route duration $Y_r(\pi)$, but also on the departure time of the route r. A passenger will prefer a route with slightly longer perceived duration, if its departure time is closer to his desired departure time. To formalize this idea, we introduce the notion of the perceived travel time of the passenger, which includes the *adaption time*. The *adaption time* $\mathrm{at}_r^t(\pi)$ for a passenger with desired moment of departure t on route r under timetable π is defined as the time difference between t and the scheduled time for the first event $\sigma(r)$ on route r $\pi_{\sigma(r)}$:

$$\mathrm{at}_r^t(\pi) := \pi_{\sigma(r)} - t \mod T. \tag{3.4}$$

Consequently, the perceived travel time of a passenger with desired departure time t on route r under timetable π is

$$\hat{Y}_r^t(\pi) := \gamma_w \cdot \mathrm{at}_r^t(\pi) + Y_r(\pi), \tag{3.5}$$

where γ_w is a weighting factor modelling the perception of adaption time in relation to perceived duration of the route.

Note, however, that our demand data does not contain information on the individual desired departure moments of passengers, but only aggregated information in the form of average hourly demand for an OD-pair.

We therefore work with expected passenger numbers per minute. For an OD-pair with an average demand of d_k passengers per time period T, we assume that for each possible departure moment t we have on average $\frac{d_k}{T}$ passengers who would like to depart at time t.

Under the assumption that passenger demand is distributed uniformly over the period, and that each passenger chooses a route which minimizes his perceived travel time, we can thus compute the perceived travel time of a timetable π as

$$\mathrm{tt}(\pi) := \sum_{k \in \mathcal{OD}} \sum_{t=1}^{T} \frac{d_k}{T} \cdot \min_{r \in \mathcal{R}^k} \hat{Y}_r^t(\pi). \tag{3.6}$$

Note that demand will vary among periods. Since d_k is the average demand for OD-pair k per period, $\mathrm{tt}(\pi)$ is the total travel time under timetable π, averaged over all considered periods.

3.3.3 Time-slice Based Reformulation of Route Choice and Objective Function

Note that the formulation of the objective (3.6) suggests that to evaluate a timetable, we have to determine $T \times |\mathcal{OD}|$ routes, one for each OD-pair and each moment in time. We can reduce the number of routes to be determined for the evaluation of a timetable considerably by dividing the time period into time slices for each OD-pair. To this end, we denote the set of *relevant* departure events for OD-pair k, that is, all first departure events $\sigma(r)$ of the routes in $r \in \mathcal{R}^k$ by

$$V^k = \bigcup_{r \in \mathcal{R}^k} \{\sigma(r)\}. \tag{3.7}$$

For an OD-pair k and a given timetable π, we divide the period into time slices, according to the departure times π_v of the relevant departure events $v \in V^k$. Let S_v^k denote the time slice between the relevant departure event v and the departure event $v'(\pi) \in V^k$ that immediately precedes v according to timetable π.

Note that while passengers with desired departure time t do not necessarily take the *next* departing route towards their destination, but the one that minimizes the perceived travel time $\hat{Y}_r^t(\pi)$ of which adaption time is only *one* component, for all passengers with departure time in the same time slice S_v^k, the same route will be optimal with respect to perceived travel time. Namely the route r with

$$Y_v^k(\pi) = \min_{r \in \mathcal{R}^k} \gamma_w \cdot (\pi_{\sigma(r)} - \pi_v \mod T) + Y_r(\pi), \tag{3.8}$$

i.e., the time difference from v to the departure event of that route (weighted with the adaption time factor γ_w) plus the perceived route duration. Therefore, for each OD-pair k, we aggregate demand in each time slice S_v^k.

Let $L_v^k(\pi)$ denote the length of time slice S_v^k in timetable π. Then the number of passengers arriving during time slice S_v^k in timetable π is $\frac{d_k}{T} \cdot L_v^k(\pi)$, and we can compute the average waiting time for a passenger arriving during time slice S_v^k until π_v is $W_v^k(\pi) = L_v^k(\pi)/2$.

Thus, the total contribution to the total perceived travel time of passengers arriving during time slice S_v^k is

$$\frac{d_k}{T} \cdot L_v^k(\pi) \cdot \left(\gamma_w \cdot W_v^k(\pi) + Y_v^k(\pi)\right) \tag{3.9}$$

and we can replace (3.6) by the rewritten objective function

$$\mathrm{tt}(\pi) = \sum_{k \in \mathcal{OD}} \frac{d_k}{T} \sum_{v \in V^k} L_v^k(\pi) \cdot \left(\gamma_w \cdot W_v^k(\pi) + Y_v^k(\pi)\right). \tag{3.10}$$

This allows us to evaluate a timetable π by computing an optimal route per relevant departure event.

3.3.4 Example

Let us reconsider the example given in Figure 3.1, to illustrate the evaluation of different timetables according to (3.6). We consider the three OD-pairs (S_1, S_2), (S_2, S_3), and (S_1, S_3) and the timetables (a), (b), and (c) from Figure 3.1. Note that in each of the three considered timetables, there are three (non-dominated) routes for OD-pair (S_2, S_3), and two (non-dominated) routes for OD-pair (S_1, S_2) and (S_1, S_3). For each OD-pair, the time spent on driving activities is independent of the route chosen, and identical for the three timetable. The difference in the evaluation of the timetables stems purely from the adaption time, and for OD-pair (S_1, S_3) additionally from the dwell time at station S_2, which increases the in-train time of OD-pair (S_1, S_3) on the second route by 10 minutes under timetable (c).

Let us consider OD-pair (S_1, S_2) in timetable (a). There are two relevant departure events for this OD-pair, σ_1 at time $\pi_{\sigma_1} = 0$ and σ_2 at time $\pi_{\sigma_2} = 40$. Since $T = 60$ in this example, $L^{12}_{\sigma_1} = 20$ and $L^{12}_{\sigma_2} = 40$. Furthermore, $W^{12}_{\sigma_1} = L^{12}_{\sigma_1}/2 = 10$ and $W^{12}_{\sigma_2} = L^{12}_{\sigma_2}/2 = 20$.

Therefore, the expected total adaption time for OD-pair (S_1, S_2) in timetable (a) is $\frac{d_{12}}{60} \cdot (20 \cdot 10 + 40 \cdot 20) = d_{12} \cdot 16.67$, or 16.67 on average per passenger of this OD-pair. In the same way, we compute the adaption time for the other OD-pairs under the three different timetables. The results are summarized in Table 3.1.

OD-pair	Timetable (a)	Timetable (b)	Timetable (c)
(S_1, S_2)	16.67	15	15
(S_2, S_3)	10	11	10
(S_1, S_3)	16.67	15	16.67(+10)

Table 3.1: Expected average adaption time per passenger for OD-pairs (S_i, S_j) in the timetables shown in Figure 3.1. For OD-pair (S_2, S_3), additional waiting time for timetable (c) is added in brackets

It is now easy to see that it depends on the number of passengers (and the weighting factor γ_w that trades off adaption time and in-train time), which of these timetables is optimal with respect to total perceived travel time. E.g., if $d_{13} = 100$, and $d_{12} = d_{23} = 0$, we have a total adaption time of 15000 and no additional waiting time for timetable (b), while for timetable (a) and (c), the adaption time would be higher.

Furthermore, we observe that for each OD-pair, adaption time is lowest if the relevant departure events are evenly spaced. This is a consequence of the fact that $\sum_{v \in V^k} L_v^k = T$ for all $k \in \mathcal{OD}$ (the time slice lengths add up to the period length) and of the corresponding term in the objective function (3.10) being quadratic in L_v^k, since $W_v^k = L_v^k/2$. A formal proof is given in Appendix 3.C.

3.3.5 Problem Statement

We can now summarize the preceding sections by restating the definition of the *Strategic Passenger-Oriented Timetabling (SPOT)* problem given in Section 3.1:
Given a railway network consisting of stations and links connecting them, and a line plan specifying lines routes and frequencies on the network, described by a periodic event-activity network as detailed in Section 3.3.1 and an OD-matrix $\mathcal{OD}$, providing for each $k \in \mathcal{OD}$ an estimate of the average number of passengers d_k that want to travel from the origin to the corresponding destination per time period, find a periodic timetable π that fulfills constraints (3.1) *and that minimizes the total perceived travel time $tt(\pi)$ as defined in* (3.10).

Summarizing, our timetabling problems differs from most other timetabling literature (compare Section 3.2) in the following two aspects:

On the one hand, our problem formulation explicitly aims at finding a timetable in the strategic planning phase. Firstly, other than most publications on timetabling targeted to the tactical planning phase, we purposefully omit *infrastructure* constraints in our experiments, to allow our model to find an ideal timetable for a given line plan, and in doing so to point at bottlenecks and necessary infrastructure improvements. Note however, that from a modelling perspective, the inclusion of infrastructure constraints into our model is straightforward. Secondly, on a planning horizon of several years, only a coarse estimate of *demand data* can be obtained. In particular, no reasonable estimate on desired departure or arrival times can be made so long in advance. Our model requires only an estimate of daily averages per OD-pair. It assumes that demand is uniformly distributed over the period, but demand may differ between different periods, as is to be expected over a whole day of service.

On the other hand, by the definition of perceived travel time as objective function, which includes *adaption time* in the evaluation of a timetable, we are able to omit many *artificial constraints* that other models need to introduce to ensure *quality* of the timetable. Our objective function balances the need for low transfer times, low in-train waiting times, and equally spaced departures from the origins taking into account passenger numbers on the corresponding connections, and their impact on the total perceived travel time. The omission of artificial, and often arbitrarily set quality constraints expands the space of feasible solutions tremendously, and thus avoids the problem of severely restricted or even infeasible timetabling instances, which occurs increasingly often in view of the growing utilization of railway infrastructure (compare Chapter 5).

3.4 Integer Programming Formulation for SPOT

In this section, we model the SPOT problem as a mathematical program and linearise it. An overview of sets, constants, and variables used in the (linearisation of the) mathematical program for SPOT is given in Table 3.10 in Appendix 3.B.

3.4.1 Precomputing Passenger Routes

We precompute possible routes for each OD-pair. For this, we use the method described in Warmerdam (2004). This method first determines all direct travel options. Next, this set of travel options is extended by all options with 1 transfer, then with 2 transfers, and so on, until a predefined maximum number of transfers is reached. The (expected) duration of each of these routes is computed as the sum of the minimum time needed for the trips, increased with a small supplement for stops that must be made, and multiplied by a percentage (which generally is 5%) to account for uncertainty in the trip durations and to incorporate some robustness against delays. If transfers are included in the route, additional time is added, based on the expected transfer times. If there are n_1 incoming connections, and n_2 outgoing connections, this expected transfer time equals $30/\max\{n_1, n_2\}$ minutes. After the possible routes have been generated, a check is done whether some routes are dominated by others. In order to check dominance, routes are compared based on their expected duration, geographical length and number of transfers. As an example, if route A is longer with respect to travel time, and has more transfers than route B, it is discarded. Similarly, if two routes are identical, except for the transfer station that is chosen, the transfer station is chosen which has the highest number of trains and other options are discarded.

Note that a different method to generate routes would be possible as well, as long as the paths can be used as input to the model.

3.4.2 Mathematical Programming Formulation

We can formalize the SPOT problem as described in Section 3.3 as follows:

$$
\begin{aligned}
\min \quad & \sum_{k\in\mathcal{OD}} \frac{d_k}{T} \sum_{v\in V^k} L_v^k \cdot \left(\gamma_w \cdot W_v^k + Y_v^k\right) && && \text{(3.11a)}\\
\text{s.t.} \quad & y_{ij} = \pi_j - \pi_i + T p_{ij} && \forall\,(i,j)\in A && \text{(3.11b)}\\
& \ell_{ij} \le y_{ij} \le u_{ij} && \forall\,(i,j)\in A && \text{(3.11c)}\\
& Y_r = \sum_{(i,j)\in r} \left(y_{ij} + \gamma_t \cdot 1_t(i,j)\right) && \forall\, r\in\mathcal{R} && \text{(3.11d)}\\
& \Delta_{v,v'} = \pi_{v'} - \pi_v + T\alpha_{v,v'} && \forall\, v,v'\in V^k && \text{(3.11e)}\\
& L_v^k = \min_{v'\in V^k\setminus\{v\}} \{\Delta_{v',v}\} && \forall\, k\in\mathcal{OD},\, v\in V^k && \text{(3.11f)}\\
& \alpha_{v,v'} + \alpha_{v',v} = 1 && \forall\, k\in\mathcal{OD},\, v\in V^k, && \\
& && \quad v'\in V^k\setminus\{v\} && \text{(3.11g)}\\
& W_v^k = \frac{1}{2} L_v^k && \forall\, k\in\mathcal{OD},\, v\in V^k && \text{(3.11h)}\\
& Y_v^k = \min_{v'\in V^k} \min_{r\in\mathcal{R}_{v'}^k} \{Y_r + \gamma_w \cdot \Delta_{v,v'}\} && \forall\, k\in\mathcal{OD},\, v\in V^k && \text{(3.11i)}\\
& L_v^k \in [0,T] && \forall\, k\in\mathcal{OD},\, v\in V^k && \text{(3.11j)}\\
& W_v^k \in [0,T/2] && \forall\, k\in\mathcal{OD},\, v\in V^k && \text{(3.11k)}\\
& Y_r \in \mathbb{R}_{\ge 0} && \forall\, r\in\mathcal{R} && \text{(3.11l)}\\
& Y_v^k \in \mathbb{R}_{\ge 0} && \forall\, k\in\mathcal{OD},\, v\in V^k && \text{(3.11m)}\\
& \pi_v \in \{0,\dots,T-1\} && \forall\, v\in V && \text{(3.11n)}\\
& p_{ij} \in \mathbb{Z}_{\ge 0} && \forall\,(i,j)\in A && \text{(3.11o)}\\
& \alpha_{v,v'} \in \{0,1\} && \forall\, k\in\mathcal{OD}, v\in V^k, && \\
& && \quad v'\in V^k\setminus\{v\}. && \text{(3.11p)}
\end{aligned}
$$

Constraints (3.11b) and (3.11c) are equivalent to (3.1). Constraints (3.11d) determine the perceived duration Y_r of each route r based on the activity durations associated with timetable π, as specified in (3.3). To determine the lengths of time slices in a periodic timetable, we introduce an additional binary variable $\alpha_{v',v}$ to replace the module operator. Then (3.11e) defines an auxiliary variable denoting the time difference between events v and v'. (3.11f) determines L_v^k, the length of time slice S_v^k, as the time distance to the previous periodic departure event in V^k. Hereby, constraints (3.11g) are required to impose an order between events, even when two departures happen at the same time. Constraints (3.11h) and (3.11i) determine average waiting time until the end of time slice S_v^k and perceived travel time on an optimal route from the end of time slice S_v^k, respectively, following Section 3.3.3. The objective aggregates the contributions of all time slices for all OD-pairs as given in (3.10). Constraints (3.11j)–(3.11p) state the domains of the variables.

3.4.3 Linearization of the SPOT Model

The model stated in (3.11) is non-linear. The objective function is quadratic as it contains the term $W_v^k \cdot L_v^k = \frac{1}{2}(L_v^k)^2$. Next to that, (3.11f) and (3.11i) contain one or two minimums. For our computations, we linearise the objective function and reformulate (3.11f) and (3.11i).

For the reformulation, we replace (3.11f) by the restrictions

$$A_v^k \leq Q_{v',v} \qquad \forall\, k \in \mathcal{OD},\ v' \in V^k \setminus \{v\} \tag{3.12a}$$

$$\sum_{v \in V^k} A_v^k = T, \qquad \forall\, k \in \mathcal{OD} \tag{3.12b}$$

where $Q_{v,v'}$ is defined as the periodic time difference between events v and v', i.e.,

$$Q_{v,v'} = \pi_{v'} - \pi_v + T\alpha_{v,v'}. \tag{3.13}$$

(3.12a) represents the minimum and (3.12b) ensures that all time differences add up to T. Note that the latter is already a valid restriction in (3.11).

In order to reformulate (3.11i), we introduce new binary variables $z_{v,v',r}^k$, denoting whether passengers for OD-pair $k \in \mathcal{OD}$, arriving before event v use route r, starting with event v'. We refer to Appendix 3.A for details.

For the linearisation of the objective function, we introduce new variables $x_{v,d}^k$, denoting whether $A_v^k \geq d$ or not. For the details, we refer again to Appendix 3.A.

The model stated in this section determines a timetable that minimizes the total perceived travel time of all passengers. No synchronisation constraints are added to the model, instead, the objective is designed such that the optimal spread of trains over time is determined.

3.5 Solution Approach

For real-life instances, even for networks of small size, the size and nature of the models easily exceed the capabilities of commercial solvers to find good solutions. Also due to the complex nature of the models, we do not expect to solve the models to optimality in a reasonable amount of time.

In this section we present the approach that we use in our experiments in order to find good solutions in a reasonable time. First of all, we set a time limit TL. Secondly, we can simplify our SPOT model in various ways. These simplifications are described in Section 3.5.1. Thirdly, we develop a heuristic to generate a feasible starting solution. This heuristic is described in Section 3.5.2.

3.5.1 Reduced Versions of SPOT

In this section we discuss some simplifications that can be made to the SPOT problem, which lead to a reduced model size and therefore possibly speed up the solution process. These can be used as heuristic approaches towards solving the full SPOT model.

The first two simplifications use a subset of the OD-pairs instead of the full set. In the first simplification, we consider only passengers with a direct travel option. Note that passengers from this set do not *need* to travel directly if a better connection is available for them.

The second simplification is motivated by the observation that in practice, the distribution of OD-pair sizes is very skewed: only a few OD-pairs are very large and cover a large part of the passengers. We expect that the timetable largely depends on the large OD-pairs, and that including the remaining OD-pairs would only lead to some minor changes to the timetable. To choose a subset of OD-pairs, we introduce a parameter $\lambda \in [0, 100]$. We then include the OD-pairs which are largest in passenger size such that in total at least $\lambda\%$ of the passengers is included. If λ is small enough, only a few OD-pairs are included, while a large part of passengers is taken into account. Note that when we combine the first two simplifications in our experiments, we include the OD-pairs which are largest in passenger size among the ones who have a direct travel option such that in total at least $\lambda\%$ of the passengers with a direct travel option are included.

The third simplification is to require in the model that passengers always take the first relevant train leaving from the station: in that case they are not allowed to wait for a later departing train. Note that also in this simplification, the order of trains departing from a station is not fixed. The intuition for this simplification is that for the majority of the passengers, waiting for a later train is in general not beneficial. Therefore, we expect this to be a simplification that does not sacrifice much in terms of quality of the solution, while still reducing the complexity of the model significantly. To implement this simplification, the first minimum in (3.11i) is taken over $v' \in \{v\}$ instead of over $v' \in V^k$. Equivalently, we take could $z^k_{v,v',r} = 0$ if $v \neq v'$ in (3.22).

As a fourth possible simplification, we choose to include only direct routes and do not allow for transfers. To achieve this, one could set the penalty γ_t to a very large value, thus allowing transfers, but making them very expensive. However, we choose to reduce the sets $\mathcal{R}^k_{v'}$, such that it includes only direct routes. This implies that OD-pairs for which no direct travel option exists cannot be included.

3.5.2 Heuristic to Find a Starting Solution

In this section we describe a heuristic approach to solve the integer program for the SPOT problem. When trying to solve SPOT to optimality, the heuristically generated solution can be used as a starting solution for the IP solver and in this way, help to speed up the solution procedure.

First of all, we take only a subset of the passengers into account when generating this heuristic solution. That is, in line with the second simplification mentioned in the previous section, we take as few OD-pairs as possible, such that $\lambda\%$ of the passengers are incorporated. Based on this subset of passengers, we compute a timetable by a heuristic procedure.

In order to state our approach, we first group the variables of the integer programming model for SPOT into the following sets:

$$\Pi = \{\pi_v \,:\, v \in V\} \tag{3.14a}$$

$$\mathcal{P} = \{p_{ij} \,:\, (i,j) \in A\} \tag{3.14b}$$

$$\mathcal{A} = \left\{\alpha_{v,v'} \,:\, k \in \mathcal{OD},\, v \in V^k,\, v' \in V^k \setminus \{v\}\right\} \tag{3.14c}$$

$$\mathcal{X} = \left\{x^k_{v,d} \,:\, k \in \mathcal{OD},\, v \in V^k,\, d = 1, 2, \ldots, T\right\} \tag{3.14d}$$

$$\mathcal{Z} = \left\{z^k_{v,v',r} \,:\, k \in \mathcal{OD},\, v \in V^k,\, v' \in V^k,\, r \in \mathcal{R}^k_{v'}\right\}. \tag{3.14e}$$

The first two sets contain variables that relate to the timetable itself. The variables in the set $\mathcal{A}$ are used to determine time differences between two events correctly. Finally, the sets $\mathcal{X}$ and $\mathcal{Z}$ are introduced in the linearisation of our model, and are related to the passenger routing.

To generate a good starting solution, we consecutively solve partial relaxations of the SPOT model, as outlined below. Since for all steps we require that the variables in the sets Π and $\mathcal{P}$ are integer, the timetabling constraints (3.11b), (3.11c), (3.11n), and (3.11o) are fulfilled. Thus, in each step, we find a feasible timetable.

Note that as soon as a timetable is fixed, it is possible to evaluate it according to objective function (3.11a). To this end, for each OD-pair we compute the lengths Y_v^k of perceived-travel-time-minimal routes from each relevant departure event $v \in V^k$ to the destination by solving a shortest path problem. Furthermore, we order the relevant departure events, and thus compute the time difference between π_v and the departure time of the previous departure, L_v^k, as well as the average waiting time for these passengers $W_v^k = L_v^k/2$. This allows us to compute the objective value of the timetable as specified in (3.11a).

We evaluate each timetable generated in the solution as described in (3.11a). The best solution according to this evaluation is stored as the incumbent and only replaced when a better solution is found.

The steps of the heuristic are detailed below. To give a quick overview, Table 3.2 displays for each step what type the variables are in that step, i.e., whether they are continuous ($\mathbb{R}$) or integer ($\mathbb{Z}$) or mixed.

Each step is solved with a time limit, that is based on the overall time limit TL. Furthermore, the solution for each step is used as a warm start for the next step. The heuristic is a variant on the 'Relax-and-Fix' heuristic, as explained in Belvaux et al. (1998), Wolsey (1998).

Step	Π	$\mathcal{P}$	$\mathcal{A}$	$\mathcal{X}$	$\mathcal{Z}$	Target gap (%)
1	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{R}$	$\mathbb{R}$	$\mathbb{R}$	0.0
2	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$ and $\mathbb{R}$	$\mathbb{R}$	$\mathbb{R}$	1.0
3	$\mathbb{Z}$	$\mathbb{Z}$	$\mathbb{Z}$ and $\mathbb{R}$	$\mathbb{R}$	$\mathbb{R}$	1.0
4	Fixed	Fixed	Fixed	$\mathbb{Z}$	$\mathbb{Z}$	0.0

Table 3.2: Overview of the integrality of variables in the heuristic

Step 1 In the first step, a solution is found that is feasible with respect to all timetabling constraints. Therefore, we relax all variables to continuous variables, except for the timetabling related variables, i.e., those in Π and $\mathcal{P}$. This model is solved to optimality, or until a time limit of $TL/10$ is reached.

Step 2 In this step, we improve the time differences between trains to get a better passenger routing, by changing a subset $\mathcal{A}'$ of the variables in $\mathcal{A}$ into integers, which we initialize as $\mathcal{A}' = \emptyset$.

In order to determine this set, we check for each pair of trains t_1 and t_2 whether their geographical routes overlap. If so, let $v_1, v_2 \in \Pi$ be the departure events of trains t_1 and t_2, respectively, at their first shared station. Then $\mathcal{A}' = \mathcal{A}' \cup \{\alpha_{v_1,v_2}, \alpha_{v_2,v_1}\}$.

We change all variables in $\mathcal{A}'$ into integers and set $\mathcal{A} = \mathcal{A} \setminus \mathcal{A}'$ and $\mathcal{A}' = \emptyset$. Then we re-optimise with a time limit of $TL/10$ or until an optimality gap of 1.0% is reached.

Step 3 In the previous step, a part of the α variables is changed into integers, but the majority is still continuous. In this step we iteratively change the remaining variables in $\mathcal{A}$ into integers, starting with the variables of which the value in the incumbent solution is not close to integer.

In each iteration of this step, we define the set of variables that are to be changed to integers as

$$\mathcal{A}' = \{\alpha \in \mathcal{A} \mid 0.05 \leq \text{val}(\alpha) \leq 0.95\}\,. \tag{3.15}$$

Here, $\text{val}(\alpha)$ denotes the value this variable α attains in the incumbent solution.

Again, we change the nature of all variables in $\mathcal{A}'$ to integers, we set $\mathcal{A} = \mathcal{A} \setminus \mathcal{A}'$, $\mathcal{A}' = \emptyset$ and re-optimise with a time limit of $TL/10$ or until an optimality gap of 1.0% is reached. This is continued until $|\mathcal{A}'| \leq 50$, in which case we set $\mathcal{A}' = \mathcal{A}$ in order to limit the number of iterations. Furthermore, this ensures that after these loops all α-variables have integer values.

Step 4 In this step, we fix all variables in Π, $\mathcal{P}$ and $\mathcal{A}$ to the value they attain in the incumbent solution (according to the evaluation with all OD-pairs). Next, we change all variables in $\mathcal{X}$ and $\mathcal{Z}$ to integers and reoptimise this model to optimality.

In order to better understand the heuristic, we highlight the rationale behind it. As headway constraints are not considered, the timetabling part is relatively easy in our model. Therefore we first find a timetable that is feasible with respect to the timetabling constraints, and include the passenger routing part only as a continuous relaxation. As this is a relatively easy task, we try to find an optimal solution here. This can however lead to a bad timetable from the perspective of passengers, as the time differences between events can be determined incorrectly, due to the continuous nature of the variables in $\mathcal{A}$. Therefore, in the next steps we try to improve this.

First, we determine where train lines meet for the first time. By making the corresponding variables integer, we aim at better spreading different train lines over time. The expectation here is that by changing only a few variables to integers, a good gain in terms of quality can be obtained, without making it too difficult. The places where train lines meet are those places where frequencies on the tracks can change and therefore the expectation is that these are crucial decisions. The next step turns the remaining variables into integers. By experiments we found that the values of the majority of the variables in $\mathcal{A}$ is very close to integer, and that the remaining variables are generally rather close to 0.5. Therefore we select these variables that have $0.05 \leq \text{val}(\alpha) \leq 0.95$, i.e., that are not too close to integer values. By selecting these variables, we expect to make a large step towards an overall feasible solution. Iteratively these variables are changed to integers. When there are not many variables left, we change the remaining variables into integers in order to limit the number of iterations needed. Finally, for the best found timetable, we determine the best routing and the heuristic finishes.

3.6 Computational Results

In this section, we apply our approach to two instances based on the network operated by Netherlands Railways. We use these instances to computationally evaluate the use of a heuristic starting solution and to investigate the effect of solving restricted versions of our model.

In all experiments we discretise time to minutes and use a period length of one hour, i.e., $T = 60$. For the perceived travel time, values for adaption time and transfer penalty must be set (γ_w and γ_t). According to De Keizer et al. (2015), the resistance for a transfer depends on many factors, but on average a penalty of 23 minutes (including 2 minutes of transfer time) is appropriate. Instead of 2 minutes, we use a minimum of 3 minutes for a transfer, so we use a value of $\gamma_t = 20$ in our models, to obtain the penalty of 23 minutes. As adaption time is a relatively new concept, there is not much research available to determine a good value for γ_w. We want to emphasize the regularity of trains to reduce adaption time, but not overstress it because adaption time already appears in the objective as a quadratic term. Therefore, we decided to use $\gamma_w = 3$.

In our implementation and when reporting objective values in this section, we only report the 'excess time'. That means, we subtract constant terms from the objective function to improve numerical stability. For trip time, that implies that we subtract the minimal duration of the shortest possible trip for that OD-pair. If some OD-pair needs at least ν transfers, we subtract ν times the transfer penalty, and only penalize additional transfers. Finally, to subtract a constant for the adaption time, we assume that departure events are spread evenly over time, which leads to the lowest possible adaption time (see Appendix 3.C for a proof), and subtract the corresponding value for the adaption time. This also explains why we do not report relative gaps. If we do not subtract the constant terms, all gaps would be relatively small. In our experiments, the lower bound is often close to zero and hence relative gaps are very large.

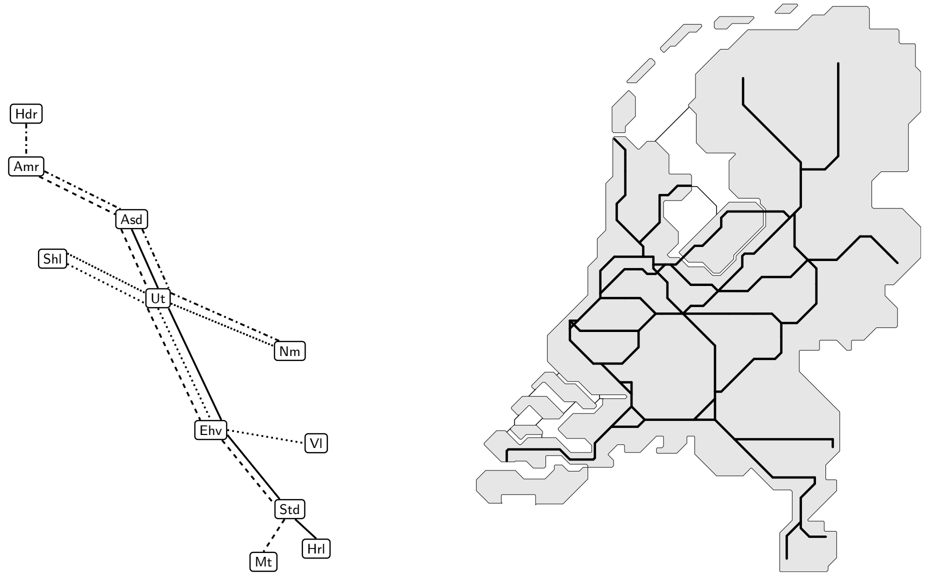
Our computations are carried out on a machine with an Intel Xeon Silver 4110 2.10Ghz processor and with 96 GB of RAM installed. The integer programs are solved by Cplex 12.9.0 under default settings, using up to 15 parallel threads (IBM, 2019).

3.6.1 Instances

The instances that we use in this chapter are real-life instances of Netherlands Railways (NS), the largest operator of passenger trains in the Netherlands. The first instance is the so-called 'A2-corridor', a network that contains 5 Intercity lines, that all share part of their route. The second instance is the 2019 Intercity network of Netherlands Railways (NS). In the remainder of this section, we describe the two instances in more detail.

A2-corridor

The first instance we consider in this chapter contains the so called 'A2-corridor', which is the part of the Dutch railway network between Eindhoven (Ehv) and Ams-



(a) A2-corridor (b) Intercity network

Figure 3.2: Overview of the used instances

terdam Centraal (Asd). The line plan for this instance is shown in Figure 3.2a, where each line is shown in a different style. The used abbreviations for the stations are mentioned mentioned in Table 3.3.

Abbreviation	Name	Abbreviation	Name
Ah	Arnhem	Mt	Maastricht
Amr	Alkmaar	Nm	Nijmegen
Asb	Amsterdam Bijlmer ArenA	Sgn	Schagen
Asd	Amsterdam Centraal	Shl	Schiphol
Ehv	Eindhoven	Std	Sittard
Hdr	Den Helder	Ut	Utrecht Centraal
Hrl	Heerlen	Vl	Venlo
Ht	's Hertogenbosch		

Table 3.3: Abbreviations of the stations

The instance consists of 5 train lines, each with a frequency of 2 trains per hour in both directions, leading to 20 trains in total. There are two lines that serve the whole corridor, i.e., between Ehv and Asd, whereas one line only serves Asd - Ut, and another line only serves Ut - Ehv. The fifth line (shown as a densely dotted line in Figure 3.2a) does not serve the corridor itself, but it is important in this instance to determine good frequencies on the remainder of the network that is not part of the corridor itself.

The reason to study this instance is that the ‘A2-corridor’ has a very high number of passengers and it has been subject to intensive study in practice recently, since Intercity-frequencies increased from four to six trains per hour here. In Asd and Ehv, four of the six trains continue to Amr and Std, respectively. This raises the question what the headway times should be between consecutive trains, both on the corridor itself and on the remainder of the network. As an example, if the headway times between all consecutive trains on the corridor is 10 minutes upon arrival in Ehv, and trains do not get additional dwell time there, the pattern between Ehv and Std will be irregular, headway times alternate between 10 and 20 minutes. In order to get a more regular pattern, trains would have to get a longer dwell time in Ehv. We study these kind of situations to find out what is the best solution from a perspective of total perceived passenger travel time.

In this instance, we consider only OD-pairs that travel either in the southbound or the northbound direction, and not in both directions. For example, OD-pairs Nm to Ehv and vice versa are not considered, as they would have to travel via Ut. In total the network has 34 relevant stations and we consider 891 OD-pairs. The average number of routes per OD-pair is 6.3, with a maximum of 24 routes. The event-activity network contains 1344 events and 1700 activities, of which 376 are transfer activities.

Intercity Network

As second instance, we consider the 2019 Intercity Network of NS. In this network, there are many OD-pairs without a direct connection. We thus expect that arrival and departure times at important transfer stations will be influenced by the need to make good transfer connections for these passengers. The network includes 95 stations and 76 trains in total. The geographical network is depicted in Figure 3.2b. There are 8870 OD-pairs.

The corresponding event-activity network contains 3816 events and 6578 activities, 2442 of which are transfer activities. On average, each OD-pair has 11.8 travel options, with a maximum of 280 possible options.

It is interesting to observe that when we consider passengers with a direct travel option only, only 1960 OD-pairs remain, but these cover 79.1% of the passengers.

In that case, 1760 of the transfer activities are not needed, which makes the model simpler to solve. But even in this case, 682 transfer activities are kept, since it may be beneficial for the passengers to transfer, even if there is a direct connection available.

3.6.2 Evaluation of the Solution Approach

In this section, we evaluate our solution approach, by computationally evaluating the benefit of generating a heuristic starting solution instead of a cold start, and by exploring the effects of, next to using the heuristic, solving several restricted versions of the SPOT model on the quality of the timetable.

The Advantage of Generating a Starting Solution

To evaluate the benefits of using a starting solution, we compare running times of the linearised SPOT model, with and without starting solution. We do this on three different cases: the A2-corridor instance, the Intercity network instance with OD-pairs which have a direct travel option, and the Intercity network instance with all OD-pairs. In this experiment, we do not use any of the reductions described in Section 3.5.1.

For generating a starting solution, we set $\lambda = 30$, i.e., at least 30% of the passengers in the network are included. Given the distribution of the OD-pair sizes (only a few OD-pairs account for a large portion of the passengers) and after performing several tests, this turned out to be a reasonable number to use for this purpose. This leads to including only a small subset of the OD-pairs in the model, while still ensuring that a large portion of the passengers is covered. We then employ the heuristic procedure described in Section 3.5.2.

To guide the search when no heuristic starting solution is generated, we first spend 20% of the allotted time on a model where all dwell times are set to their lower bound. The remaining 80% of the time is spent on solving the full model.

For the A2-corridor, we set a total time limit of 2 hour for the computations. For the Intercity case, we set a total time limit of 10 hours.

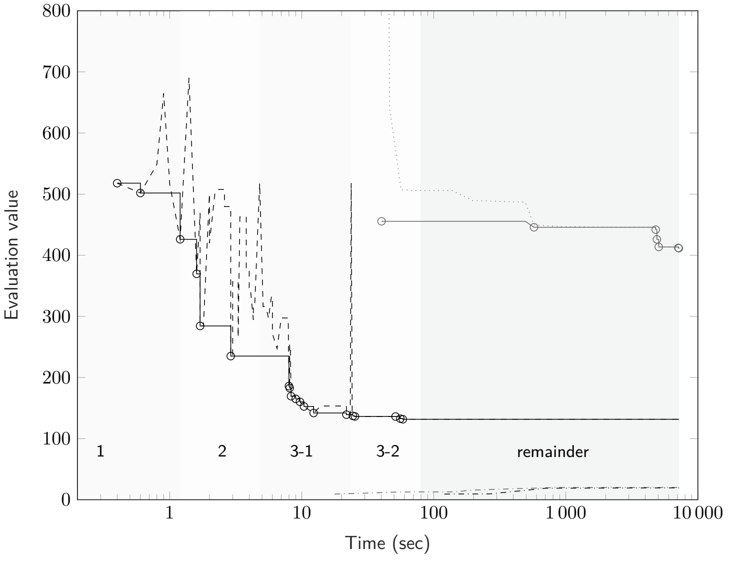
The results of our computations are shown by means of convergence plots in Figure 3.3. The horizontal axes display time in seconds on a logarithmic scale. Note that the heuristic solves a strongly restricted problem with a subset of the passengers, and therefore the objective values of the heuristic and the objective value of the full model cannot be compared. Therefore, every time a new timetable is found in the solving process, its objective value (3.11a) is evaluated based on the full set of OD-pairs, in the way described in Section 3.5.2. Figure 3.3 plots these evaluation values over time.

In the convergence plots, each individual plot displays six lines. The dashed dark line corresponds to the evaluation value of the timetables found by the heuristic. As this model has incomplete information and is a partial relaxation, not every timetable is necessarily an improvement over the previously found timetable and therefore the dashed dark line is not monotonically decreasing. In fact, every time the solver restarts in a new step of the heuristic, it may find an arbitrarily bad timetable and passenger routings, thus causing the peaks in the dashed dark line. The evaluation value of the best timetable found so far is shown by the solid dark line. The points in time in which a better timetable (according to the evaluation value based on all OD-pairs) is found is indicated by a circle. Once the heuristic finishes, its solution is fed to the linearised SPOT model as a starting solution. From that point on, a lower bound is available, which is shown as a dash-dotted dark line.

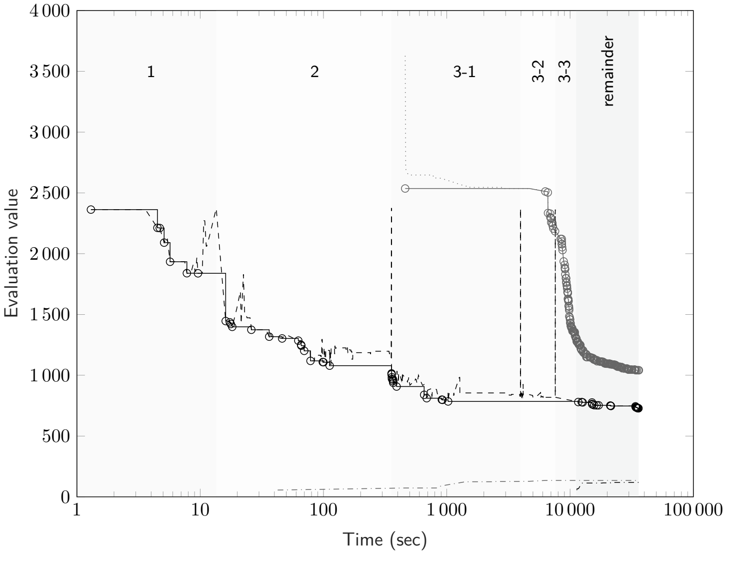
To indicate how much time is consumed for the different steps of the heuristic, we indicate the time taken by the different steps in Figure 3.3 by means of shaded bars. Each bar displays the step of the heuristic as well. As the third step iteratively changes a subset of the α variables, we also display the iteration number, i.e., 3-2 denotes step 3, iteration 2. Only a few iterations are needed to turn all α variables into integers. Step 4 is not displayed because this interval is too short to be visible on the logarithmic time scale we use. More details on how much time is spent on each step can be found in Table 3.4. This table also displays the number of OD-pairs used for the heuristic, and its percentage of the total number of OD-pairs in the instance.

	A2-corridor	Intercity network only direct	Intercity network all OD pairs
# OD-pairs	17 (1.9%)	55 (2.8%)	70 (0.8%)
Time (s):			
- Step 1	1	14	108
- Step 2	4	342	978
- Step 3-1	19	3613	3616
- Step 3-2	55	3620	3611
- Step 3-3	-	3606	2882
- Step 4	0	1	1
- Total:	79	11195	11195

Table 3.4: Details about the heuristic procedure

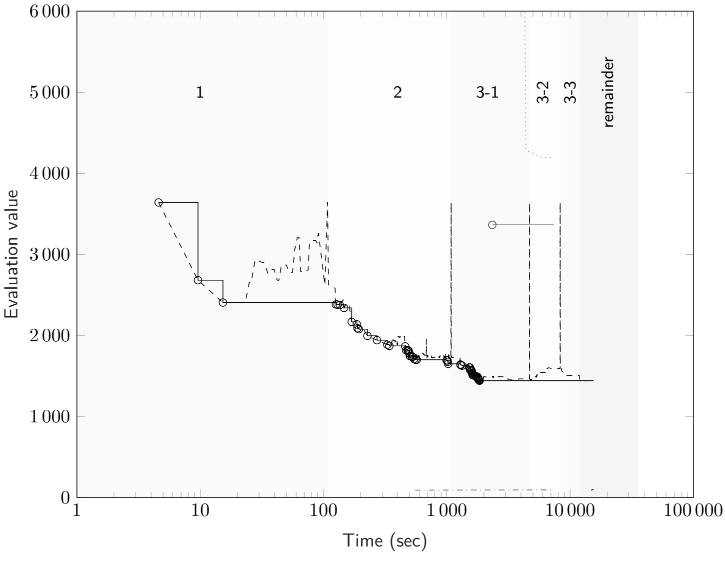


(a) A2-corridor



(b) IC network, only direct passengers

Figure 3.3: Convergence plots for the comparison of solution approaches. The dark lines display the heuristic approach, the light lines the standard approach. Solid line: best solution found so far. Dash-dotted line: lower bound. Dashed dark line: Current evaluation value. Dashed light line: Objective value according to solver.



(c) IC network with all passengers

Figure 3.3: Convergence plots for comparison of solution approaches. (cont.)

For the solution process without starting solution, the dotted light line displays the objective value according to the solver. Note that here we do not use a relaxation nor do we restrict to a subset of the passengers. However, it is still possible that in intermediate solutions, the chosen passenger routes are suboptimal for the chosen timetable. The solid light lines shows the evaluation value of the timetable found. We observe that in the beginning, there can be quite a difference between objective value reported by the solver and evaluation value, but soon better routes are found and the objective value reported by the solver decreases. The light dash-dotted line displays the lower bound according to the solver.

Characteristics of the solutions and solving process are reported in Table 3.5. For each instance, the left column displays the objective values after the time limit has passed, for the case where a heuristic starting solution is used. It shows the value of the heuristic solution, the value of the final solution, and a lower bound (provided by CPLEX). The right columns displays the values for solving the model without warm-start, and shows the final objective value and lower bound.

We observe in Figure 3.3 and Table 3.5 that although neither the full model, nor the solution approach applying the heuristic could find an optimal solution, in all three cases, better solutions are found when generating a heuristic starting solution first.

	With heuristic	Without heuristic
Instance:	A2-corridor	
Heuristic objective value:	131.6	
Final objective value	131.6	411.5
Lower bound	19.4	19.1
Instance:	IC network - direct passengers	
Heuristic objective value:	784.6	
Final objective value	728.3	1041.2
Lower bound	118.4	123.1
Instance:	IC network - all passengers	
Heuristic objective value:	1440.6	
Final objective value	1440.6	3364.1
Lower bound	98.5	98.9

Table 3.5: Comparison between using a heuristic starting solution or not

Even stronger, the heuristic finds a good solution, before the full model finds the first feasible solution.

Solving Reduced Versions of SPOT

In this section we experiment with solving different reduced versions of SPOT in the second stage.

The rationale behind this is that on the one hand, within the same time limit, we may be able to get closer to optimality when working on a reduced problem version. On the other hand, we hope that when reducing in the 'right' way, little relevant information is lost, such that the timetable we find is good when evaluated for the full problem.

To reduce the model, we experiment with four different parameters: (1) We take only passengers who have a direct travel option (only Direct), or all passengers (all OD); (2) We take $\lambda \in \{95, 99, 100\}$, i.e., we restrict the number of passengers that we take into account; (3) We either allow passengers to transfer (trans) or not (noTrans); (4) We either force passengers to take the first departing train (noWait), or let them wait for a later train (wait). When we combine (1) and (2), we include the OD-pairs which are largest in passenger size among the ones who have a direct travel option such that in total at least λ% of the passengers with a direct travel option are included. See also Section 3.5.1 for detailed explanations of the simplifications.

We follow the two-stage approach motivated in the previous section: generate a heuristic start solution, then use this to warm-start the reduced linearised SPOT model. Note that take the above parameters already into account when constructing the heuristic starting solution. E.g., when we do not consider OD-pairs with transfer options (only Direct), only these are considered in the heuristic. Furthermore, when we solve the reduced linearised SPOT model with a subset of the passengers, e.g., only 95% of the passengers, the heuristic uses a subset of this subset, that is, 30% of this 95% of the passengers.

In the following we compare the evaluation values for our approach under different parameter setting on the Intercity network instance. We chose this instance because this is the most difficult instance to find good solutions for. Note that none of the approaches finds a provably optimal solution within the given time limit.

In order to properly compare the resulting timetables, we have evaluated each of them considering all passengers with full route choice. The corresponding evaluation values are displayed in Figure 3.4.

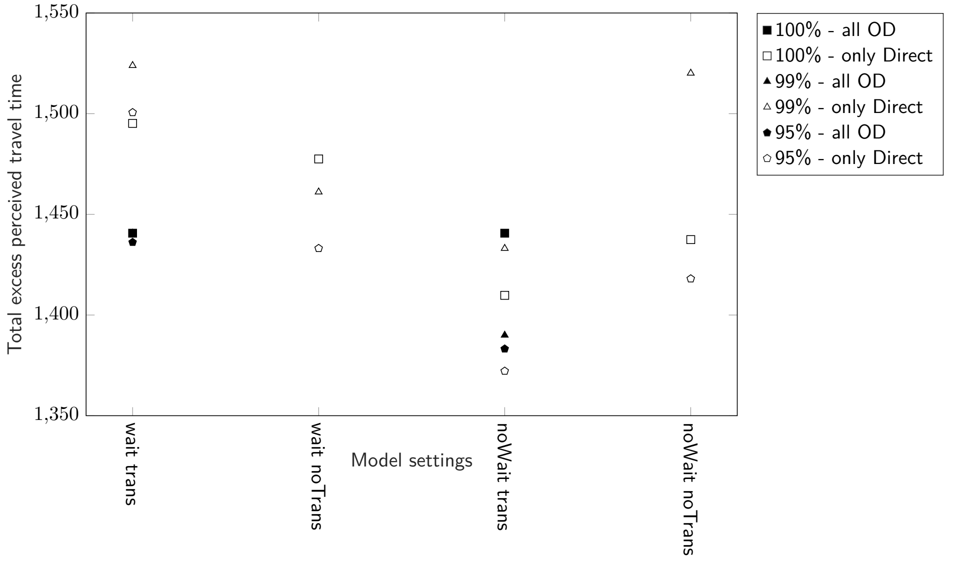


Figure 3.4: Evaluation values for Intercity network for different parameter settings

Note that when including all OD pairs, but not allowing transfers, this leads to many OD-pairs not having a transfer option in the model. Therefore, we leave out these situations.

As can be seen in this figure, no single parameter choice seems to lead to clearly superior results. As a tendency, in this instance it appears that the combination of 'noWait' and 'trans', i.e., forcing passengers to take the first train that is leaving, and allowing transfers, leads to lower objective values. A possible explanation for this is that the cases in which waiting at the station of origin is beneficial will be very limited, especially as we are studying an Intercity network in which trains run at the same speed. The option 'noWait' hence leads to a smaller model without sacrificing much in terms of quality. Next to this, transfer options can significantly improve route choices and providing them seems to be relevant. Overall, we conclude that reducing the set of OD-pairs or the travel options on the Intercity network in the proposed way is possible without leading the solution procedure towards *bad* solutions. Since these reductions allow speed-ups in the solution procedure, working on reduced models can therefore even lead to slightly better solutions if the solution time is limited.

3.7 Case Studies

In this section, we take a closer look into the solutions obtained by the SPOT model. In Section 3.7.1 we focus on the A2-corridor and demonstrate the trade-offs that our model is able to make, between additional dwell times and regularity of train service. In Section 3.7.2 we demonstrate how our method makes choices regarding transfer connections between trains on the Intercity network. We discuss our findings, and give insights on how our approach can be used in strategic railway planning.

Note that the solutions discussed in this section are the best found solutions within the time limit of the approach, and thus not necessarily optimal. In this section, we again use the two-stage approach by employing our heuristic algorithm first, and using the result as a warm start for solving the full linearized SPOT model. For the second stage we do not use any of the reductions mentioned in Sections 3.5.1.

3.7.1 Balancing Regularity and Dwell Times

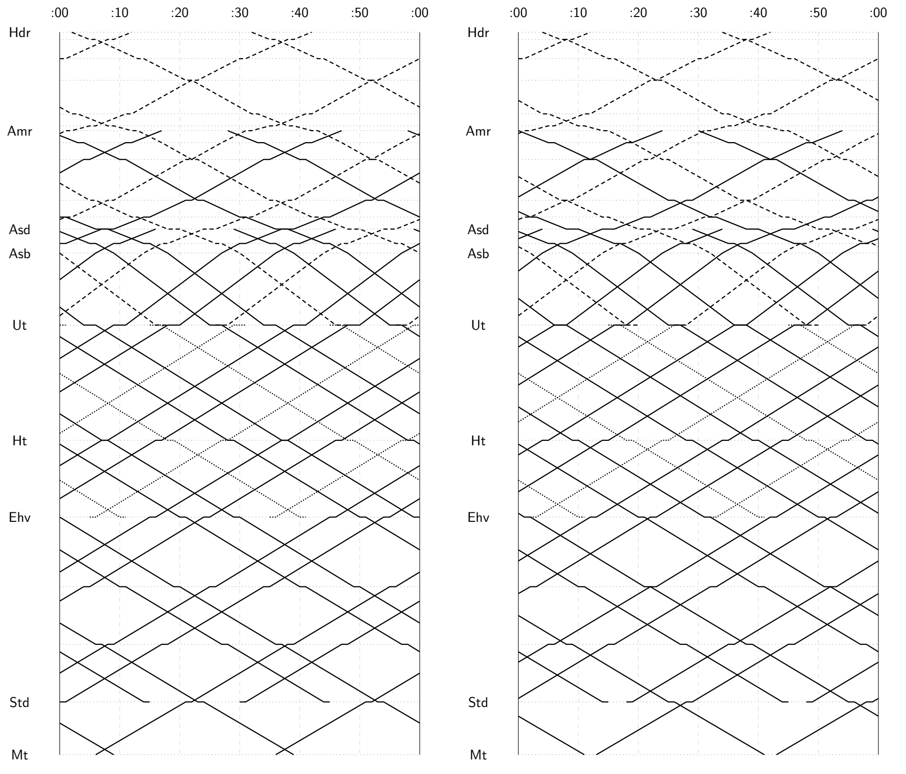
We illustrate the outcome of our approach on the A2-corridor instance in more detail. Specifically, we focus on what happens at the two locations Asd and Ehv where frequencies change from six to four trains per hour. Remember that, when passing from the corridor with frequency six to the part of the network with frequency four, if the trains arrive with 10 minutes headway time between each pair of consecutive trains and no additional dwell time at the station is allowed, the headway times will be irregular outside the corridor, alternating between 10 and 20 minutes. If we want the patterns to be regular both on and off the corridor, additional dwell times are required at Asd and Ehv, in order to make the transition between the different frequencies. In order to shed light on the trade-offs at different values of adaption time, we visualize two timetables for the A2-corridor. To find the first one, we run our solution approach for different values of parameter γ_w, which relates the perceived duration of adaption time to in-train time. In the first situation $\gamma_w = 3$, thus adaption time is considered to be less pleasant than being in the train itself. This will put a higher emphasis on the regularity of trains. In the second situation $\gamma_w = 1$, thus adaption time is valued equal to in-train time.

Time space diagrams of the timetables for both situations are shown in Figure 3.5. Time is shown on the horizontal axis, distance on the vertical axis, where also the relevant stations are shown. Three train lines are drawn with a solid line, the other two with a different line style because they are relevant for the description of the results later in this section. More detailed timetables for the stations Asd and Ehv are shown in Tables 3.6 and 3.7. Interesting to note is that the arrival pattern in Asd with $\gamma_w = 1$ is perfectly regular, all headway times between consecutive trains are exactly 10 minutes. However, when continuing towards Amr, the pattern becomes very irregular, as there is no additional dwell time added, and the headway times now alternate between 10 and 20 minutes. Also in Figure 3.5b, this irregularity is clearly visible. With $\gamma_w = 3$, the irregularity north of Asd is reduced to headway times of 13 and 17 minutes. In this case the headway times on the corridor are no longer equal to 10 minutes and vary between 8 and 13 minutes.

		$\gamma_w = 3$		$\gamma_w = 1$	
From	To	Arrival	Departure	Arrival	Departure
Mt	Amr	:07	:09	:14	:16
Hrl	Asd	:16	–	:04	–
Nm	Hdr	:24	:26	:24	:26
Mt	Amr	:37	:39	:44	:46
Hrl	Asd	:46	–	:34	–
Nm	Hdr	:54	:56	:54	:56

Table 3.6: Timetable for northbound trains at Asd

The arrival headway times at Ehv are fairly regular, they vary between 9 and 11 minutes, for both values of adaption time. Here also, departure headway times are not perfectly regular. Instead, for $\gamma_w = 3$, they are even as large as 21 minutes, which



(a) $\gamma_w = 3$ (b) $\gamma_w = 1$

Figure 3.5: Time space diagrams for Hdr to Mt.

is larger than the departure headway times at Asd. This clearly shows that in this case, the emphasis lies on reducing waiting times for passengers already in the train, and not on reducing adaption time for boarding passengers at Ehv and subsequent stations.

Interesting to note in Figure 3.5, is what happens around Ut, for both values of γ_w. The train line starting in Vl goes towards Shl (it is shown with a dotted line style in Figure 3.5) and hence the corresponding trains leave the corridor at Ut. This happens at :01 and :31 respectively when $\gamma_w = 3$. At Ut, a line coming from Nm and going to Hdr (shown as a dashed line) enters the corridor to guarantee the six train per hour frequency again. Interestingly, in both timetables, the trains of this line depart from Ut at :58 and :28, i.e., around the same times that the trains of the other line leave the corridor. That means, in the regular pattern that is present in the corridor, these two lines replace each other.

		$\gamma_w = 3$		$\gamma_w = 1$	
From	To	Arrival	Departure	Arrival	Departure
Shl	Vl	:10	–	:10	–
Asd	Hrl	:20	:21	:20	:21
Amr	Mt	:29	:30	:31	:32
Shl	Vl	:40	–	:40	–
Asd	Hrl	:50	:51	:50	:51
Amr	Mt	:59	:00	:01	:02

Table 3.7: Timetable for southbound trains at Ehv

This instance clearly shows that trade-offs are made between regularity on the corridor and on the branches on the one hand, and additional dwell times at border stations at the other hand. It shows that the value of regularity and the value of short waiting and transfer times depend on the relative value of adaption time versus in-train time γ_w.

3.7.2 Insights on the Intercity Network

The Intercity network of Netherlands Railways contains many train lines that are linked to each other, because they share part of their route and because they can offer good connections. We have analysed the timetable that was evaluated best in the end of Section 3.6.2 (that is, the one found for parameter setting 'only Direct', $\lambda = 95\%$, 'trans', 'noWait') to demonstrate how our method can be used to generate insights on the desired timetable. For the purpose of this chapter, we analyse the timetable at two different stations, namely Leiden (Ledn) and Zwolle (Zl).

Leiden (Ledn). This station is located in the western part of the Netherlands and can be reached from four different directions, which are Haarlem (Hlm), Schiphol (Shl), The Hague (Gvc) and Rotterdam (Rtd). Figure 3.6 displays a schematic overview of the line plan around Ledn. Each of the neighbouring stations is displayed, and a line indicates a train line serving this station twice per hour per direction. As can be seen, there are direct trains between each of the northern stations and each of the southern stations. The timetable of the southbound trains is detailed in Table 3.8.

Two interesting observations are worth noting here. First of all, the trains depart from Ledn two minutes apart from each other, and in a fixed order: the trains towards Gvc depart first. Furthermore, all dwell times at Ledn are exactly 1 minute, i.e., no dwell time is prolonged. Therefore, the trains need to arrive in Ledn following an

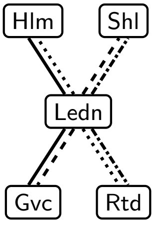


Figure 3.6: Network at Ledn

From	Arrive	To	Depart
Shl	:00	Gvc	:01
Hlm	:02	Rtd	:03
Hlm	:16	Gvc	:17
Shl	:18	Rtd	:19
Shl	:30	Gvc	:31
Hlm	:32	Rtd	:33
Hlm	:46	Gvc	:47
Shl	:48	Rtd	:49

Table 3.8: Timetable for southbound trains at Ledn

alternating pattern. They arrive in groups of two, but in subsequent groups, the order is reversed: In the first group, first the train from Shl arrives, and after that the train from Hlm, while for the second group this order is reversed. However, passengers from the first train always have a transfer connection to the second train of exactly 3 minutes. That means, in the first group of trains, passengers from Shl can travel quickly to Rtd and Gvc, while passengers from Hlm towards Gvc have to wait much longer. In the next group of trains, this order is reversed, now passengers from Shl to Gvc have to wait.
Secondly, the headway times between trains on a leg of this network are never exactly 15 minutes. For the trains coming from Hlm and for the trains towards Gvc and Rtd, it varies between 14 and 16 minutes. For the trains coming from Shl, headway times are even as large as 18 minutes. This illustrates that regularity is not a necessary condition for a good timetable.

Zwolle (Zl). Even though this station is situated in the less-populated northern part of the country, it is a crucial station in the network as all trains going further north have to pass Zl. Secondly, trains come and go in four main directions, Groningen (Gn), Leeuwarden (Lw), Amersfoort (Amf) and Lelystad (Lls). The relevant directions in this Intercity network are displayed in Figure 3.7. It is a similar situation to the Leiden station. However, in Zl, the frequency of the lines is only one train per hour. In Table 3.9, the arrival and departure times of the trains in the southbound direction are shown.

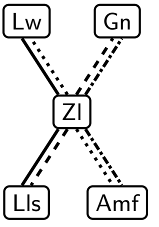


Figure 3.7: Network at Zl

From	Arrive	To	Depart
Gn	:28	Lls	:29
Lw	:30	Amf	:32
Gn	:57	Amf	:02
Lw	:59	Lls	:00

Table 3.9: Timetable for southbound trains at Zl

The trains arrive in Zl in two groups, approximately at :00, and approximately at :30, with the train from Gn arriving earlier than the train from Lw in both cases. The train from Gn to Lls arrives in Zl at :28. Passengers can either stay in their train and continue towards Lls, or transfer to the train to Amf, leaving at :32. In contrast, the passengers in the train from Lw to Amf do not have a connection to Lls and should take the direct train half an hour later. However, half an hour later, the dwell time of the train coming from Gn is prolonged, so now passengers can actually transfer between both trains. Note that, so far, we have seen few cases in which dwell time was prolonged. However, in this case, a slightly prolonged dwell time of the train in Gn to Amf can greatly reduce the travel time for passengers fron Lw to Amf, as they would have to wait approximately 30 minutes for their transfer connection otherwise. In the timetable at Ledn, we have not seen these prolonged dwell times, as this would be less beneficial there due to the higher frequencies.

The above discussion indicates how SPOT can be used to design timetables, and give valuable insights in the strategic timetabling phase. We observe that in Ledn and Zl, the solution is often close to regular, but that exceptions from these patterns can improve the timetables in some cases. This implies that we may overlook good timetables, when imposing regularity constraints. Especially in Ledn, we see that

these irregular patterns allow the alternating order of trains, and in turn the alternating connections between trains.
Secondly, we see that for some stations longer dwell times are good to ensure transfers, especially if the alternative for missing the transfer would be a long waiting time, when frequencies are low.

3.8 Conclusions and Further Research

In this chapter we introduced the Strategic Passenger Oriented Timetabling (SPOT) problem. This problem aims at finding a timetable pattern which is optimal for passengers, explicitly including adaption time into the perceived passenger travel time. In our approach to solve the SPOT problem, we formulated a quadratic integer program. We linearised it, and we proposed and tested an approach for solving it. We have shown in our case studies, how the solutions generated by the SPOT model can be used to learn about desirable patterns at key points of the network.

Due to the strategic nature of the problem at hand, we formulated the SPOT problem without including headway constraints, so that the underlying timetabling problem is relatively simple. However, the inclusion of adaption time in the model formulation leads to a quadratic objective, making the model harder to solve again. We achieved improvement with respect to the solution time by warm-starting the model with a heuristically achieved solution. Still, in none of our instances we were able to prove optimality of the solution found, with a lower bound far off the best solution found. It may be promising to investigate further solution methods, possibly working directly on the quadratic formulation of the program, or to investigate how better bounds can be determined.

Most timetabling models for the tactical planning phase do not include adaption time into the perceived travel time, thus implicitly assuming that passengers will fully adapt to the timetable and not suffer from inconveniently placed departure times. To overcome this questionable assumption, the SPOT model could also be applied to a setting that includes headway constraints in the PESP model. In addition, our current model formulation uses the assumption that passengers arrive uniformly distributed over the period for the definition of passenger groups and average waiting times. It would be worth investigating to what extend it is possible to incorporate different passenger distributions.

A different idea on how to move from the strategic towards the tactical planning phase is to consider the timetable obtained with SPOT as an ideal timetable, and adjust it, where needed, to make it ‘fit’ with infrastructure requirements. A method to do this is proposed in Chapter 4.

Appendix

3.A Linearisation

The SPOT model in (3.11) contains a quadratic objective and has several minimums in the constraints. In Section 3.4.3, constraints (3.11f) are linearised. In this section, we linearise the remainder of the model.

3.A.1 Objective

The objective function (3.11a) is a quadratic function. Using that $W_v^k = L_v^k/2$, the objective can be written as

$$\min \quad \sum_{k \in \mathcal{OD}} \frac{d_k}{T} \sum_{v \in V^k} \frac{\gamma_w}{2} \left(L_v^k\right)^2 + L_v^k \cdot Y_v^k. \tag{3.16}$$

We linearise this expression by writing L_v^k as a sum of binary variables, defined as

$$x_{v,d}^k = \begin{cases} 1 & \text{if } L_v^k \geq d \\ 0 & \text{otherwise} \end{cases} \qquad \forall\, k \in \mathcal{OD},\ v \in V^k,\ d \in \{1, \dots, T\}. \tag{3.17}$$

For a stronger formulation, we can impose the additional restrictions that

$$x_{v,d}^k \leq x_{v,d-1}^k \qquad \forall\, k \in \mathcal{OD},\ v \in V^k,\ d \in \{2, \dots, T\}. \tag{3.18}$$

Using these new variables, we can write

$$L_v^k = \sum_{d=1}^{T} x_{v,d}^k, \qquad \left(L_v^k\right)^2 = \sum_{d=1}^{T} (2d-1) \cdot x_{v,d}^k. \tag{3.19}$$

Substituting this in (3.16) results in a multiplication of binary variables $x_{v,d}^k$ by bounded variables Y_v^k. This can be resolved by introducing new variables $R_{v,d}^k = Y_v^k \cdot x_{v,d}^k$. The objective then becomes to minimize

$$\sum_{k \in \mathcal{OD}} \frac{d_k}{T} \sum_{v \in V^k} \sum_{d=1}^{T} \left[\frac{\gamma_w}{2} (2d-1) \cdot x_{v,d}^k + R_{v,d}^k \right]. \tag{3.20}$$

By defining l_v^k and u_v^k as the lowest and highest possible values for Y_v^k respectively, we add the following additional restrictions to correctly determine the value for the variables $R^k_{v,d}$:

$$R^k_{v,d} \leq u^k_v \cdot x^k_{v,d} \tag{3.21a}$$
$$R^k_{v,d} \geq l^k_v \cdot x^k_{v,d} \tag{3.21b}$$
$$R^k_{v,d} \leq Y^k_v - l^k_v \cdot \left(1 - x^k_{v,d}\right) \tag{3.21c}$$
$$R^k_{v,d} \geq Y^k_v - u^k_v \cdot \left(1 - x^k_{v,d}\right). \tag{3.21d}$$

3.A.2 Minimums

Constraints (3.11f) and (3.11i) both contain a minimum, which we can linearise. (3.11f) is already linearised in Section 3.4.3. Constraints (3.11i) are replaced by the following set of restrictions for every $k \in \mathcal{OD}$ and every $v \in V^k$:

$$Y^k_v \leq Y_r + \gamma_w \cdot Q_{v,v'} \qquad \forall\, v' \in V^k,\, r \in \mathcal{R}^k_{v'} \tag{3.22a}$$
$$Y^k_v \geq Y_r + \gamma_w \cdot Q_{v,v'} - M^k_v \cdot \left(1 - z^k_{v,v',r}\right) \qquad \forall\, v' \in V^k,\, r \in \mathcal{R}^k_{v'} \tag{3.22b}$$
$$\sum_{v' \in V^k} \sum_{r \in \mathcal{R}^k_{v'}} z^k_{v,v',r} = 1. \tag{3.22c}$$

We introduced new binary variables $z^k_{v,v',r}$, which correspond to the route that is chosen. That means, if $z^k_{v,v',r} = 1$, passengers wait from event v to v' (which can be the same), and take route r, starting at v'. For computational stability, the newly introduced constants M^k_v have to be chosen as small as possible, but still large enough to make the second set of constraints redundant if $z_{v,v',r} = 0$, i.e., we can take

$$M^k_v = \gamma_w \cdot T + \max_{r \in \mathcal{R}^k} \left\{\overline{Y}_r\right\} - \max_{r \in \mathcal{R}^k} \left\{\underline{Y}_r\right\}, \tag{3.23}$$

where $\overline{Y_r}, \underline{Y_r}$ denote the highest and lowest possible value for the variable Y_r respectively.

As we are minimizing the perceived passenger travel time, we can exclude the newly introduced constraints (3.21a), (3.21c) and (3.22a) in order to reduce the model size.

To summarize, in the linearisation we take several steps. First of all, the objective (3.11a) is replaced by (3.20). Here, additional variables $x^k_{v,d}$ and $R^k_{v,d}$ are introduced, with additional restrictions (3.18), (3.21b) and (3.21d). Secondly, the minima are replaced by linear restrictions. First of all, for notational reasons we defined $Q_{v,v'}$ in (3.13). By using this, we replace (3.11f) by (3.12), and (3.11i) by (3.22b)–(3.22c).

3.B List of Symbols

This appendix summarizes all notation used in this chapter in Table 3.10.

Sets	
V	The set of events (indexed by v, i or j)
A	The set of activities (indexed by a or (i,j))
$\mathcal{OD}$	The set of all OD-pairs (indexed by k)
$\mathcal{R}$	The set of all routes (indexed by r)
$\mathcal{R}^k$	The set of routes for OD-pair k (indexed by r)
V^k	The set of departure events for OD-pair K (indexed by v or v')
Constants	
T	The cycle period
d_k	The number of passengers for OD-pair $k \in \mathcal{OD}$
γ_w	The objective coefficient for adaption time
γ_t	The penalty for using a transfer
Variables	
π_v	The timing of event $v \in V$
p_{ij}	A modulo parameter used for the shift from one cycle period to another, for activity $(i,j) \in A$
y_{ij}	The duration of activity $(i,j) \in A$
Y_r	The duration of route $r \in \mathcal{R}$
L_v^k	The number of minutes before event v, in which no other departure event for OD-pair k takes place
W_v^k	The expected waiting time for passenger for OD-pair k, for who event v is the first departure event
Y_v^k	The perceived travel time for passengers of OD-pair k, from the timing of event v onwards
$\alpha_{v,v'}$	A binary variable ensuring the correct determination of the time difference between event v and v'
$Q_{v,v'}$	The time difference between event v and v'
$x_{v,d}^t$	A linearisation variable, indicating whether $L_v^k \geq d$ or not
$z_{v,v',r}^k$	A linearisation variable, indicating which route is chosen
$R_{v,d}^k$	A linearisation variable, replacing $Y_v^k \cdot x_{v,d}^k$

Table 3.10: List of sets, constants, and variables for the (linearised) SPOT model

3.C Proof of Lower Bound on Adaption Time

Lemma 3.1. *Let $k \in \mathcal{OD}$ be an OD-pair and V^k the set of relevant departure events, let T denote the period length. Then the adaption time contribution for OD-pair k, $\sum_{v \in V^k} \frac{d_k}{T} \cdot L_v^k \cdot W_v^k$ is bounded below by $d_k \cdot \frac{T}{2|V^k|}$, that is*

$$\sum_{v \in V^k} \frac{d_k}{T} \cdot L_v^k \cdot W_v^k \geq d_k \cdot \frac{T}{2|V^k|}. \tag{3.24}$$

Proof. Remember that $W_v^k = \frac{1}{2} L_v^k$ and $\sum_{v \in V^k} L_v^k = T$, thus the adaption time contribution of OD-pair k can never be smaller than

$$\begin{aligned} \min \quad & \frac{d_k}{T} \cdot \sum_{v \in V^k} \frac{1}{2} (L_v^k)^2 && (3.25a) \\ \text{s.t.} \quad & \sum_{v \in V^k} L_v^k = T && (3.25b) \\ & L_v^k \geq 0 \qquad \forall v \in V^k. && (3.25c) \end{aligned}$$

Consider any feasible solution $L^k = (L_v^k)_{v \in V^k}$ to this optimization problem. We can write $L_v^k = \frac{T}{|V^k|} + \delta_v$ with $\sum_{v \in V^k} \delta_v = 0$ due to (3.25b) and obtain

$$\begin{aligned} \sum_{v \in V^k} (L_v^k)^2 &= \frac{1}{2} \sum_{v \in V^k} \left(\frac{T}{|V^k|} + \delta_v \right)^2 && (3.26a) \\ &= \sum_{v \in V^k} \left(\left(\frac{T}{|V^k|} \right)^2 + 2 \frac{T}{|V^k|} \delta_v + \delta_v^2 \right) && (3.26b) \\ &= \sum_{v \in V^k} \left(\frac{T}{|V^k|} \right)^2 + 2 \frac{T}{|V^k|} \underbrace{\sum_{v \in V^k} \delta_v}_{=0} + \underbrace{\sum_{v \in V^k} \delta_v^2}_{\geq 0} && (3.26c) \\ &\geq |V^k| \cdot \left(\frac{T}{|V^k|} \right)^2 = \frac{T^2}{|V^k|}. && (3.26d) \end{aligned}$$

Hence, we obtain

$$\frac{d_k}{T} \cdot \sum_{v \in V^k} \frac{1}{2} (L_v^k)^2 \geq d_k \cdot \frac{T}{2|V^k|}. \tag{3.27}$$

□

Chapter 4

An Iterative Heuristic for Passenger Centric Train Timetabling with Integrated Adaption Times*

4.1 Introduction

In modern day rail transportation systems, many trains are full, in particular in rush hours. Combining this with an increased attention to environmental pollution in terms of carbon dioxide and nitrogen emissions, a shift towards public transportation is desired. This makes the trains even more crowded, leading to a higher number of trains on the tracks. The problem of designing a good railway timetable with limited infrastructure resources already is a challenging problem. Increasing the frequencies of trains makes it even more complicated to find a good timetable. In a good timetable, travel options for passengers are spread regularly over time, so passengers can travel whenever they like and do not have to wait a long time (Chapter 3). However, when infrastructure resources become limiting to accommodate all trains, this

*This chapter is, up to minor modifications, a direct copy of G.J. Polinder, V. Cacchiani, M.E. Schmidt, and D. Huisman (2020): *An iterative heuristic for passenger-centric train timetabling with integrated adaption times.*

regularity of train services cannot always be realised. In such a situation, trade-offs must be made between having a regular service or having increased waiting times. Furthermore, improving the level of service on one part of the network often implies a decreased level of service on other parts of the network.

In this chapter, we aim at constructing a timetable that minimizes average perceived passenger travel time, and that can be safely operated on a given infrastructural network. The perceived travel time does not only consist of the time a passenger actually travels, but also includes *adaption time*. Adaption time is the time difference between the desired moment of departure and the actual departure time. That means, if a passengers would like to depart at :40, and can depart only at :50, the adaption time is 10 minutes. As is common in many European countries, we compute a periodic timetable, i.e., a timetable for a base period that is repeated throughout the day.

We consider infrastructure constraints on a macroscopic level (Goverde et al., 2016). That is, we consider railway stations with a number of tracks connecting them. Note that we do not include infrastructure *within* stations, both only *between* station. An approach for timetabling within stations can be found in Zwaneveld et al. (1996). In macroscopic timetabling, safety constraints impose a minimum time difference between trains that share part of the tracks to avoid crossings and overtakings that are not possible due to the infrastructure and to enforce a minimum safety distance between trains running on the same track.

Many timetabling approaches impose an upper time limit on transfer times and regularity of services by additional constraints. Instead of using these strict constraints, we omit them, since our objective, the minimization of the perceived travel time of passengers, will penalize long transfer times and waiting times at origin stations. In particular, adaption time minimization allows to effectively synchronize trains at stations based on passenger demand. Furthermore we allow to cancel planned trains if that leads to a better objective value.

We define the *Passenger-Oriented Timetabling (POT)* problem as follows: Given an infrastructure network with stations and tracks connecting them, and a line plan, specifying line routes, stopping patterns and frequencies: find a timetable including all or a subset of the trains that satisfies the headway restrictions induced by the infrastructure network and minimizes average perceived travel time, where we assume that passengers will travel on shortest route according to perceived travel time.

POT can be formulated as a mixed-integer linear program combining a periodic event scheduling model with an approach for modelling average perceived travel time including adaption time as developed in Chapter 3 for strategic timetabling. However, the resulting model is very difficult to solve. Therefore, in this chapter we propose an iterative approach that combines (extended versions of) two existing approaches. First of all, we compute an *ideal* timetable, that is: a timetable that does not need to respect infrastructure restrictions, using the method of Chapter 3 for strategic timetabling. Secondly, we transform this ideal timetable into a feasible timetable, that is: make it satisfy the infrastructure restrictions, using an extension of the Lagrangian heuristic (LH) proposed in Cacchiani et al. (2010) with the goal to find a timetable that stays as close as possible to the ideal timetable, but satisfies the safety restrictions induced by the infrastructure. In a next step, we compare the resulting timetable to the ideal timetable and evaluate how the changes influence the quality of the timetable. Based on this, we provide feedback to the Lagrangian heuristic to improve the quality of the newly found timetable.

Our contribution in this chapter is threefold: First, we define the POT problem, which calls for determining a timetable that minimizes the average perceived travel time (that includes the adaption time) and satisfies safety restrictions. Secondly, we propose an iterative approach to POT that extends and combines an integer programming approach to find an ideal (but possibly infeasible) timetable with a Lagrangian heuristic to repair this timetable to feasibility, and employs feedback from the former to the latter to improve the timetable from a passenger perspective. Thirdly, we demonstrate our approach on three case studies on the Dutch railway network. We show that our algorithmic approach performs better than the alternative of directly incorporating safety restrictions in the integer program for timetabling, and that it converges to a feasible timetable very close to the ideal one.

The remainder of this chapter is organized as follows. In Section 4.2, we give an overview on research that is related to and relevant for this study. In Section 4.3 we introduce and define the POT problem in detail. Afterwards we describe our iterative approach to solve this problem in Section 4.4. We test our approach on three case studies on the Dutch railway network in Section 4.5. Finally, the chapter is concluded in Section 4.6.

4.2 Related Work

The problem of finding a train timetable using the Periodic Event Scheduling Problem (Serafini and Ukovich, 1989) is discussed in Chapter 2. In this section, we focus on extensions of this formulation and on other modelling approaches.

Timetabling based on time-space graphs. Time-space graphs constitute an alternative graph-based modelling approach to event-activity networks. In these approaches, time is discretized, and a time-expanded network is used: nodes correspond to events at specific time instants, and a path in the graph corresponds to a timetable. In time-space graph models, variables represent the choice of arcs (or paths) of this graph. Approaches based on these kind of models have mainly been used for aperiodic timetabling, although recent works have shown their effectiveness for periodic timetabling as well (Martin-Iradi and Røpke, 2019; Zhang et al., 2019).

An advantage of time-space graph models is that constraints on run and dwell times of a single train are directly embedded within the definition of the graph: only arcs corresponding to feasible run or dwell times are added. In addition, computing a timetable for a single train corresponds to solving a shortest path problem, and can be efficiently done by dynamic programming algorithms. Time-space graph models easily allow the option of not scheduling some trains by assigning them a dummy path: this is particularly useful when no feasible solution containing all trains exists, for example in highly congested networks. The drawback of these models is the size of the graph that can be extremely large for practical instances. For this reason, most approaches in this category solve the timetabling problem heuristically by decomposing it through column generation or Lagrangian relaxation, i.e., trains are scheduled in sequence. As a consequence, it is difficult to handle constraints that involve multiple trains.

Several works propose models based on time-space graphs. Brännlund et al. (1998) apply Lagrangian relaxation of infrastructure constraints, and propose a heuristic algorithm based on this relaxation that uses subgradient optimization and bundle methods. A similar approach is developed in Caprara et al. (2002) for timetabling on a corridor: safety restrictions are relaxed in a Lagrangian way, and near-optimal multipliers are obtained through a subgradient optimization procedure. Cacchiani et al. (2010) extend the time-space graph based approach from Caprara et al. (2002) to insert freight trains into an existing aperiodic timetable, staying as close as possible to given *ideal* timetables for the freight trains.

Cacchiani et al. (2008b) and Martin-Iradi and Røpke (2019) propose methods that are based on column-generation for models in which variables represent paths in the time-space graph. Zhang et al. (2019) propose a multi-commodity network flow model for periodic timetabling, and apply Lagrangian relaxation and Alternating Direction Method of Multipliers. Recently Ait-Ali et al. (2020) presented a bundle method based on a disaggregate approach, where the optimisation is performed with separate dual information for each train.

In this chapter, we extend the heuristic from Cacchiani et al. (2010) to deal with periodic timetabling, and use it in the second phase of our approach to make a given *ideal* timetable feasible with respect to safety constraints induced by the infrastructure.

Passenger-centric objective functions. There are several approaches to measure the quality of a timetable from the viewpoint of the passengers, and, as shown in Hartleb et al. (2019), the choice of evaluation approach will have an impact on which timetables are considered to be 'good' and 'optimal'. A common approach in Operations Research approaches to timetabling is to minimize the total passenger travel time. The most simple models for measuring and optimizing travel time within a PESP approach minimize a function over the weighted durations of the activities in the timetabling instance, where weights represent the number of passengers using that activity (cf. Section 2.1.3). This relies on the assumption that it is a priori known on which activities passengers travel.

Schmidt and Schöbel (2015a) propose a mixed-integer linear programming model that *integrates* (aperiodic) timetabling and passenger routing. Borndörfer et al. (2017) provide a similar, PESP-based model for the periodic case and study the impact of different routing assumptions. As PESP is already a challenging problem in itself, the integration of this problem with passenger routing makes it even more difficult to find good solutions. Schiewe and Schöbel (2018) propose an 'applicable' approach that relies on (heuristic) preprocessing and bound generation. Lübbe (2009) and Siebert and Goerigk (2013) solve the problem iteratively: first passengers are routed through the network. Based on these fixed routes a timetable is computed. Then passengers are rerouted based on the timetable. This is repeated until a stopping criterium is met.

Martin-Iradi and Røpke (2019) propose a time-space-graph-based approach to find periodic timetables minimizing passenger travel time, and include frequency constraints to guarantee that trains of the same line are spread along the cycle time.

In a column generation approach that is designed to minimize the travel times of the trains, each feasible solution found during the process is evaluated with respect to the passenger travel time, and the best solution is kept. In Farina (2019), the same problem as in Martin-Iradi and Røpke (2019) is considered and modelled on a time-space graph, and a Large Neighbourhood Search algorithm is proposed.

The above-mentioned approaches have in common that they evaluate timetables based on the assumption that every passenger will choose the shortest route (with respect to (perceived) travel time) towards his destination, just as we do in this chapter. It is well understood in transport modelling, however, that not all passengers will choose the shortest route (de Dios Ortúzar and Willumsen, 2011). Instead, in transport modelling discrete choice models are used to describe how passengers distribute over different route options. Hartleb and Schmidt (2019) investigate how to integrate passenger distribution models instead of routing along shortest routes into PESP.

Including adaption time into passenger-centric objectives. Next to the travel time between departure at the origin and arrival at the destination, also the number of travel options between origin and destination and their timing play a crucial rule in evaluating timetables from a passenger perspective (de Dios Ortúzar and Willumsen, 2011). E.g., a timetable with four travel options between origin and destination, offered every fifteen minutes, would most likely be preferred to a timetable where there is just one such option (or four, all departing at the same time), even if in the latter case the travel time is slightly shorter.

Focusing solely on passenger travel time, measured from departure at the origin in the evaluation of a timetable, neglects the effect that the spread of travel options over time has on the quality of a timetable. This can be overcome by including *adaption time* in the objective function, while making an assumption on the distribution of ‘desired departure times’ of passengers.

There are several publications on timetabling on lines and corridors, where adaption time is explicitly included in the objective function. Often, ‘adaption time’ is called ‘waiting time’ in this context. We use the term ‘adaption time’ throughout this literature review, also when referring to literature where the authors use the term ‘waiting time’, to avoid confusion with the waiting time at transfers. For single rail rapid transit lines, Barrena et al. (2014a) and Barrena et al. (2014b) propose, respectively, an exact and an adaptive large neighborhood search minimizing adaption time. A single rail line is also considered in Zhu et al. (2017), where a bi-level model is proposed: the upper level model determines the train headway times to minimize the total passenger perceived costs (given by adaption time, in-vehicle time and penalty costs associated with arriving at the destination outside the desired interval), while the lower level determines passenger arrival times at their origin stations. A genetic algorithm is used to solve it. Yin et al. (2017) proposed an integrated approach to determine train schedules and speed profiles with the aim of minimizing energy consumption and passenger adaption time. A Lagrangian based algorithm is developed for solving it for a bidirectional urban metro line. A rail corridor is considered in Niu et al. (2015), where a quadratic model to determine train timetables based on given time-varying passenger demand data is proposed. It aims at minimizing the total adaption time at stations, and is solved for a high-speed rail line.

Wang et al. (2015) propose a very detailed event-driven model for timetabling on urban networks with the objective to minimize a weighted sum of travel time (including adaption time) and energy consumption. Their solution approach is based on sequential quadratic programming and a genetic algorithm, and tested on a small network with two cyclic lines.

Instead of including adaption time into the objective function, Gattermann et al. (2016) group passengers into time slices, and add a penalty to the objective function if passengers do not depart within the respective time slice. They propose a (non-linear) PESP-based mathematical programming formulation, and transfer it to a SAT formulation to solve it. While the model allows to group passengers into (predefined) time slices and penalize deviation from the respective time slice, which is a heuristic to include adaption time. However, in the numerical experiments reported in Gattermann et al. (2016), only one time slice (that spans the whole period) is used.

In Chapter 3 we considered timetabling in the strategic railway planning phase. Like in the POT problem, that chapter aims at finding a periodic timetable that minimizes perceived travel time (a weighted sum of in-train, transfer, adaption time and transfer penalties) under the assumption that passenger demand is uniformly distributed over the period. However, in contrast to the POT problem, safety constraints are not considered, with the argument that these are not relevant in the strategic planning phase. In this chapter, we use the approach developed in Chapter 3 to compute an *ideal* timetable in the first phase of our solution approach.

Compared to the existing literature on passenger timetabling, we include the adaption time minimization in the objective, instead of having strict regularity constraints in order to gain flexibility. In addition, we consider a large railway network while most works tackle the problem on a single line or corridor.

4.3 Problem Description

4.3.1 Input

The timetable that is to be designed is based on three items: First, the infrastructure network on which the trains operate. Secondly, an origin-destination matrix representing passenger demand. Thirdly, a line plan that specifies line routes and frequencies.

As usual in tactical planning, we consider the infrastructure on the macroscopic level (see also Section 2.1). That means, the network contains the stations, a number of tracks between the stations, estimated drive and dwell times, and headway times between consecutive trains. Further details like block sections and signalling systems are not important in the tactical planning stage and can be included in a later planning stage (Radtke, 2014, Chapter 3.4).

Passenger demand is given in the form of an origin-destination matrix $\mathcal{OD}$. For each OD-pair $k \in \mathcal{OD}$, the corresponding matrix entry d_k represents the number of passengers who want to travel from the origin to the destination in one period.

A line plan specifies a set of train lines that are to be operated on the given infrastructure network. Each train line consists of a route through this network, a list of stations where the train stops and a frequency that specifies how often the line is operated per hour. We assume that all lines are operated in both directions. Note that in the line planning phase, no timetable is known yet. Therefore, while line planning can take into account constraints on the infrastructure utilization, and on eligible frequencies, it is not ensured that there exists a feasible timetable where all trains specified in the line plan can be operated, see also Odijk et al. (2006). Therefore, we allow our method to cancel trains if necessary.

4.3.2 Passenger Oriented Timetabling

Based on the infrastructure network, the demand encoded in the OD-matrix $\mathcal{OD}$, and the line plan, the *passenger-oriented timetabling (POT)* problem can be summarized as

$$\text{Minimise} \sum_{k \in \mathcal{OD}} d_k \cdot R_k(\pi) \tag{4.1a}$$

$$\text{Such that TimetablingRestrictions}(\pi) \tag{4.1b}$$

$$\text{RoutingRestrictions}_k(\pi) \qquad \forall\, k \in \mathcal{OD}. \tag{4.1c}$$

Here, π is the timetable, d_k is the demand for OD-pair k and $R_k(\pi)$ is the average perceived travel time for the passengers of OD-pair k.

We aim at finding a timetable π that is operationally feasible (as ensured in constraints 4.1b) and integrate the routing of passengers through the network (constraint (4.1c)) in such a away that the average perceived travel time (objective (4.1a)) is minimized.

Timetabling restrictions. The timetabling restrictions (4.1b) can be formulated as standard PESP constraints. This is detailed in Chapter 2, with the particularity that synchronization constraints and upper bounds on transfer times are not included in this chapter. However, each PESP-activity requires an upper bound, hence transfer activities as well. We deal with this by setting for such $(i, j) \in A$ the upper bound to $u_{ij} = \ell_{ij} + T - 1$. Due to this, all time differences between events i and j are possible, i.e., there is no operational restriction. When for example $\pi_j = \pi_i + 2$ and $\ell_{ij} = 3$, the time difference between events i and j is two minutes, but also any multiple of T minutes can be added (due to the Tp_{ij} term in the PESP constraints, see Section 2.1.1). In this case, the actual transfer time is not 2 minutes, but 62 (when $T = 60$), since $\ell_{ij} = 3$ en $u_{ij} = 62$. By this way of modelling, the proper transfer times for passengers can be determined. The same principle holds when upper bounds of, for example, trip and dwell times are omitted. In this way, no operational restrictions are added. In Section 4.4, when our model is explained, these transfer activities are used to determine the correct passenger paths and their perceived duration.

Passenger route choice and evaluation of the timetable. Constraints (4.1c) are auxiliary constraints that compute passenger routes according to shortest perceived travel times in the timetable, to be able to evaluate the timetable with respect to total perceived travel time in the objective (4.1a). We model this following the approach outlined in Chapter 3. The perceived travel time consists of the following components:

Adaption time This is the time difference between the desired departure time of the passenger and the moment the train departs that brings him to his destination. The adaption time is weighted by a factor γ_w.

In-train time This is the time the passenger actually spends in the train, both when the train is driving and when it dwells at a station.

Transfer time This is the time a passenger has to spend on some station to transfer from one train to another. The transfer time is weighted by a factor γ_s.

Transfer penalty If the passenger needs to transfer from one train to another, a penalty of γ_t is added for each transfer. This is done to model the fact that passengers in general do not like to have a transfer (De Keizer et al., 2015).

By optimizing with respect to total perceived travel time, which includes the adaption time, we are able to exclude synchronisation constraints and upper bounds on transfer times from our modelling, and thus trade-off the synchronization on different part of the network in our model. This is illustrated by means of an example in Section 3.1.

Note that while the adaption time depends both on the chosen route and on the desired departure time of the passenger, in-train time, transfer time, and transfer penalty are characteristics of the route. We therefore refer to the weighted sum of these as *perceived route length* in the remainder of this chapter.

To evaluate the timetable, we follow the approach described in Chapter 3, which we briefly restate here. We assume that passenger demand per OD-pair is distributed uniformly over the period. I.e., every time unit (in our model: every minute) $\frac{d_k}{T}$ passengers would like to depart from the origin station of OD-pair k to travel to the destination station of OD-pair k. The rationale behind this assumption is that the timetable is usually constructed a number of years to six months before the actual day of operation, and we cannot expect that time-dependent demand is known accurately.

To compute the average perceived travel time for an OD-pair k, $R_k(\pi)$, we group passengers according to their first possible departure time and the routes they would take correspondingly. This allows us to compute in-train time, transfer time, transfer penalty and average adaption time per route and weigh it with the corresponding passenger number, to obtain $R_k(\pi)$ as a weighted average. If, due to train cancellations, there is no route from origin to destination of OD-pair k, we set $R_k(\pi) := M$, where M is a (high) penalty value. In the remainder of this chapter, we refer to $d_k \cdot R_k(\pi)$ as the *evaluation contribution* of OD-pair $k \in \mathcal{OD}$.

To evaluate the timetable we sum up the evaluation contributions of the OD-pairs and obtain

$$\sum_{k \in \mathcal{OD}} d_k \cdot R_k(\pi) \tag{4.2}$$

which we use as objective function (4.1a) in our problem.
For a more detailed description of the grouping of passengers, we refer to Chapter 3.

4.3.3 Lower Bound and Excess Evaluation Contribution

Based on minimum drive, dwell, and transfer times and penalties, we can compute lower bounds on perceived route lengths. Furthermore, by predetermining routes for each OD-pair, we can compute a lower bound on the total adaption time of one OD-pair, by assuming that departure times for this OD-pair are perfectly synchronized. The sum of the lower bound for perceived route length for an OD-pair, multiplied with the number of passenger for this OD-pair, and the lower bound on adaption time, gives us a lower bound on the OD-pair's perceived travel time. We refer to the difference between the evaluation contribution of an OD-pair and the so-computed lower bound as *excess evaluation contribution*. The sum of lower bounds over all OD-pairs provides us with a lower bound on the evaluation value of the timetable.

4.4 Solution Approach

This section describes the algorithmic approach that we use to solve the problem. We first outline the overall approach, before we describe the involved steps in more detail.

4.4.1 High Level Description of the Solution Approach

It is possible to extend the mathematical programming formulation for strategic passenger-oriented timetabling from Chapter 3 to include safety constraints (headway, overtaking and crossing contraints) that model infrastructure requirements, by adding more PESP-constraints. In that way, POT can be modelled as a mixed-integer (quadratic) program. However, the incorporation of the safety restrictions leads to a strongly interconnected event-activity network, which in itself leads to a challenging PESP problem to solve on networks of realistic size (cf. Goerigk and Liebchen, 2017;

Liebchen et al., 2008). Combined with the variables and constraints introduced to the model and the objective function, the problem formulation becomes unsuitable to solve large real-world instances on general purpose IP solvers. In Section 3.6.2, it is motivated that this is already challenging for instances without infrastructure constraints.

Therefore, in this section we propose an iterative approach. A graphical overview of our approach is shown in Figure 4.1. The numbers mentioned in the sequel refer to the numbers in this figure.

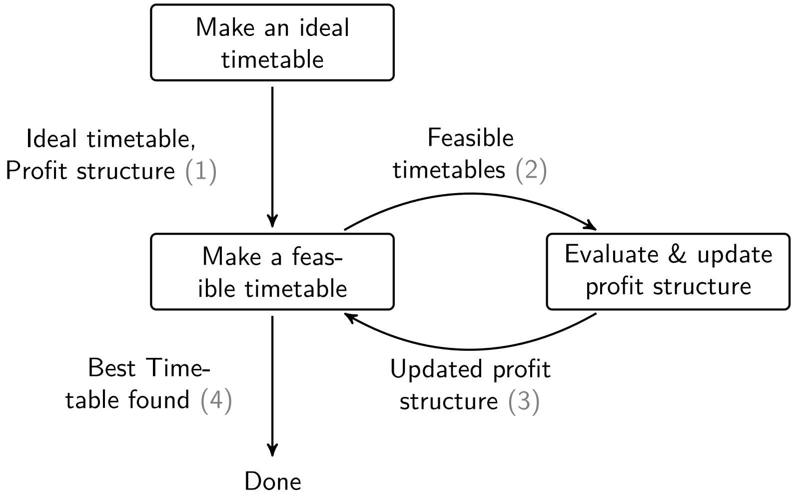


Figure 4.1: Flow diagram of our approach

In a first step, we construct an 'ideal' timetable (i.e., one which does not need to take safety restrictions into account) using the solution approach for the strategic timetabling problem from Chapter 3. An extension of the Lagrangian heuristic (LH) from Cacchiani et al. (2010) is used to modify the timetable to make it feasible with respect to infrastructure restrictions, while staying as close as possible to the ideal timetable. As LH requires a particular structure for the objective (as outlined in Section 4.4.2), a transition has to be made from one module to the next. This is done by specifying a profit structure for each train in the line plan, that is based on the relative importance of the trains (1). In fact, different profit structures are used, to generate a pool of feasible timetables with LH. We detail in Section 4.4.2, in the part about new features of LH, how profit structures are chosen. We evaluate all feasible timetables from the pool of timetables found with LH (2) with the evaluation function (4.2). Based on the evaluation values, we select one or several feasible timetables for comparison with the ideal timetable. We check for which OD-pairs the evaluation contribution is improved became better and for which it gets worse. Based on this, we update the profit structure (3) and rerun LH to hopefully find a better timetable.

We repeat this procedure, until no improvements are found any more. We then end with the best found timetable (4). How to find an ideal timetable is already explained in Chapter 3. The other phases of the algorithm are explained in more detail below.

4.4.2 Make a Feasible Timetable

In this section, we focus on the second phase of our solution approach. After an ideal timetable is computed, we compute a timetable that is feasible, i.e., satisfies all the safety restrictions, and is as similar as possible to the ideal one. A Lagrangian heuristic algorithm (LH) is used in this second phase. It extends the method proposed in Cacchiani et al. (2010), where it has been applied for solving a non-periodic train timetabling problem. For the sake of clarity, we briefly describe the main steps of LH, and refer to Cacchiani et al. (2010) for further details. Then, we present the new features added to LH in order to cope with additional real-world constraints.

Main Steps of LH

LH takes as input the description of the infrastructure network, an ideal timetable, and a profit structure. The ideal timetable contains, for every train, the desired departure and arrival times at every visited station. In order to derive a feasible timetable, LH can change the ideal timetable (i) by moving (earlier or later) the departure time of some trains from their origin stations (*shift*) and consequently moving the arrival and departure times, at all the stations visited by the train, by the same amount, (ii) by increasing the dwell time at some of the visited stations (*stretch*) and (iii) by cancelling trains. Each of these changes is undesirable, and thus it is penalised, but not all changes have the same importance. Clearly, train cancellation has a deeper impact on passengers, as the line frequency is reduced or even some OD-pairs might not have a travel option to reach their destinations. However, shift and stretch also affect the passenger travel as they influence the adaption, in-train and transfer times. In addition, the same change applied to different trains (e.g., intercity versus local trains, high- versus low-frequency train lines) has different consequences.

In order to obtain a feasible timetable that is as similar as possible to the ideal one, and also to give different importance to the different changes, we define a *profit structure*. In particular, each train is associated a *train profit* $theta_0$, a *shift penalty*

ρ^{sh}, a *stretch penalty* ρ^{st}, a *maximum shift value* λ^{sh}, and a *maximum stretch value* λ^{st}. The train profit corresponds to the importance of scheduling the train, and is decreased by ρ^{sh} for every minute of shift, and by ρ^{st} for every minute of stretch. The maximum shift and stretch values represent the bounds on the changes of the respective type that can be applied to obtain a feasible timetable. Note that, as in Cacchiani et al. (2010), we do not consider the option of decreasing the dwell time at a station, nor that of increasing the train travel time between consecutive stations.

To represent the train timetabling problem, LH uses a time-space multi-graph, in which every node corresponds to a train event, i.e., to a departure or an arrival time of a train from/at a station along a track. Arcs represent the travel of a train between two consecutive stations or the stop of a train at a station, and are partitioned into arc sets, one for each train. Different trains can have different trip and dwell times, but it is also possible that two (or more) trains have the same departure and arrival nodes: therefore, there can be multiple arcs (of different trains) between the same nodes, i.e., we deal with a multi-graph. A *path* in this time-space multi-graph corresponds to a train timetable that respects the train trip and dwell times. An Integer Linear Programming (ILP) model based on this time-space graph contains one binary variable for each arc, that assumes value one if the arc is selected in the solution. In this ILP, each arc is assigned a *profit* that is used to obtain a timetable as close as possible to the ideal one. The profit associated with the travel arc of a train from its origin station to the consecutive one is given by the train profit decreased by the shift penalty counted for every minute of shift incurred by that departure time, i.e., it is $\theta_0 - \rho^{sh} \cdot \delta^{sh}$, where δ^{sh} corresponds to the number of minutes of shift. The profit associated with each arc corresponding to a stop at a station is zero, if that arc corresponds to the minimum dwell time, or is decreased by the stretch penalty counted for every minute of stretch incurred at that station, i.e., by $\rho^{st} \cdot \delta^{st}$, with δ^{st} being the number of minutes of stretch. Hence, for any train t, the total profit θ^t of this train is computed (with a slight abuse of notation) as $\theta^t = \theta_0 - \rho^{sh} \cdot \delta^{sh} - \rho^{st} \cdot \delta^{st}$. Clearly, if a train is cancelled, no profit is obtained.

The objective of LH is to maximise the total profit of all trains, i.e., to maximize $\sum_{t \in \mathcal{T}} \theta^t$, with $\mathcal{T}$ being the set of trains. The constraints require to select, for each train, arcs that form a path in the time-space graph, and do not conflict with arcs selected for any other train, i.e., satisfy all safety restrictions. Obviously, the difficulty of solving the problem comes from the latter constraints: therefore, LH applies a Lagrangian relaxation of all these constraints. This allows us to easily compute the solution of the relaxed problem (*Lagrangian solution*) by dynamic programming, since it consists of finding, for each train, the most profitable path in the time-space graph. In order to improve the Lagrangian multipliers associated with the relaxed constraints, LH iteratively executes a subgradient optimization procedure, in which Lagrangian multipliers are updated and added to the arc profits, so as to take into account the constraint violations or looseness. Meanwhile, at each iteration of the subgradient optimization procedure, to determine a feasible timetable, LH applies the following steps: (i) it orders trains based on their profits in the Lagrangian solution (*Lagrangian profit*), (ii) schedules one train at a time in the most profitable way while avoiding all conflicts with the previously scheduled trains, and (iii) applies a local search procedure that tries to find a better path for one train at a time (if it had shift, stretch or was cancelled) by keeping all other paths as fixed, and this time considering the original arc profits.

New Features of LH

Adapting LH to periodic timetabling. The version of LH developed in Cacchiani et al. (2010) is developed for a non-periodic train timetabling problem. However, it is capable of handling a 'periodicity' of one day: namely, in the non-periodic timetabling problem, the timetable was repeated in the same way every day. In this chapter, we apply LH to a periodic train timetabling problem, where the cycle time is one hour: in this context, beside changing the length of the period, we need to ensure that the duration of the total shift time window (earlier and later shift) plus the total stretch is smaller than the cycle time: $2 \cdot \lambda^{sh} + \lambda^{st} < T$. If this is not guaranteed, then we cannot uniquely define the shift or stretch penalty of an arc: for example, a shift of one minute would not be different from a shift of sixty-one minutes. We note that this limitation is reasonable, since usually at least two trains with the same origin and destination stations should be scheduled in each cycle time, and thus it is not useful to globally shift or stretch a train more than the cycle time.

Rolling stock restrictions. Another change is applied to the original LH as described in Cacchiani et al. (2010) in order to deal with basic rolling stock constraints: in practice, usually trains of the same line (i.e., trains with the same origin and destination stations, and stopping at the same intermediate stations) are scheduled in 'pairs', so that when a train is scheduled in one direction, another train is also scheduled in the opposite direction. The reason is that, in this way, the same rolling stock (physical train) is assigned to both services, and we also obtain a more regular timetable that has the same number of trains running in both directions. In general, according to the line frequency, there can be more than two trains of the same line in a period: in this case, we need to guarantee that the same number of trains is scheduled in both directions. Since LH allows train cancellation as one of the changes that can be applied to obtain a feasible timetable, we need to guarantee that, if a number of trains of a line is cancelled in one direction, then the same number of trains is also cancelled in the opposite direction. Clearly, this can be easily obtained by simply cancelling additional trains: however, cancelling trains is highly undesirable. Therefore, we modify LH by including a new procedure as follows. At each iteration of the subgradient optimization procedure, when a feasible timetable has been determined and the local search procedure has been executed to improve it, we check, for every train line, the number of trains cancelled in each direction. If this number is not the same in both directions, then we cancel additional trains, so that the same number of trains is scheduled in both directions. Once all train lines have been processed, for each train line, we try to reschedule trains in pairs by computing, for each train, the most profitable path compatible with the previously scheduled ones: this computation is performed by dynamic programming considering the original arc profits. Note that it is possible to schedule previously cancelled trains thanks to the additional train cancellation applied at the beginning of this procedure. If the same number of feasible paths is found for both directions of a line, then the corresponding train paths are fixed in the execution of this procedure, otherwise trains are cancelled again. Since this procedure can change the set of scheduled trains, after executing it, we apply the local search procedure to possibly further improve the timetable. We observe that, for the considered instances, the number of trains per line is two or four, and thus this procedure can be executed efficiently.

Intermediate shift penalties. A final extension we developed is used for the feedback process, that is applied after the timetables have been evaluated. In this process, LH takes as input the infrastructure description, the ideal timetable, and a new profit structure. Beside the possibility of updating shift and/or stretch penalties, we also include, during the feedback, the option of penalising the shift at some intermediate stations visited by a train (so not only at the origin station of the train). Indeed, we observed that the timetables produced by LH sometimes show irregular departure headway times from some intermediate stations, and it is thus useful to penalise this irregularity. To this aim, we add, for every minute of shift, an *intermediate shift penalty* ρ_s^{is} to the profit of each arc that corresponds to the stop of a train at a station s: based on the dwell time at the station, the departure time from that station determines the corresponding intermediate shift. LH is then executed by considering these additional penalties in the computation of the Lagrangian solution and of the feasible timetable, and in the local search procedure.

Note that adding an intermediate shift penalty at a station where a train line starts, has no effect on this specific train line, since there is no arc associated with a dwell time of this train line at this station. However, for departing trains of this line, we do have the regular shift penalty, but this is not penalized on an arc associated with a stop of a train.

4.4.3 Evaluate and Update Profit Structure

In the remainder of this section, we describe how we provide feedback and update the profit structures, in order to find better timetables.

We evaluate each timetable generated by LH using the evaluation function (4.2). The best timetable according to this evaluation function after running LH is referred to as the *best pure Lagrangian* (BPL) timetable. By comparing the evaluation contributions of all OD-pairs in the ideal timetable π and in the BPL-timetable π', we can identify the OD-pairs for which the evaluation contribution increased the most. We inspect the routes chosen for these OD-pairs and the corresponding trains to find the reason of the increase. In a 'feedback' step, we generate a new set Ψ of updated profit structures, based on the initially chosen profit structure S to penalise the undesired changes more. Since we are not able to predict by how much we should penalize deviation from the ideal timetable, we use a set of penalty values P and create several profit structures based on S and P. We proceed as follows:

1. Identify the OD-pairs for which the evaluation contribution increased the most. This increase can be caused by high passenger numbers or by a high increase in the perceived travel time. We refer to these OD-pairs as *relevant* OD-pairs in the remainder of this section. Let $\mathcal{O} = \{o_1, o_2, \ldots, o_\kappa\}$ be the set of origin-stations for the relevant OD-pairs.

2. We create $|\mathcal{O}| \cdot |P|$ new profit structures, one for each combination of penalty values from P at each station. We create the corresponding profit structure as follows: for each station $o_i \in \mathcal{O}$ and for each $p_j \in P$, we create a new profit structure that is based on S, with an additional intermediate shift penalty of value p_j that is assigned to all trains passing station o_i. Furthermore, all trains

for which o_i is an origin or terminal station and which are *relevant* for the corresponding OD-pair identified in step 1, receive shift penalty $\max\{\rho^{sh}, p_j\}$, where ρ^{sh} is the regular shift penalty for train t in profit structure S.

If $|\mathcal{O}| > 1$, we generate additional profit structures. In these new profit structures we apply the same principle as above, but now we apply the penalties to all *pairs* of stations. That means, for each $o_i, o_j \in \mathcal{O}$ and for each $p_m, p_n \in P$, apply (intermediate) shift penalty p_m to station o_i and penalty p_n to station o_j.

Given the set of updated profit structures Ψ, we again run LH, as outlined in Section 4.4.1. Each of these profit structures leads to a new timetable which we evaluate. If any of these timetables gives a better evaluation value, we stop the feedback process and finish with the best timetable generated using Ψ. Else, execute steps 1 and 2 again with the original profit structure as input, but now also identify OD-pairs as relevant for which the evaluation contribution increased the most in the best timetable generated using the profit structures in Ψ. The timetable that is the best after providing feedback is referred to as the *best after feedback* (BF).

We underline that the intermediate shift penalty is not adopted at every intermediate station where there are irregular departure headway times, but only at those that cause a significant increase in the evaluation value. Indeed, it would not be effective to penalise shifts at every station, since some changes are needed in order to get a feasible timetable. Therefore, we aim at penalizing the changes that have most impact on the evaluation value of the timetable. For the same reason, we do not use the intermediate shift penalty when LH is applied to the ideal timetable in the first round before the feedback process.

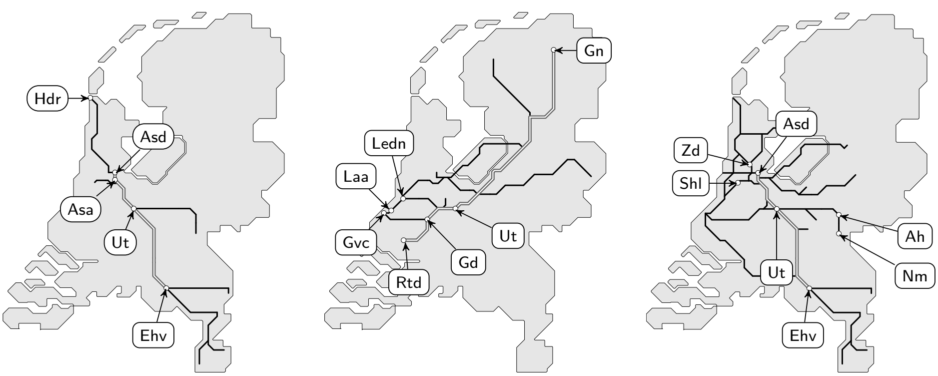
The rationale behind our feedback approach, is that adaption times have a strong influence on the evaluation value of a timetable. If LH causes a higher irregularity in the new timetable compared to the ideal timetable, the evaluation value is likely to increase. For this reason, we focus on the shift penalties in the feedback process. However, many alternative strategies to provide feedback in order to reduce the evaluation contributions of OD-pairs can be thought of, as we can update initial train profits, shift and stretch penalties, maximum shift and maximum stretch, as well as adding intermediate shift penalties. We experimented with different strategies on how to update profit structures before we identified this one which was successful on our three test instances and that is presented here.

4.5 Case Study

In this section we perform three case studies. First, we describe the three instances in Section 4.5.1. Next, in Section 4.5.2 we describe the parameters that are used in our approach. In Section 4.5.3, we describe and discuss the obtained results. Finally, in Section 4.5.4 we benchmark our iterative approach with solving the POT model as an integer program.

4.5.1 Instances

Here we describe the instances that we consider in more detail. Each instance is based on a central corridor. Figure 4.2 displays for each instance an overview of the network, the central corridor on which the instance is based is highlighted.



(a) A2-corridor (b) Rotterdam-Groningen (c) Extended A2-corridor

Figure 4.2: Networks of the three instances considered

A2-corridor instance

The first instance we consider is the ‘A2-corridor’, which is a corridor between the stations Eindhoven (Ehv) and Amsterdam Central (Asd). The network contains 34 stations. Furthermore, five train lines are operated on this network with a frequency of two trains per hour in both directions, so there are 20 trains in total. All of them are Intercity-lines. The map of the corresponding network is shown in Figure 4.2a. We label five out of the 34 stations because they are used in the feedback of our approach. The stations Ehv and Asd are the ends of the central corridor, as is indicated by the highlighted line.

In this instance, we consider 891 OD-pairs in total. The underlying event-activity network contains 1344 events and 1460 drive and dwell activities. Note that these events do not only cover arrivals and departures at stations where the train stops, it also covers departure and arrival times of stations that the train passes, as also in these locations, consecutive trains need to satisfy the headway time of 3 minutes. To build the model for constructing the ideal timetable, 376 transfer activities are included to ensure the transfer possibilities. Note that they do not impose operational restrictions on the timetable as explained in Section 4.3.2. For a full mathematical programming formulation of POT, infrastructure constraints should be added. This leads to 2964 additional activities modelling safety distances.

Rotterdam-Groningen instance

The second instance covers part of the 2019 line plan of Netherlands Railways (NS, 2017). It is centered on the line between Rotterdam (Rtd, in the South-West) and Groningen (Gn, in the North-East). This line is highlighted in Figure 4.2b. All lines that share a part of their route with the indicated line are added. Note that a consequence of this way of instance construction is that some, but not all lines operating on the network arcs indicated in black are included in our instance. The network contains 77 stations, of which 6 are labelled because they occur in our description of the results. In total we have 60 trains in the network, of which 26 are Intercity trains.

The underlying event-activity network of this instance contains 1664 events and 1716 drive and dwell activities. The model for constructing the ideal timetable has 1402 additional transfer activities to enable all passenger routes. In order to deal with the restrictions modelling safety distances, 4004 additional activities are needed. There are 3810 OD-pairs. Note that, although there are many more trains compared to the A2-corridor instance, the number of events, as well as of drive and dwell activities, increased only slightly compared to the A2-corridor. The reason for this is that we now have a number of trains which only have a short route (and thus less events are needed per train line). The main increase is seen in the number of transfer activities, to generate all possible transfer routes, as well as in the number of activities modelling safety distances.

Extended A2-corridor instance

The third instance is an extension of the A2-corridor instance. The line plan is based on all train lines in the 2019 network of Netherlands Railways that share a part of their route with the corridor between Amsterdam Central (Asd) and Eindhoven (Ehv). This corridor is highlighted in Figure 4.2c, which also shows the remainder of the network, including the locations of several stations that we use in the discussion of the results later on. The total number of trains in this network is 88 and the network contains 140 stations.

This instance considers 11121 OD-pairs. The event-activity network contains 3160 events and 3308 drive and dwell activities. Furthermore, 3592 transfer activities are added to enable all passenger routes in the model to construct the ideal timetable. Adding the restrictions modelling safety distances leads to 8360 additional activities.

4.5.2 Parameters

This section describes the parameters that we use in our experiments. First, we describe instance parameters, as well as the parameters for the objective function. Next, we detail the profit structure that we use for LH.

To compute an ideal timetable, an integer programming problem has to be solved. The same has to be done for solving POT directly, as we do in Section 4.5.4. For this, we use a machine with an Intel Xeon Silver 4110 2.10Ghz processor with 96 GB of RAM installed. These mathematical programs are solved by Cplex 12.9.0 under default settings, using up to 15 parallel threads (IBM, 2019).

Instance and objective function parameters

In all our experiments we discretise time to minutes and use a period length of one hour, i.e., $T = 60$. In each of the instances, we take a headway time of three minutes into account between two trains leaving or entering a station. Furthermore, trains cannot overtake each other between two stations. When trains in opposite directions must be separated in time when entering or leaving a station, we require this time to be at least one minute. If there are two tracks between stations, trains in the same direction will share one track, the other track is for the other direction. Finally, if there are four tracks available, two tracks are used per direction. The Intercity trains use one track and the local trains use the other, the same holds for the other direction.

The perceived travel time for the passengers consists of in-train time, transfer time, a transfer penalty, and the adaption time. In line with Chapter 3 and De Keizer et al. (2015), we set $\gamma_t = 20$, i.e., the transfer penalty equals 20 minutes. Secondly, we use $\gamma_s = 1$, i.e., transfer time weighs as much as in-train time. This is done since we already have a transfer penalty. Finally, we take $\gamma_w = 3$, i.e., adaption time weights three times as much as in-train time.

If passengers no longer have a travel option when trains are cancelled, we add a penalty of value M to the average perceived travel time of these passengers. In our experiments, we set this value M to be $24 \cdot T$, i.e., it equals a full day of travel time.

Finally, since the numbers in the OD-matrix are confidential, we scale all evaluation values. That is, we divide all evaluation numbers by the evaluation value of the ideal timetable, and multiply this by 100, so the evaluation value of the ideal timetable is indexed to 100. That means that an increase in the evaluation value by one unit means the evaluation value is 1% higher than that of the ideal timetable.

Chosen profit structure

As explained in Section 4.4.2, the Lagrangian heuristic requires the specification of a profit structure. Different profit structures can produce different timetables. As an initial profit structure, we define the train profits based on the train type (Intercity, local, etc.) and also on line frequencies. For the train type ‘Intercity’, we consider a base profit of 4000. For the train type ‘local train’, the base profit is reduced by 10% to 3600. For trains that partly operate as an Intercity and partly as a local train, the base profit is reduced by 5% (to 3800). Then, for each pair of consecutive stations that the train visits, we identify the number of trains that travel between the same pair of consecutive stations, and take the minimum m of these numbers along the train route. The train profit θ_0 is then computed as its base profit divided by m. As an example, if we consider a local train line, whose line frequency is two, and that is the only train line offering a service on some part of the network, then on this part of the network, there are only two trains and hence $m = 2$. Then the profit for the trains in this line is $\theta_0 = (4000 \cdot 0.9)/2 = 1800$. If another line with frequency two is present as well on the considered part of the network, we have $m = 4$ and the profit is 900.

For the shift and stretch penalties (ρ^{sh} and ρ^{st}), we consider equal values for all trains, and globally three alternative options: (1) $\rho^{sh} = 20$ and $\rho^{st} = 10$; (2) $\rho^{sh} = \rho^{st} = 15$; (3) $\rho^{sh} = 10$ and $\rho^{st} = 20$. Namely, we assign more importance to the shift penalty in the first case, same importance to both changes in the second case, and more importance to the stretch penalty in the third case. Indeed, it is not known a priori whether a shift or a stretch is worse: it depends on the location where this happens and what the influence is on the regularity of trains in general. In addition, we want to explore a rather broad spectrum of profit structures, because it is not a priori known which changes in the timetable have the least negative effect on the evaluation of the timetable according to evaluation function (4.2).

As maximum shift and maximum stretch (λ^{sh} and λ^{st}), we also consider the same values for all trains, and start by setting $\lambda^{sh} = \lambda^{st} = 5$: this means that each train can have its departure time from its origin station up to 5 minutes earlier or 5 minutes later, and a total stretch along its route of up to 5 minutes. Then, we also consider two other options, that increase the possibilities of scheduling trains: $\lambda^{sh} = 10$ and $\lambda^{st} = 5$, and $\lambda^{sh} = 5$ and $\lambda^{st} = 10$. Indeed, when the maximum shift and stretch are set to small values, it might not always be possible to schedule all trains, leading to low quality of the solutions. Overall, by combining the different shift/stretch penalties and the maximum shift/stretch, we thus have 9 different profit structures that lead to 9 timetables. The total number of iterations for each run of LH is set to 250 in our experiments.

In the feedback process, we set values for the intermediate shift penalties ρ_s^{is} at some stations s, as well as for the initial shift penalty ρ^{sh} of a train starting in such a station. We use values of 10, 20 and 30 for these penalties.

4.5.3 Results of the algorithm

We execute our timetabling approach presented in Section 4.4 on the three instances.

A2-corridor

Make an ideal timetable. Since obtaining an optimal solution is out of reach for this instance, we use a time limit of two hours. The resulting timetable has, as mentioned earlier, a normalized objective value of 100. All other values are reported in relation to this value. The lower bound that is proven by CPLEX is 97% of the objective value. Hence the remaining gap is 3%.

Figure 4.4a displays a time-space diagram, showing the ideal timetable between Hdr and Ut. Hdr is in the most northern part of the network and Ut is halfway the corridor (see Figure 4.2a). Time is shown on the horizontal axis, between 0 and 60, i.e., one cycle period is displayed. Space is shown on the vertical axis, where several stations are mentioned. The lines on the right of the figures display the number of tracks that are present. In the diagram itself, each line corresponds to a train and displays at what time a train visits a given location. Figure 4.3 displays the routes of the five train lines in this instance.

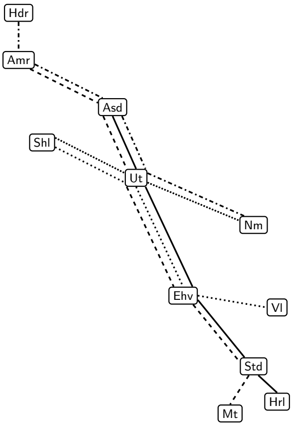
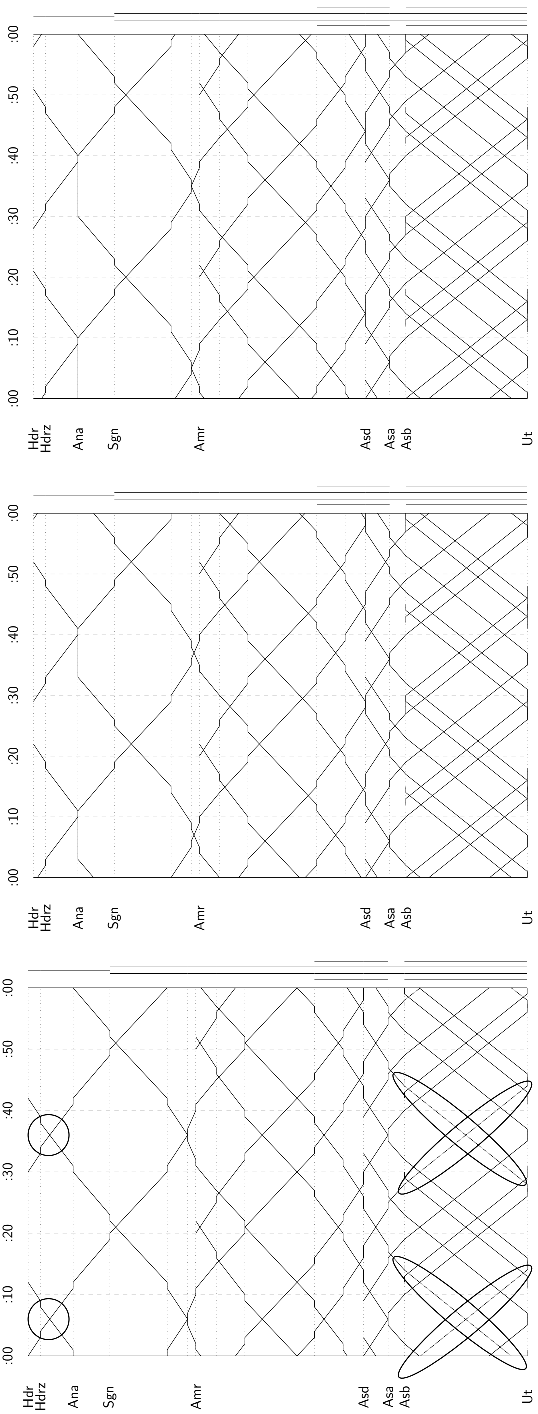


Figure 4.3: Overview of lines in A2-corridor.

Even though no regularity restrictions are added to the model, we see in Figure 4.4a that the trains are spread over time rather regularly in this network. This is due to the inclusion of the adaption time in the objective function. Adaption time is low, if departure times of routes for an OD-pair are equally spaced in time. However, the ideal timetable on the A2-corridor does not satisfy the headway restrictions. There are two conflicts, which are indicated by circles in Figure 4.4a. Between Ut and Asb, two trains are scheduled at the exact same time (one line is shown dashed). Between Hdr and Sgn, two trains are scheduled to cross each other in a single track area, where trains can only pass each other at stations.

Make a feasible timetable. In order to make a feasible timetable, we run LH with the nine different parameter sets specified in Section 4.5.2 for the profit, shift and stretch penalties and bounds. This leads to nine feasible timetables, all satisfying the headway restrictions. Note that although LH allows to cancel trains, for all nine chosen parameter sets, all trains are scheduled. The best found timetable in this step, BPL, has an evaluation value of 100.18, i.e., the evaluation value increased by 0.18% with respect to the ideal timetable. For this BPL timetable, the time space diagram is shown in Figure 4.4b. There it can clearly be seen that the conflicts are



(a) Ideal timetable

(b) BPL-timetable

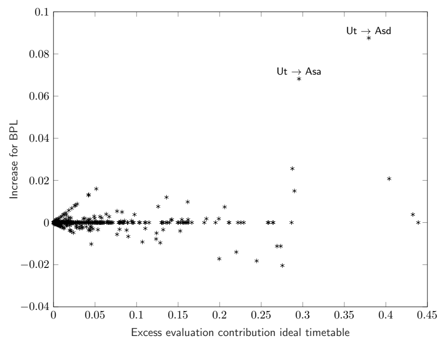
(c) BF-timetable

Figure 4.4: Time space diagrams for A2 network

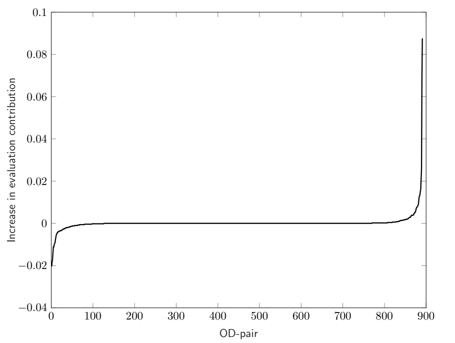
resolved. Trains that crossed on a single track area are now stretched in such a way that they pass each other at a station. Secondly, the trains that were scheduled at the same time are now moved away from each other.

Evaluate & update profit structure. In order to provide feedback, we follow the approach stated in Section 4.4.3.

Identify OD-pairs. The first step is to identify the OD-pairs for which the evaluation contribution worsened the most. In order to do so, and to investigate the differences between the ideal timetable π and the best found feasible timetable BPL, we compare the timetables with respect to the excess evaluation contribution of each individual OD-pair in Figure 4.5.



(a) Evaluation contribution comparison



(b) Evaluation contribution increase per OD-pair.

Figure 4.5: A2-corridor: ideal timetable vs BPL-timetable.

In Figure 4.5a, we see the difference in excess evaluation contribution for individual OD-pairs between timetables π and BPL. Each OD-pair is represented by a star, the stars are sorted from left to right according to the excess evaluation contribution of the corresponding OD-pair in the ideal timetable. As can be seen, for many OD-pairs the excess evaluation contribution is small in the ideal timetable, for only a few OD-pairs it is large. In this instance, the latter correspond to OD-pairs which have a high demand and a small irregularity in their departure pattern: recall that the evaluation contribution is weighted by the number of passengers, i.e., a large increase for only a few passengers can count less than a small increase for many passengers. The vertical coordinate indicates the difference in evaluation value between π and BPL. In particular, if a star lies above 0, the evaluation contribution of the OD-pair has increased after applying LH. But there are also some OD-pairs which have a lower evaluation contribution after applying LH, these can be found below 0. Two OD-pairs are labelled in the figure, these are the OD-pairs with the highest increase in perceived travel time that clearly stand out and on which we base the feedback. As can be seen, the OD-pair Ut-Asd has an excess evaluation contribution of 0.38 in the ideal timetable, and that evaluation contribution now increased to 0.47 in the BPL-timetable, due to a more irregular departure pattern at Ut.

In Figure 4.5b we see a different visualization of the differences between π and BPL with respect to the evaluation contribution. All 891 OD-pairs are shown on the horizontal axis sorted by their corresponding increase in evaluation contribution. As can be seen well in this figure, there are many OD-pairs for which the evaluation contribution hardly changed, only for a minority there are major changes. Hence, also in this figure it can be seen that the BPL timetable is very close to the ideal one, and only few OD-pairs were subject to an increase in perceived travel time.

Update profit structure. Two OD-pairs are identified as relevant: Ut-Asd and Ut-Asa. These two OD-pairs have the same set of travel options, as the trains from Ut to Asd first pass Asa (see Figure 4.2a). In fact, Asa is the only station where the trains from Ut to Asd stop. Passengers on these OD-pairs can choose from six different trains. Thus, to minimise adaption time upon departure in Ut, the headway times between consecutive trains would be 10 minutes, in that case the trains would be perfectly spread over time. In the ideal timetable we have headway times of 12 minutes (2 times) and 9 minutes (4 times), as is visible in Figure 4.4a. That means that in the ideal timetable the trains are not spread equally over time, and this causes the excess evaluation contribution of the relevant OD-pairs in the ideal timetable to be relatively large, in particular because these adaption times are weighted with the (high) passenger numbers. In the BPL-timetable, the departure pattern in Ut becomes even less regular. The headway times now are 15 minutes (twice), 9 minutes (twice) and 6 minutes (twice), see Figure 4.4b.

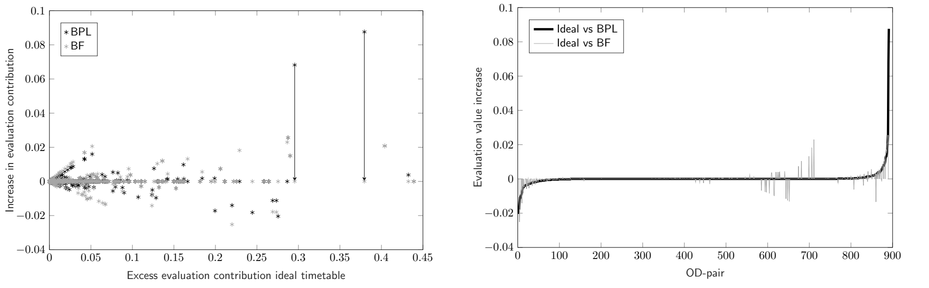
Based on these observations, we add intermediate shift penalties at Ut as described in Section 4.4.3, to improve the timetable for Ut-Asd and Ut-Asa. We start with the profit structure leading to the BPL timetable, and generate three new profit structures using values of 10, 20 and 30 as penalty values. Ut is intermediate station to 20 trains where these are added as intermediate shift penalties ρ^{is}_{Ut}. No trains start their journey in Ut so no regular shift penalties have to be updated. We run LH with the three new profit structures and evaluate the resulting timetables. Unfortunately, in this case, none of these timetables give an overall improvement.

Identify OD-pairs. In all of the three timetables obtained using the updated profit structures, we see that the OD-pairs Ut-Asd and Ut-Asa are not improved, in fact, for the OD-pairs in the other direction (Asd-Ut and Asa-Ut) the situation becomes worse compared to the BPL timetable. The regularity at Asd (and in line with that also at Asa) is lost. Therefore, we identify these two OD-pairs as the new OD-pairs to provide additional feedback.

Update profit structure. As described in Section 4.4.3 we modify the initial profit structure that was used to find the BPL timetable and add shift penalty values at Ut *and* Asd. Asd is intermediate station to 8 trains, and for 2 trains the start station. This leads to 12 new profit structures: 3 for adding the three different penalty values 10, 20, and 30 at Asd and 9 for the combinations of Ut and Asd. Note that a shift penalty at Asd can also improve the timetable at Ut, since these two stations are close to each other and there is only one stop in between.

Evaluation. After evaluating the 12 additional timetables, we find that the best evaluation value among the 12 created timetables now is 100.10, which is an improvement with respect to the BPL timetable. The time space diagram for this timetable is shown in Figure 4.4c. Note the improved regularity between Ut and Asd. The best timetable is found in the second feedback-iteration, and we refer to it as the *Best after feedback* (BF).

We now extend Figure 4.5 by adding grey stars, showing the evaluation contributions in the BF-timetable. Note that the OD-pairs Ut-Asa and Ut-Asd improve in the second iteration, as the grey stars representing these OD-pairs are now on the horizontal axis (the indicated arrows show the link between the two black and the corresponding grey stars). However, there are also some OD-pairs for which the perceived travel time increases in comparison to the BPL timetable. This is best seen in Figure 4.6b. Here, the same line as in Figure 4.5b is shown as a thick line. Now, the excess evaluation contributions for the OD-pairs in the timetable after feedback are added in grey. It is visible that some OD-pairs for who the evaluation contribution did not change in the BPL timetable are now changed: for some the perceived travel time decreases, but for others it increases.



(a) Evaluation contribution comparison (b) Evaluation contribution increases per OD-pair.

Figure 4.6: A2-corridor: Timetable comparisons after feedback.

Summary. A summary of the progress of our approach on the A2-corridor is given in Table 4.1. The table displays the evaluation values for the best timetables found in the steps of the algorithm. The last row of the table shows the lower bound that is found by CPLEX when solving the integer programming model for finding an ideal timetable, which is, of course, also a lower bound on the objective value of a feasible timetable.

Timetable	Evaluation value
Ideal	100
Best Pure Lagrangian	100.18
Feedback step 1 (FB-1)	100.23
Feedback step 2 (FB-2)	100.10
Lower bound	97.00

Table 4.1: Evaluation values for A2-corridor

A visual summary of our approach is displayed in Figure 4.7. Here, the evaluation values of all computed timetables are displayed. The horizontal axis displays the step in the algorithm. The vertical axis shows the evaluation value of the timetable. First, the evaluation value of the ideal timetable is shown at the bottom left. Then, the black lines and dots link this evaluation value to the evaluation values of the nine timetables computed by LH. Next, the grey lines and dots display the values of the timetables computed during the feedback process. The evaluation values of the three timetables computed in the first and second feedback step (FB-1 and FB-2) are shown, with lines linking these evaluation values to the evaluation value of the

BPL timetable. It is clearly visible that the evaluation value of the BF timetable is lower than that of the BPL timetable.

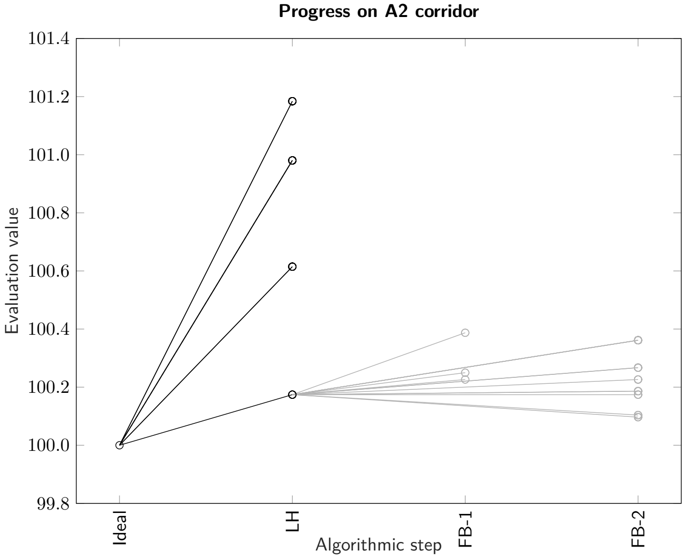


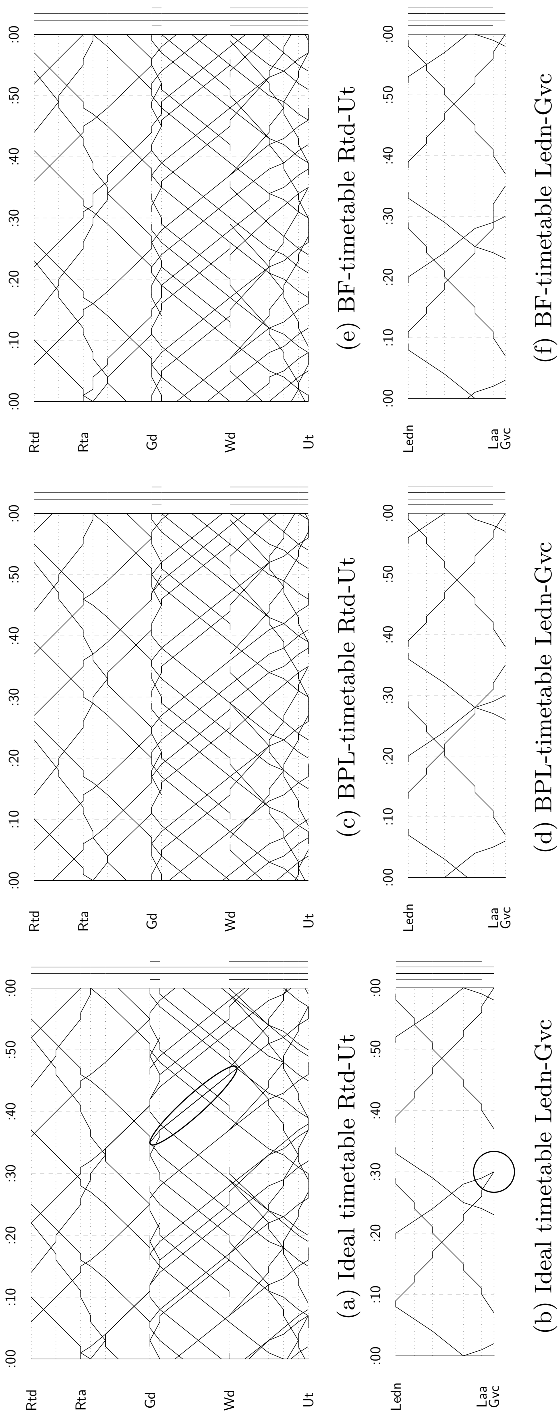
Figure 4.7: Overview of the progress in A2-corridor

Rotterdam-Groningen instance

Make an ideal timetable. Since this instance is more challenging than the previous one, we set a time limit of 4 hours to solve the model for constructing the ideal timetable. The best solution that we find within the time limit has a normalized objective value 100. The lower bound that is proven by CPLEX is 92.69% of the objective value, hence the remaining gap is 7.31%.

The ideal timetable is shown in terms of time-space diagrams in Figure 4.8 for two corridors that play a role in the description of the results: Rotterdam (Rtd) to Utrecht (Ut) and Leiden Central (Ledn) to The Hague Central (Gvc). See Figure 4.2b for the location of the stations.

The ideal timetable does not satisfy the headway restrictions on the two mentioned corridors. For example, between stations Gd and Wd, two trains are scheduled at the exact same time on the same track and hence do not satisfy the headway restrictions (see the circled area in Figure 4.8a). Also on the other corridor violations of the headway restrictions occur. For example, two trains from the same direction arrive



(a) Ideal timetable Rtd-Ut

(b) Ideal timetable Ledn-Gvc

(c) BPL-timetable Rtd-Ut

(d) BPL-timetable Ledn-Gvc

(e) BF-timetable Rtd-Ut

(f) BF-timetable Ledn-Gvc

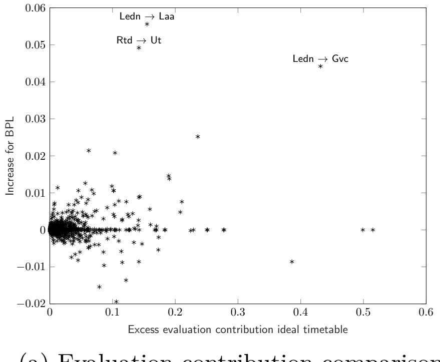
Figure 4.8: Time space diagrams for the Rotterdam-Groningen instance

in Gvc at the same time, while there is only one track available for them, so the headway restriction upon arrival of two trains is not satisfied (see Figure 4.8b).

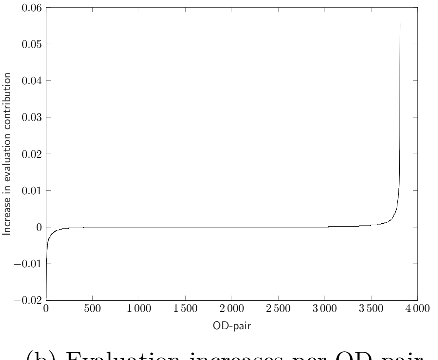
Make a feasible timetable. In order to find a timetable that is feasible with respect to current infrastructure, we run LH with the standard parameters. In seven out of the nine resulting timetables, all trains are scheduled. In two of them, two trains are cancelled. However, also in these timetables there are still travel options for all passengers. The best evaluated timetable has an evaluation value of 100.59 and has all trains scheduled. The time space diagrams for this BPL-timetable on the two aforementioned corridors are displayed in Figures 4.8c and 4.8d, which clearly show that the conflicts are resolved.

Evaluate & update profit structure.

Identify OD-pairs. In order to investigate the differences between the ideal timetable π and the BPL-timetable, we make the same plots as for the previous instance, showing the differences in evaluation contribution per OD-pair. Figure 4.9a displays each OD-pair as a star, where the excess evaluation contribution to the ideal timetable is the horizontal coordinate and the increase in evaluation contribution is the vertical coordinate. These increases in the evaluation contribution are summarised in Figure 4.9b, where all 3810 OD-pairs are shown on the horizontal axis, sorted by their corresponding increase in evaluation contribution. Note that for many OD-pairs the evaluation contribution does not change.



(a) Evaluation contribution comparison



(b) Evaluation increases per OD-pair.

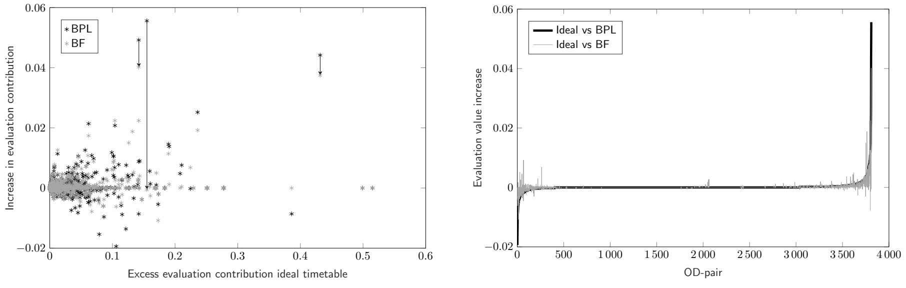
Figure 4.9: Rotterdam-Groningen: Ideal timetable vs BPL-timetable.

In the BPL-timetable, there are three OD-pairs that stand out, as is indicated in Figure 4.9a: Ledn-Laa and Ledn-Gvc are two OD-pairs that have very similar routes, they have the same origin and their destinations are close to each other, see also Figure 4.2b. The corresponding excess evaluation contributions increased by +0.056 and +0.044 respectively with respect to the ideal timetable. The third OD-pair for which the evaluation value increased significantly is Rtd-Ut, its excess evaluation contribution increased by +0.050 with respect to the ideal timetable. This OD-pair is located on a different part of the network than the other two OD-pairs.

Update profit structure. The origin stations for the aforementioned OD-pairs are Ledn and Rtd. At both of these stations, the regularity of departure times has worsened when shifting from the ideal to the BPL-timetable. Therefore, we make new profit structures, based on the one leading to the BPL-timetable, where we add the intermediate shift penalties of values 10, 20 and 30 to Ledn and Rtd. Ledn is intermediate station for 8 trains and no relevant trains start their journey in Ledn. Rtd is for no train an intermediate station, instead, 6 trains start their journey there so we update the regular shift penalty. This leads to 15 new profit structures: 3 with the penalty only at Ledn, 3 with the penalty only at Rtd, and 9 for all combinations.

Evaluation. We run LH on these new profit structures and evaluate the resulting timetables. We find the best timetable to have an evaluation value of 100.55. This timetable is referred to as the *best after feedback* (BF).

Like for the previous instance, we extend Figure 4.9 by plotting the evaluation contributions of the OD-pairs in the BF-timetable, in order to get insight into the differences with respect to the BPL-timetable. The result is displayed in Figure 4.10. Figure 4.10a shows for each OD-pair the increases in evaluation contribution with respect to the ideal timetable. The black stars correspond to the BPL-timetable, and the newly added grey stars correspond to the BF-timetable. To increase visibility, we connected the black and grey stars which correspond to the relevant OD-pairs. As can be seen in this figure, the evaluation contribution for the three OD-pairs is improved. The increase in evaluation contribution with respect to the ideal timetable of Ledn-Laa reduces to -0.001, i.e., the timetable for this OD-pair is better than the ideal timetable. The increase in evaluation contribution for Ledn-Gvc now reduces to $+0.037$ and Rtd-Ut reduces to $+0.040$.



(a) Evaluation value comparison (b) Evaluation increases per OD-pair.

Figure 4.10: Rotterdam-Groningen: Timetable comparisons after feedback.

To understand how the evaluation contribution of OD-pair Ledn-Laa is improved in the feedback step, we examine the time-space diagrams in Figures 4.8b–4.8f. In the ideal timetable for our instance, passengers can either take a local train directly from Ledn to Laa. The alternative is to take an Intercity train that does not stop at Laa, but travels to Gvc, where passengers can transfer to a local train back to Laa.

In the ideal timetable, the Intercity train from Ledn arrives in Gvc at :02, and passengers have a 5 minute transfer connection to a local train back to Laa. In the BPL-timetable though, the Intercity train comes in four minutes later and arrives in Gvc at :06, while the local train back to Laa is not shifted in time. A transfer time of one minute is too short and hence passengers have to wait half an hour, thus leading to a very high evaluation value contribution.

In the feedback, trains get an intermediate shift penalty at Ledn. This causes the Intercity train still to arrive in Gvc later than in the ideal timetable, but earlier than in the BPL-timetable: it now arrives at :03. Passengers can again make the connection to the local train and we are in a similar situation as in the ideal timetable.

This illustrates how the additional penalties lead to a different timetable and how feedback can be used to improve the timetable that is found.

Note that the 2019 NS line plan (NS, 2017) contains more trains between Ledn and Laa that we did not include in our instance. The observed effect is thus particular to our instance and would not be observed in the actual Dutch network.

Summary. Table 4.2 summarizes the results of our approach on the Rotterdam-Groningen instance. The evaluation values are displayed for each step in our algorithm.

Timetable	Evaluation value
Ideal	100
Best Pure Lagrangian (BPL)	100.59
Feedback step 1 (FB-1)	100.55
Lower bound	92.69

Table 4.2: Evaluation values for Rotterdam-Groningen instance

Also for this instance, we summarize the evaluation values found in the different steps of the algorithm in Figure 4.11. On the left, we see the evaluation value of the ideal timetable. Right of it, we see the evaluation values of the nine timetables found by the Lagrangian heuristic. Two lines are drawn with a dash-dotted line, these correspond to the timetables where some trains are cancelled. We observe that cancelling trains does not automatically lead to bad timetables, as long as all OD-pairs can still travel. This is because the cancelling of trains gives us more freedom to schedule other trains.

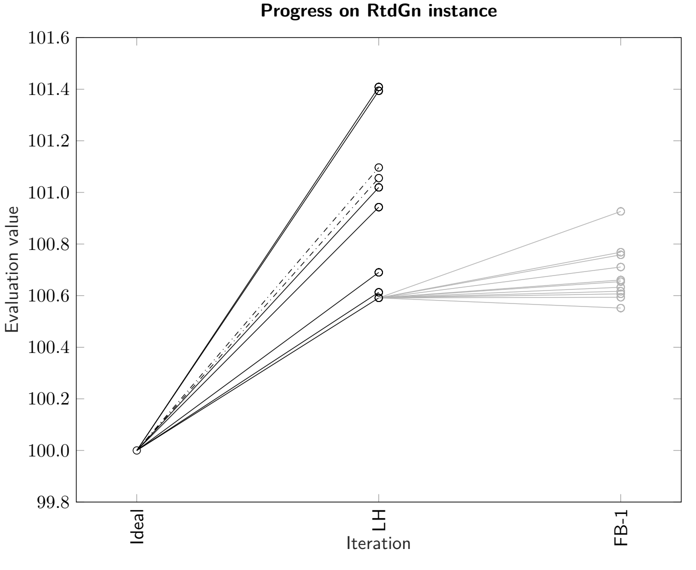


Figure 4.11: Overview of the progress in Rotterdam-Groningen instance

Extended A2-corridor

Make an ideal timetable. Also for this instance, we use a time limit of four hours to solve the model for constructing the ideal timetable. Normalizing the best timetable to an evaluation value of 100, the best lower bound is 93.00, i.e., the remaining gap is 7.0%.

Time-space diagrams displaying the ideal timetable are shown in Figure 4.12 for three corridors in the network: Arnhem (Ah) to Nijmegen (Nm), Amsterdam Central (Asd) to Schiphol (Shl) and Zaandam (Zd) to Utrecht (Ut). See Figure 4.2c for the location of these stations. In this dense network, there are numerous conflicts between trains that have to be resolved in the next step.

Make a feasible timetable. Using the same initial profit structures as in the earlier cases, in the first step of LH nine feasible timetables are computed. In three of these timetables, all trains are scheduled. In the other six, trains are cancelled in a way that causes some passengers not to have a travel option any longer. This is highly undesirable and highly penalized in the evaluation function. This leads to these six timetables having a bad evaluation value. Consequently, the feasible timetable with the best evaluation value is one in which all trains are scheduled. Its evaluation value is 101.51. The corresponding time-space diagrams are shown in Figure 4.12.

Evaluate & update profit structure.

Identify OD-pairs. As a first step, we identify the OD-pairs for which the shift from the ideal timetable to the BPL-timetable caused a high increase in the evaluation contribution. The changes in the evaluation contribution for each OD-pair are pictured in Figure 4.13, in a similar way as in the previous two cases. We observe that for many OD-pairs there are only few or little changes (see also Figure 4.13b). However, there are a few OD-pairs for which the evaluation contribution increases significantly as is visible in Figure 4.13a. The highest increase is visible for the the OD-pair Ah-Nm. Its evaluation contribution increases by +0.062. Besides Ah-Nm, Asd-Ut (+0.045), Asd-Zd (+0.039) and Asd-Shl (+0.038) account for the largest increases in the evaluation value.

And indeed, inspecting the timetable differences (see the supporting time-space diagrams in Figure 4.12), we see that departure and arrival patterns have changed at Asd, trains now sometimes depart in a less regular pattern. Also the pattern of departures at Ah is less regular in the BPL-timetable than in the ideal timetable (compare also Figures 4.12a and 4.12d).

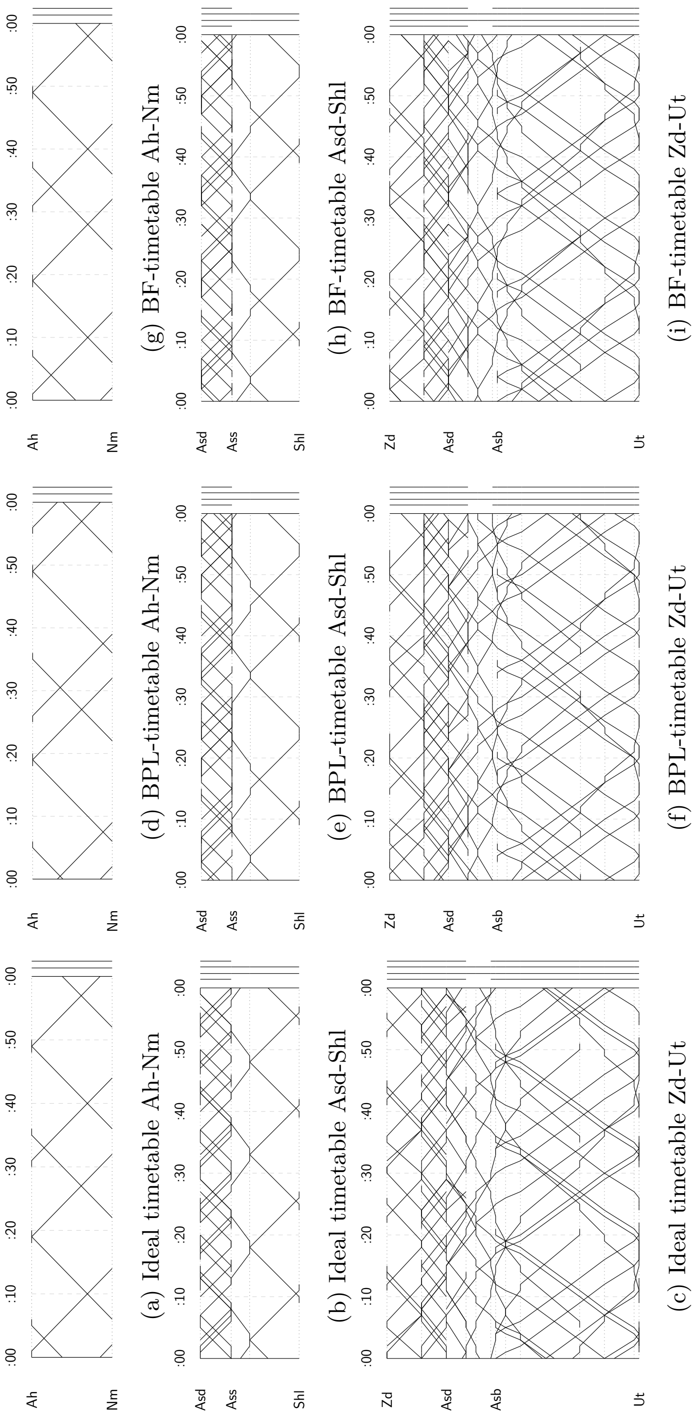
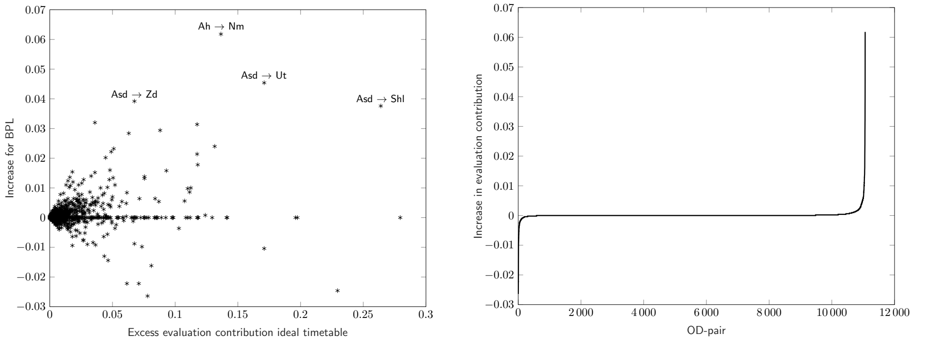


Figure 4.12: Time space diagrams for the extended A2-corridor instance



(a) Evaluation contribution comparison (b) Evaluation increases per OD-pair.

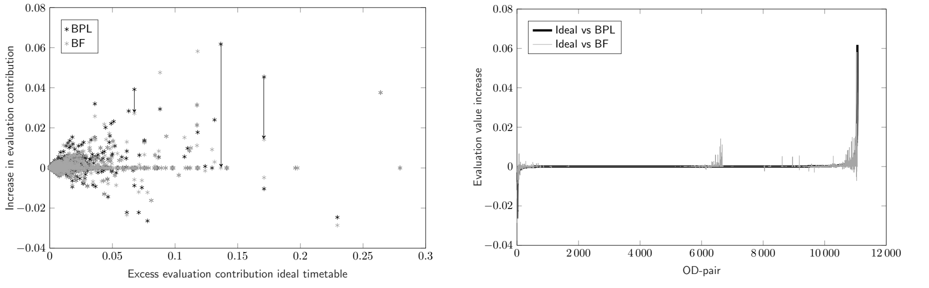
Figure 4.13: Extended A2-corridor: Ideal timetable vs BPL-timetable.

Update profit structure. As prescribed in our approach from Section 4.4.3, we update the profit structure for trains at the stations Ah and Asd with penalty 10, 20, or 30, leading two 15 new profit structures (3 for the shift penalties at Asd, 3 for the shift penalties at Ah, and 9 for the combinations).

Evaluation. We run LH with the new profit structures and obtain 15 new solutions. When evaluating these timetables, we find an improved timetable with an evaluation value of 101.28, i.e., a reduction of 0.23 with respect to the BPL-timetable. This new timetable is referred to as the BF-timetable.

Figure 4.14 displays the new evaluation contributions, both for the BPL-timetable and BF-timetable. The black stars correspond again to the increase in evaluation contribution when comparing the ideal timetable with the BPL-timetable. The grey stars show the same result when comparing the ideal timetable with the BF-timetable. The OD-pairs that had a high increase in evaluation contribution now improved significantly: the increase in evaluation contribution for Ah-Nm reduced from +0.062 to 0 (see also the time space diagram in Figure 4.12g). For Asd-Ut, the increase in evaluation contribution reduced from +0.045 to +0.014 and for Asd-Zd it reduced from +0.039 to +0.027. The increase for Asd-Shl remained the same. The improvements for the first three mentioned OD-pairs are shown by means of arrows in Figure 4.14a.

Although the overall evaluation value of the new timetable improved and the contributions of the aforementioned three OD-pairs improved as well, now other OD-pairs have a higher evaluation contribution, as already indicated in Figure 4.14b. An improvement for some OD-pairs can indeed imply a worsening for others. In the



(a) Evaluation value comparison (b) Evaluation value increases per OD-pair.

Figure 4.14: Extended A2-corridor: Timetable comparisons after feedback.

BF-timetable, there is an OD-pair (Hlm-Asd) which now has an excess contribution which is similar to that of Ah-Nm in the BPL-timetable.

Summary. Table 4.3 summarizes the main results of our approach on the extended A2-corridor instance in terms of evaluation values for each of the steps.

Timetable	Evaluation value
Ideal	100
Best Pure Lagrangian (BPL)	101.51
Feedback step 1 (FB-1)	101.28
Lower bound	93.00

Table 4.3: Evaluation values for extended A2-corridor instance

A visual summary of the computations is shown in Figure 4.15. For each computed timetable, the evaluation value is plotted. On the left the evaluation value of the ideal timetable is shown. Next, the results of the timetables after running LH are indicated by the black dots and lines. If in some timetable not all trains are scheduled, this is indicated by the dash-dotted line. As can be seen, many timetables have a bad evaluation value, they are not even visible in the figure. The reason for this is that in these timetables not all passengers have a travel option. In grey, the evaluation values of the timetables after feedback are shown. Again, we use dash-dotted lines for the timetables where not all trains are scheduled.

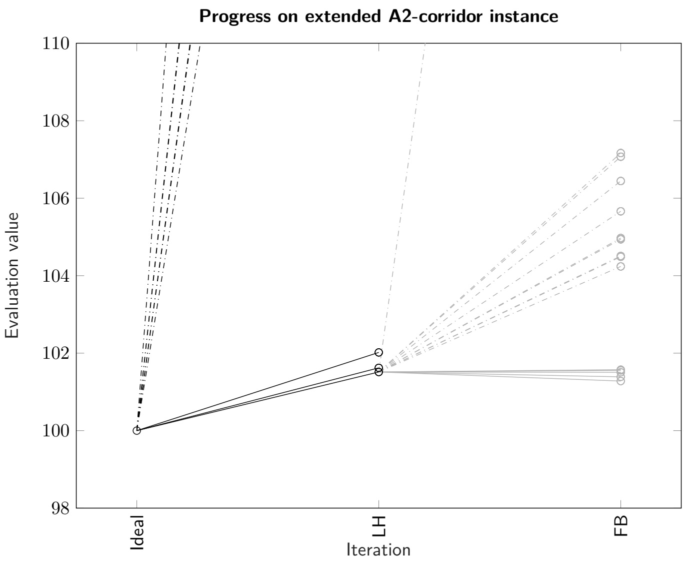


Figure 4.15: Overview of the progress in the extended A2-corridor instance

4.5.4 Comparison to a Benchmark Approach

In order to further evaluate our approach to solving POT, we compare it to the benchmark approach of modelling POT as a mixed-integer linear program. To make this comparison, we model POT as a mixed-integer linear program as described in Chapter 3, but additionally include infrastructure capacity constraints in the there-described model for strategic timetabling. We then solve it with the approach described in Chapter 3. Note that the extended mixed-integer programming approach from Chapter 3 does not allow to cancel trains, as does our model. However, since there are no trains cancelled in any of the best timetables we found in the three considered instances, a fair comparison of results is possible. The results are displayed in Table 4.4.

In this table, for each instance, the result of several approaches to find a feasible timetable are shown. The first result is the best timetable that is obtained after computing an ideal timetable, and then running LH. The table reports the evaluation value of the timetable in the third column. The fourth column shows the time it took to compute this timetable. Where the time is split up in two parts, the first number shows the time spent on computing the ideal timetable, the other number shows the time spent on LH and feedback. Note that the time for inspection of the timetable and adjustment of the profit structure in the feedback loop is not included,

Instance	Approach	Evaluation value	Time (hours)
A2	Ideal + LH	100.18	$2 + 0.03$
	Ideal + LH + FB	100.10	$2 + 0.11$
	POT		
	- After 2.11 hours	105.80	2.11
	- After 8 hours	104.88	8
	Lower bound CPLEX	97.09	
Rotterdam Groningen	Ideal + LH	100.59	$4 + 0.06$
	Ideal + LH + FB	100.55	$4 + 0.18$
	POT		
	- After 4.18 hours	105.64	4.18
	- After 16 hours	103.69	16
	Lower bound CPLEX	92.72	
Extended A2	Ideal + LH	101.51	$4 + 0.14$
	Ideal + LH + FB	101.28	$4 + 0.49$
	POT		
	- After 4.49 hours	-	4.49
	- After 16 hours	-	16
	Lower bound CPLEX	93.00	

Table 4.4: Benchmark results

because this is a manual process. The second result that is shown for each instance is the evaluation value of the best timetable found after applying feedback. Third, the value of the best timetable after solving the integer programming formulation for POT including infrastructure capacity restrictions is shown. Next to the value that is obtained when reaching the time limit, we also show the evaluation value of the best timetable found in the time it took the iterative approach to compute the solutions listed. That means, for the A2-corridor instance, computing an ideal timetable, running LH and including feedback took 2.11 hours. In the same time, the full POT model found a solution of value 105.80. Finally, we mention the lower bound as computed by CPLEX when solving the POT model until the time limit is reached. Note that the CPLEX lower bounds are stronger than those mentioned in Section 4.5.3 where we report lower bounds on the MILP model for finding the *ideal* timetable, because the POT model is more restrictive and therefore, combined with a longer computation time, a stronger lower bound is more likely.

We observe that our approach is able to find better solutions in less time, even when no feedback is included. In particular, for the extended A2-corridor instance, we were not able to find any feasible timetable within 16 hours using the MILP formulation for POT, while the approach of this paper generates a reasonably good one within a bit more than four hours.

4.6 Conclusion and Further Research

In this chapter, we proposed an approach to solve the tactical timetabling problem. Hereby we specifically focused on the quality of the timetable for the passengers. In order to find a feasible passenger-oriented timetable for challenging real world instances, for which the timetabling model itself already is challenging, we used variants of two existing approaches. These two approaches are combined into an algorithmic framework. First, an ideal timetable is computed, thereby neglecting infrastructure related restrictions. Next, through a Lagrangian heuristic, this timetable is modified to obtain a feasible timetable with respect to infrastructure. A feedback mechanism is used to improve the found solutions. We showed that for real-life instances, based on the network operated by Netherlands Railway, we can obtain satisfying results. Furthermore, we show that the provided feedback indeed leads to (overall) better timetables.

Interesting further research would include the further automatisation of the feedback procedure. Although this procedure is formalized in Section 4.4.3, it can still require manual inspection of the results in order to find a good feedback option. Furthermore, it would be interesting to investigate effects of including station capacity in our models. In addition, our current approach uses the assumption that passengers arrive uniformly distributed over the period. To include more detailed passenger information that may become available when entering the tactical or operational planning phase, different modelling approaches may be needed.

Chapter 5

Resolving Infeasibilities in Railway Timetabling Instances.*

5.1 Introduction

In Europe, many public transportation companies operate a cyclic timetable, which means that the timetable is repeated every time period, usually every hour. A mathematical model that is commonly used for finding such a cyclic (also called periodic) timetable is the Periodic Event Scheduling Problem (PESP) as introduced in Serafini and Ukovich (1989) and explained in Chapter 2. The task in PESP is to schedule a set of events V in such a way that a set of activities A (restrictions) are satisfied. In this chapter, we use the notion of a PESP-instance: an instance $\mathcal{I}$ contains events V, activities A and bounds ℓ_{ij}, u_{ij} for all $(i,j) \in A$. If for an instance $\mathcal{I}$ a feasible solution $\pi : V \mapsto \{0, 1, \ldots, T-1\}$ exists satisfying all activities, we call the instance $\mathcal{I}$ feasible. Else, we call it infeasible. The problem of infeasible instances is already recognized by Kroon and Peeters (2003).

Chapter 1 argued that there can be instances for which no feasible solution exists, especially since railway networks can be very crowded with trains. However, it can

*This chapter is a modified version of G.J. Polinder, L.G. Kroon, M.E. Schmidt, K.I. Aardal, M. Molinaro (2018): *Resolving infeasibilities in railway timetabling instances.*

also occur that too many requirements are formulated that the timetable has to satisfy, which results in the fact that no feasible solution exists. These restrictions are not necessary to ensure a safe operation, they are only added to improve the quality of the timetable. To show how a conflict can arise in a practical situation, we consider the following example.

Example 5.1. Consider the situation where two trains with a different speed use the same track between two stations. A mathematical model that describes this situation and schedules the departure of train t ($t \in \{1, 2\}$) at the first station (π_1^t), and the arrival (π_2^t) and departure (π_3^t) at the next station, is as follows:

$$\pi_2^1 - \pi_1^1 \in [6, 7] \tag{5.1a}$$
$$\pi_2^2 - \pi_1^2 \in [7, 8] \tag{5.1b}$$
$$\pi_3^t - \pi_2^t \in [1, 1] \qquad \forall\, t \in \{1, 2\} \tag{5.1c}$$
$$\pi_j^2 - \pi_j^1 \in [3, 57] \qquad \forall\, j \in \{1, 2, 3\} \tag{5.1d}$$
$$\pi_j^2 - \pi_j^1 \in [30, 30] \qquad \forall\, j \in \{1, 3\} \tag{5.1e}$$

The first train needs at least 6 and at most 7 minutes to travel between the stations (5.1a). The other train needs at least 7 minutes and at most 8 minutes (5.1b). Both trains dwell for exactly one minute after their arrival at the station (5.1c). The trains share the same track and hence are separated in time (the headway time) by 3 minutes upon departure and arrival (5.1d). Secondly, planners want the trains to be perfectly spread over time, thus enforcing a 30-minute interval between their departures at both stations (5.1e).
Due to the flexibility in the trip times, a feasible solution can be found by setting the trip times for both trains to 7 minutes. However, in many practical timetabling models, trip times are fixed, leading to infeasible instances as we see next.

The example stated above schedules both departure and arrival times. When travel times are assumed to be fixed, it is possible to merge pairs of trip and dwell activities and state PESP instances solely in departure events. When trip times are fixed, the width of the intervals in (5.1a) and (5.1b) is zero. Then, knowing the time of a departure event, the time for the next arrival event follows immediately. Fixed trip times have several advantages. First of all, fewer event times have to be determined, thus reducing the complexity of the problem. Secondly, when modelling single track safety, fixed trip times are needed to model this correctly using only PESP-constraints. Furthermore, fixed trip time reduce the solution space and can thus decrease running time of solution methods. However, assuming fixed travel times implicitly leaves less flexibility in the constraint set, since there is no freedom in choosing a travel time. Liebchen and Möhring (2007) describe under which circumstances this flexibility loss can be overcome, but this has a cost also.

Now assume that trip times are fixed in Example 5.1: the first train uses 6 minutes to travel and the other train uses 7 minutes. The instance can now be stated solely in departure events as is done in (5.2). A graph representation of the instance (an event-activity network) is shown in Figure 5.1. Note that events π_2^1 and π_2^2 now have vanished, and the instance contains three activities less.

$$\pi_3^1 - \pi_1^1 \in [7,7] \tag{5.2a}$$
$$\pi_3^2 - \pi_1^2 \in [8,8] \tag{5.2b}$$
$$\pi_j^2 - \pi_j^1 \in [3,57] \qquad \forall\, j \in \{1,3\} \tag{5.2c}$$
$$\pi_j^2 - \pi_j^1 \in [30,30] \qquad \forall\, j \in \{1,3\} \tag{5.2d}$$

Clearly, no feasible timetable exists for this instance, as activities (5.2a), (5.2b) and (5.2d) prevent the existence of a solution. We say these activities together form a *conflict*.

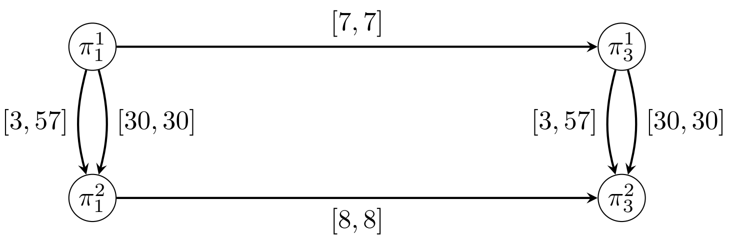


Figure 5.1: Infeasible PESP instance

Problem Statement

In this chapter, we deal with PESP instances for railway timetabling for which no feasible timetable exists. That is, the instances contain sets of activities that together form a conflict, thus prohibiting the existence of a feasible solution. The focus of this chapter is not to *solve* PESP but to *resolve the infeasibilities* in infeasible PESP instances. This 'resolving' is done by changing bounds of the PESP-activities as little as possible where this is needed.

Formally, we define the problem which we solve in this chapter as follows.

Definition 5.1 (Infeas-PESP). *Let $\mathcal{I}$ be an infeasible PESP instance, with event-activity network $G = (V, A)$. For each $a \in A$, let ℓ_a (u_a) denote lower (upper) bounds of the activity and τ_a^ℓ (τ_a^u) the given maximum allowed deviations of these bounds. The task is to find new activity bounds $\ell'_a = \ell_a - s_a^\ell$ ($u'_a = u_a + s_a^u$), where $s_a^\ell \in [0, \tau_a^\ell]$ ($s_a^u \in [0, \tau_a^u]$), such that, with respect to these new bounds, a feasible timetable $\pi : V \to \{0, 1, \ldots, T-1\}$ exists, or to decide that no such change exists.*

Note that it might not always be possible to find changes to the activities satisfying the given bounds τ. In that case, no resolution of a conflict is possible. If a resolution of a conflict exists, we aim at finding one that violates the original activities as little as possible. Since activities model different real-life constraints, like drive and dwell times, transfer times, or safety distances, the violation of one activity may be more undesirable than that of another activity. For this reason, we introduce weights on the activity violations, which we try to minimize heuristically.

The remainder of this chapter is structured as follows. In Section 5.2, we review the literature on PESP and conflict solving and state our contribution. In Section 5.3, we describe the above outlined procedure in more detail. First, the notion of a *conflict* is formally defined in Section 5.3.1. Next, several models are formulated to resolve a conflict in Section 5.3.2. Also several methods to add more relevant activities to conflicts are shown. The iterative algorithm to subsequently resolve conflicts is proposed in Section 5.3.3. Computational results are presented in Section 5.4. We end with a conclusion and discussion in Section 5.5.

5.2 Literature Review and Contribution

As mentioned in Section 5.1, a commonly used model for finding a feasible periodic railway timetable is the Periodic Event Scheduling Problem (Serafini and Ukovich, 1989). This model is stated and explained in Chapter 2, including an overview of what can be included and how it can be solved. In the review of this chapter, we focus on conflicts in PESP instances.

5.2.1 PESP Conflicts

As described in Section 2.2, there exist various techniques to solve PESP instances. Many of the techniques find good solutions for large networks within a reasonable amount of time. However, many of these as well rely on the assumption that a feasible solution always exists. That this is not always true for real life PESP instances has already been noted in literature. Kroon and Peeters (2003) mention that this problem can be caused by assuming fixed trip times which is an assumption that is often made in railway timetabling (cf. Kroon and Peeters, 2003; Kümmling et al., 2015a). In order to incorporate more flexibility into the used mathematical models, the authors state under which necessary and sufficient conditions trip times can be allowed to vary. This provides slightly more flexibility in the planning processes, but in real-life busy railway networks, conflicts still arise.

A powerful technique to solve PESP instances which has been used for about three decades is constraint programming (Schrijver and Steenbeek, 1993). Here, some event in the network is fixed to a specific time and the implications this has for other event times are propagated throughout the network. By using smart rules and search strategies, a good solution can often be found rapidly. This technique has been in use at Netherlands Railways for many years. It has the additional property that, if no solution exists, it provides a set of constraints that together can never be satisfied. These sets of constraints can be used to resolve the infeasibility. However, this technique is not very suitable to resolve conflicts. A tool is available that determines the minimum slack needed per constraint bound (Odijk et al., 2003), however, this does not take domain specific knowledge into account and the proposed relaxations can actually not be allowed in practice.

Also more recently more attention has been paid to infeasible PESP instances. In the TAKT program (Deutschland-TAKT, 2019), timetables are generated for the German railway network. In these large instances, often infeasibilities are found that have to be repaired (Kümmling et al., 2015a). Powerful solvers that are used to compute timetables for the German railway network are based on a Satisfiability formulation (Großmann et al., 2012). In such a formulation, an infeasibility (a conflict) is a set of clauses for which no solution exists. According to Kümmling et al. (2015b), powerful solvers exist to extract such a set of clauses, which then are resolved. In order to do this, a binary search heuristic is combined with a MaxSAT approach. More details on this method are provided by Großmann et al. (2015).

The most intuitive way to resolve infeasibilities is to relax the bounds of the activities involved. The method by Großmann et al. (2015) finds changes to upper bounds of activities, and in some cases lower bounds as well, in order to determine if a feasible timetable exists. The authors use a function which assigns a weight to each activity, indicating how expensive it is to change this activity. The weight is set to infinity if an activity cannot be changed. However, in the paper, no specific weight function is provided. The authors, however, claim that in railway instances typically only dwell time activities and connection activities can be relaxed. This is due to the fact that PESP does not allow for dependencies between activity (see also Kümmling et al. (2015b)), which is the case when trip times are not assumed to be fixed. The method is tested on several instances, among which the largest one contains all the high speed lines and some of the most important regional train lines in Germany.

5.2.2 Our Contribution

In this chapter, we propose a heuristic methodology to resolve infeasible PESP instances by altering bounds of PESP activities. Our method is applicable both to PESP instances with variable trip times, and to PESP instances with fixed trip times. In the former case, the altering of the bounds corresponds to a widening of the interval which bounds the time difference between two events. For PESP with fixed trip times, we relax activities in the above-described way, except the trip time activities. For these, we instead consider a change of the trip time itself, i.e., we *change* the bounds instead of *relaxing* them. This is a novelty with respect to existing approach that resolve infeasibilities. To take implicit interdependencies in the activity set into account, we introduce relations between the PESP activities.

We propose a mixed-integer program (MIP) that is able to alter PESP activities in both ways described above, and can hence solve conflicts both in PESP instances with variable and with fixed trip times. While in theory, our MIP model could be used to resolve all conflicts in the PESP instance at the same time, infeasible PESP instances encountered in (timetabling) practice are too large to be resolved in one go. To be able to resolve large PESP instances, we propose a methodology that iteratively detects minimal conflicts (that is, a minimal set of activities that cannot be satisfied simultaneously), enriches them in a heuristic way, and solves them using the MIP. This procedure is repeated until all conflicts are resolved.

In our computational experiments, we systematically analyse and compare which methods of enriching a conflict work well and demonstrate that this is a good strategy to obtain better solutions. We evaluate our approach on timetabling instances from Netherlands Railways (NS), which has one of the most intensively used railway networks in the world, with over 75.000 activities in the PESP model. Our method is able to resolve conflicts, even in such a large and highly utilized network, within minutes.

We ran our computational experiments based on weights defined after communication with planners from NS. However, the short computation times of our method would even allow to use our method in an interactive way: Planners could set weights based on a first judgement of the importance of activities, and then adjust them based on the result found, to guide the solution into the desired direction.

5.3 Conflict Resolving

PESP instances arising in practice can be infeasible. This means they contain a set of activities that cannot be satisfied simultaneously. The aim is to make such instances feasible and in this section we describe our approach to achieve this goal (see Section 5.1 for a formal definition of the problem).

We propose an iterative procedure, as follows:

- Identify a local conflict (an exact definition is given in Section 5.3.1).
- Carefully select additional activities and events to be added to this local conflict, in order to find a better resolution of this conflict.
- Resolve the conflict using a MIP that minimizes a weighted sum of the changes in the network activities.
- Modify the activities based on the solution provided by the MIP and search for a new conflict (go back to the first step)
- As soon as we do not encounter a conflict anymore, we try to further optimize the changes that are made.

This procedure is detailed in the following sections.

5.3.1 Conflicts in PESP

In order to be unambiguous in the terms we are using, we give a definition of what we mean by the term 'conflict'.

Definition 5.2 (Conflict)**.** *A* conflict *is a set of events and PESP activities such that no timetable exists for these events satisfying all the activities.*

In line with the definition of a conflict, we have the following definition of a minimal conflict:

Definition 5.3 (Minimal conflict)**.** *A* minimal conflict *is a conflict that has the additional property that no single activity can be removed, while still satisfying the definition of a conflict.*

Note that minimal conflict does not mean it is of minimum size, i.e., with the least number of events. In fact, minimal conflicts can arbitrarily differ in size, as we show next.

In Figure 5.2 a conflict graph is shown with events $V = \{1, \ldots, 5\}$. For each $i \in V \setminus \{1\}$, we have an activity joining this event to the previous event $i - 1 \in V$. Furthermore, we have the activities $\pi_5 - \pi_1 \in [11, 11]_T$ and $\pi_5 - \pi_1 \in [12, 12]_T$. Clearly, the latter two activities form a conflict, which we denote by C_1. However, the activity $\pi_5 - \pi_1 \in [12, 12]$ together with the set of activities $\pi_i - \pi_{i-1} \in [\ell_{i-1,i}, u_{i-1,i}]$ for all $i \in V \setminus \{1\}$ form a conflict, too. We denote this conflict by C_2. Both conflicts are minimal: removing any constraint from the conflicts would resolve these conflicts. However, the number of activities in each conflict is different.

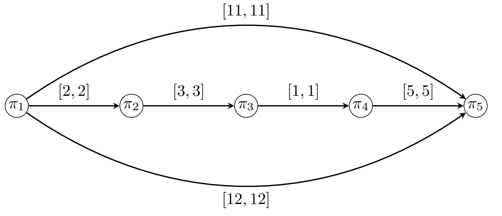


Figure 5.2: Minimal conflicts

By definition, any infeasible PESP instance contains at least one conflict. There exist different methods on how to identify conflicts, see, e.g., Kümmling et al. (2015b). In this chapter, we describe a methodology to make the PESP instance feasible by resolving the conflicts through a change in activity bounds, as defined in Definition 5.1.

When modelling railway timetabling instances as PESP problems, PESP activities describe different types of real-life railway timetabling constraints (see Chapter 2). One of these types are market requirements, which refer to *transfer activities*, stating

the arrivals and departures of different trains at the station need to be matched to allow for convenient transfers, and *synchronization activities*, which lead to an equal spacing in time of trains of the same line. That this can be suboptimal from a passenger perspective is motivated in Chapter 3.
By how much one is allowed to deviate from these activities and how much a deviation is penalized depends on the instance, and should in practice be decided based on discussion with the train operator.

5.3.2 Resolving a Single Conflict

In order to find a model to resolve conflicts in PESP, note that each PESP activity states that the difference in time between pairs of events should be within a given periodic interval. In case of a conflict, these intervals are too small, thus preventing the existence of a solution to PESP. Therefore, the bounds must be changed in order to obtain a feasible solution. The goal of this section is the following: given a specific conflict in the network G (represented by a subgraph G_C of G), find a change of activity bounds, such that the conflict is removed. More specifically, we are given an instance $\mathcal{I} = (G, w, \tau)$ being a triple of:

1. a PESP instance represented by its event activity network $G = (V, A)$;
2. a weight vector $w \in \mathbb{R}^{2|A|}$: for each bound of each activity a weight is given;
3. a vector $\tau \in \mathbb{R}^{2|A|}$ denoting by how much each bound of each activity can be changed;

Among all possible activity bound changes, we aim to find the one that minimizes a weighted sum of the deviations from the original activity set (to be defined more specifically in the next paragraph).

In the following paragraphs, we present our approach for solving conflicts in PESP. We do this based on the integer-programming formulation of PESP (see Section 2.1.1), since this is the most intuitive MIP formulation for PESP. However, the cycle periodicity formulation (CPF, Section 2.1.2) can be extended analogously, and we use an extension of this formulation for our experiments in Section 5.4 to speed-up computation times. Details can be found in Appendix 5.A.

The methods we describe in this section can be applied to conflicts of any size. In particular, by choosing $G_C := G$ we could consider the whole network as *one* conflict to be resolved. However, there is a trade-off between computation time for solving individual conflicts: the smaller the conflict, the shorter the time to solve it, and the more activities contained in each conflict, the less conflicts we have to solve. In particular, in realistic instances as the ones we consider in our experiments, G is too big to be resolved at once, which is why we follow the proposed iterative heuristic of detecting and resolving conflicts one by one. The trade-off between time needed to resolve one conflict and total number of conflicts to be resolved is further discussed later in this section, as well as several ways of enriching conflicts. The described trade-off is illustrated in our experiments in Section 5.4.

Basic model

A conflict can be represented by a conflict graph $G_C = (N_C, A_C) \subseteq G$ where N_C and A_C are the events and activities in the conflict. Let h_a and t_a be the 'head' and 'tail' of activity $a \in A$ respectively, i.e. for $a = (i, j) \in A$ we have $h_a = j$ and $t_a = i$. In order to resolve a conflict, activity-bounds must be changed. To be able to decrease lower bounds and to increase upper bounds, we introduce two new sets of (slack) variables: $s_a^\ell \in [0, \tau_a^\ell]$ indicates by how much the lower bound is decreased for an activity a, and $s_a^u \in [0, \tau_a^u]$ gives by how much its upper bound is increased. Then, for a general graph $G = (V, A)$, and thus specifically for the conflict graph G_C, we obtain the model PESP-IP-Ext:

$$
\begin{aligned}
\min \quad & \sum_{a \in A} w_a^\ell s_a^\ell + w_a^u s_a^u && && (5.3a)\\
\text{s.t.} \quad & \ell_a - s_a^\ell \leq \pi_{h_a} - \pi_{t_a} + T p_a \leq u_a + s_a^u, && \forall\, a \in A && (5.3b)\\
& s_a^\ell \in [0, \tau_a^\ell],\ s_a^u \in [0, \tau_a^u]. && \forall\, a \in A && (5.3c)\\
& \pi_i \in \{0, 1, \ldots, T-1\} && \forall\, i \in V && (5.3d)\\
& p_a \in \mathbb{Z}_{\geq 0} && \forall\, a \in A && (5.3e)
\end{aligned}
$$

where w_a^ℓ and w_a^u are the weights corresponding to the lower and upper bound of activity a respectively. The choice of weights in such a model depends heavily on the specific application and what kind of solutions are sought (Großmann et al., 2015). An equivalent formulation based on the cycle periodicity formulation for PESP can be found in Appendix 5.A.

Fixed trip times

We now discuss an extension of the described model which includes adjustments to bounds related to fixed trip times.

One advantage of assuming fixed trip times is that the problem size is reduced, only half of the event times need to be determined. Furthermore, in order to model safety constraints on single tracks, a known and fixed trip time is necessary (see Appendix 2.A).

Using fixed trip times implies that trip-dwell and trip time activities are stated between two departure events and incorporates both a trip period and a dwell period at a station. This dwell time interval is denoted by $[\underline{d}, \overline{d}]$. If the trip time for this part of the network equals r, the trip-dwell activity is of the form

$$r + \underline{d} \leq \pi_{h_a} - \pi_{t_a} + Tp_a \leq r + \overline{d}. \tag{5.4}$$

In the case that dwell time is fixed (i.e. $\underline{d} = \overline{d}$), or that a train does not stop at a station, it holds that $\ell_a = u_a = r + \underline{d}$ for this activity.

When adjusting activities involving fixed trip times, we do not relax activity bounds as in the previous paragraph, but allow the trip time r itself to change. That means, when the trip time is changed by δ, constraint (5.4) is rewritten to

$$r + \delta + \underline{d} \leq \pi_{h_a} - \pi_{t_a} + Tp_a \leq r + \delta + \overline{d}, \tag{5.5}$$

rather than widening the span of the constraint. Trip time changes can imply changes to bounds of safety activities, as is illustrated in the following example.

Example 5.2. Consider the headway time between trains 1 and 2 entering a station. For a headway time h, the constraint modelling this requirement reads

$$h \leq \pi_{a_2} - \pi_{a_1} + Tp \leq T - h, \tag{5.6}$$

where a_i denotes the arrival event of train i. However, since we state activities in departure events when assuming fixed trip times we use the fact that $\pi_{a_i} = \pi_{d_i} + r_i$ to reformulate (5.6) as

$$h \leq \pi_{d_2} + r_2 - \pi_{d_1} - r_1 + Tp \leq T - h, \tag{5.7}$$

or equivalently

$$h + r_1 - r_2 \leq \pi_{d_2} - \pi_{d_1} + Tp \leq T - h + r_1 - r_2. \tag{5.8}$$

When r_1 or r_2 changes, this safety activity should be changed accordingly.

In order to model these dependencies among activities, we introduce variables c_a^ℓ and c_a^u to denote the total change made to an activity $a \in A$. Changes can be made to an activity because either the activity *itself* is relaxed, for example by relaxing market requirements, or due to trip time changes in related activities. So the introduced change-variables consist of two parts: a slack component $s_a = (s_a^\ell, s_a^u) \in [0, \tau_a^\ell] \times [0, \tau_a^u]$, as was introduced in the previous model, accounting for changes in its own activity bound, and an implied change component c_a^{imp}, accounting for changes to this activity based on changes in other activities. This leads to the definition:

$$\begin{cases} c_a^\ell = s_a^\ell + c_a^{\ell,imp} \\ \\ c_a^u = s_a^u + c_a^{u,imp} \end{cases} \qquad \forall\, a \in A. \tag{5.9}$$

Here, the implied changes can be defined as

$$\begin{cases} c_a^{\ell,imp} = \left(\lambda_a^\ell\right)^\mathsf{T} s \\ \\ c_a^{u,imp} = \left(\lambda_a^u\right)^\mathsf{T} s \end{cases} \qquad \text{with } \lambda_a^\ell, \lambda_a^u \in \{0, \pm 1\}^{2|A|} \qquad \forall\, a \in A, \tag{5.10}$$

where $s \in \mathbb{R}^{2|A|}$ is the vector of all slack variables for each activity and each bound. Here, λ is a vector containing the coefficients how the bounds of this activity depend on changes in other activities, like the $+r1 - r_2$ in (5.8). T denotes that the vector is transposed.

Summarizing the constraints stated above leads to the following mixed integer programming model PESP-IP-Dep:

$$\min \quad \sum_{a \in A} w_a^\ell s_a^\ell + w_a^u s_a^u \tag{5.11a}$$

$$\text{s.t.} \quad \ell_a - c_a^\ell \leq \pi_{h_a} - \pi_{t_a} + T p_a \leq u_a + c_a^u, \qquad \forall\, a \in A \tag{5.11b}$$

$$c_a^\ell = s_a^\ell + c_a^{\ell,imp}, \quad c_a^u = s_a^u + c_a^{u,imp} \qquad \forall\, a \in A, \tag{5.11c}$$

$$c_a^{\ell,imp} = \left(\lambda_a^\ell\right)^\mathsf{T} s, \quad c_a^{u,imp} = \left(\lambda_a^u\right)^\mathsf{T} s \qquad \forall\, a \in A \tag{5.11d}$$

$$s_a^\ell \in [0, \tau_a^\ell], \quad s_a^u \in [0, \tau_a^u] \qquad \forall\, a \in A \tag{5.11e}$$

$$\pi_i \in \{0, 1, \ldots, T-1\} \qquad \forall\, i \in V \tag{5.11f}$$

$$p_a \in \mathbb{Z}_{\geq 0} \qquad \forall\, a \in A. \tag{5.11g}$$

The model presented above is an extension of PESP-IP-Ext in the sense that it can deal with changes in (fixed) trip times in the way that is shown in equation (5.5) in addition to relaxations. If we take $\lambda_a^\ell = \lambda_a^u = 0$ for each $a \in A$, we have $c_a^{\ell,imp} = c_a^{u,imp} = 0$, and we arrive at the PESP-IP-Ext model again.
An equivalent formulation based on the cycle periodicity formulation for PESP can be found in Appendix 5.A.

Redistribute time supplements. As described in the previous sections, the choice of the weights in the objective function is very important to find good solutions to PESP instances arising in timetabling problems. Another method that can help to guide the solution towards a specific direction is to redistribute time supplements which are added when designing a timetable. When designing a timetable, planners take into account that in a daily practice operations do not go as planned. Especially in train scheduling, this is very important, since the vehicles are bound to the tracks, and hence disturbances can easily influence the daily operations, leading to trains being delayed. In order to avoid the disruption of the daily operations as much as possible, schedulers add time supplements to the travel time on some parts of the train journey. That means, they plan that the train uses more time than actually needed, in general between 5% and 7% on the total journey. These additional minutes that are added are called *time supplements*. Within a journey, these supplements are added to parts of the journey by some rules. However, they could as well be added at slightly different places, without loosing much in terms of resilience. In this paragraph, we propose a method that finds solutions that redistribute these supplements to different trip activities, in order to obtain a feasible timetable. This does not change the total journey time of a train, but only redistributes the supplements.

Guiding the search for a solution towards these solutions is modelled by defining a variable $z_a = s_a^u - s_a^l$ for each trip activity $a \in A_{trip}$, accounting for the change in trip time. Here, $A_{trip} \subseteq A$ is the set of trip activities. Let $T_a \subseteq A_{trip}$ be the set of trip activities between the two main stations closest to the part that is involved by activity a. Then the total change C_a on this part of the train series is calculated as

$$C_a = \sum_{e \in T_a} z_e. \tag{5.12}$$

If nothing is changed in the trip time between these two main stations, we have $C_a = 0$. Also if the time supplements are redistributed between the stations, this value will be zero as well, since the changes to individual activities cancel out.

In order to find solutions that redistribute time supplements, we penalize solutions in the objective functions, that do not do this. In order to do so, note that T_a consists of a set of consecutive trip activities. Let $a^* \in T_a$ be the first of these consecutive activities. Next, we introduce a variable $t_a \geq 0$ for each $a \in A_{trip}$ and add

$$\gamma \cdot \sum_{a \in A_{trip}} t_a \tag{5.13}$$

to the objective function, where γ is an objective coefficient to be chosen. Furthermore, we add the constraints

$$-t_{a^*} \leq C_a \text{ and } C_a \leq t^{a^*} \qquad \forall\, a \in A_{trip}, \tag{5.14}$$

to the model, where $a^* \in T_a$ is as defined before. By doing this, we penalize the changes only once. By adding the above objective term and constraints, changes in trip times that are not compensated for in other trip time activities, get an additional penalty.

Resolving only minimal conflicts

In this section, we give an example to illustrate why identifying and resolving minimal conflicts within our iterative approach can lead to bad performance. Next we describe how to enrich conflicts to overcome this issue.

A minimal conflict in the sense of definition 5.3 does not necessarily grasp the real-life conflict in full detail. There might be more activities that are relevant to this conflict, but do not appear in the minimal conflict. We now slightly extend Example 5.1 with two additional activities which are often used in practice, to demonstrate that activities that are not included in the minimal conflict, can be relevant for making the PESP instance feasible.

Example 5.3. The reduced model in (5.2) contains a conflict, formed by the activities (5.2a), (5.2b) and (5.2d). These activities form a cycle in the constraint graph as well, see Figure 5.1. Considering this cycle $\mathcal{C}$, the bounds on the multiples of T that have to be in this cycle ($q_\mathcal{C}$) can be calculated (see Section 2.1.2) as

$$a_\mathcal{C} = \left\lceil \frac{7 + 30 - 8 - 30}{60} \right\rceil = 0,$$

$$b_\mathcal{C} = \left\lfloor \frac{7 + 30 - 8 - 30}{60} \right\rfloor = -1.$$

Note that $a_{\mathcal{C}} > b_{\mathcal{C}}$, which shows that this cycle forms a conflict.

The following changes to the activity bounds would resolve the minimal conflict:

1. Change the trip time of the first train from 7 to 8 minutes.
2. Change the trip time of the second train from 8 to 7 minutes.
3. Reduce the synchronisation requirements to allow a little flexibility, i.e., set the bounds in (5.2d) from $[30, 30]$ to $[29, 31]$.

However, suppose that the trains not only are synchronized to depart 30 minutes apart from each other, but that their departure times are actually fixed to a specific time. That is, the restrictions $\pi_1^1 = 20$ and $\pi_1^2 = 50$ are added. Note that these restrictions can be added in a PESP-form as well, by introducing an additional event π_0, and the activities $\pi_1^1 - \pi_0 \in [20, 20]$ and $\pi_1^2 - \pi_0 \in [50, 50]$. Due to symmetry of solutions, we can always shift a solution in such a way that $\pi_0 = 0$.

These additional restrictions show that option 3 to resolve the conflict does not lead to a feasible model. Therefore, resolving a minimal conflict might not lead to an overall feasible solution and more activities must be taken into account to find a good conflict resolution and to better capture the real-life conflict.

Methods to enrich a conflict

We now propose several ways to add more activities to a conflicts. In the remainder of the section we call a subgraph of the total timetabling graph containing some conflict a *conflict graph*. If the problem represented by this graph is a minimal conflict, we call it a *minimal conflict graph*. When enriching a conflict, we always start with a minimal conflict.

The models described in the previous section aim at resolving a part of the full PESP network, representing the railway timetabling problem. In this section we describe various methods to enrich minimal conflicts. First of all, for all trip time and trip-dwell time activities that are involved in the conflict, we add all activities for which the bounds depend on these trip times.

An event in a PESP instance specifies a departure or arrival at some location in the railway network. A conflict, or more generally, a set of events, corresponds to some physical location in the railway network. This property is used in defining several of the methods listed below.

1. **Use n_1 previous conflicts**
   If hardly any activities are added to a conflict, it might imply that there still is a conflict around the same location in the network, but now represented by a different set of activities. Hence, it may happen that in a next step of our iterative approach we find a conflict that is related to the previously found conflict. Therefore, it can be beneficial to resolve these conflicts simultaneously in order to obtain a better conflict resolution. Even the conflict found more iterations back can be relevant for this conflict conflict. Therefore we define a parameter n_1. When solving a conflict, we check the previously found conflicts up to at most n_1 conflict back, to see if they share some events. If so, we add these events to the minimal conflict graph. More precisely, let V_i and V_{i-j} be the set of nodes in the current conflict and the conflict j iterations ago respectively. Then, start with $j = 1$ and as long as $j \leq n_1$, check if $V_i \cap V_{i-j} \neq \emptyset$. If this is true, set $V_i = V_i \cup V_{i-j}$ and increase j by 1, else we stop.

2. **Add neighbouring activities (depth n_2)**
   Example 5.3 demonstrated that it can be useful to add more events to a conflict. Therefore, in this method we take all events in the conflict graph and add all incoming and outgoing activities of the corresponding nodes to the conflict graph, including the corresponding events. After having performed this addition, it can be repeated several times. The number of times we perform this operation is called the *depth* and is denoted by n_2.

   A practical interpretation of adding neighbouring activities is that in this way more events of the same train or more events of trains using the same tracks are added.

3. **Add all interrelated activities**:
   One way to add more useful activities to a conflict, without including new events, is to add all activities for which both corresponding events are already involved in the conflict. This rule for enriching a conflict is called *add all interrelated activities*. This does not enlarge the number of events in the conflict, or in its geographical interpretation, it only adds more activities.

If the conflict shown in Example 5.1 were to be enriched by only adding interrelated activities, no new useful relations would be added. If the incoming and outgoing activities were added (already for $n_2 = 1$), the conflict would be fully captured and a good resolution would be proposed.

4. **Add single track**
   Many conflicts in the rail network arise when single track legs are involved. The best way to resolve conflicts here is to consider all trains sharing this part of the infrastructure. Otherwise, there is a high risk of not resolving the conflict in the right way, thus postponing the problem. If we *add single track*, we add all activities and events that involve trains on this single track leg of the infrastructure.

5. **Add n_3 trips**
   In some cases, it is useful to look at the locations where the trains travel that are involved in the conflict. For each train, we add, if possible, n_3 trips before and n_3 trips after the part that is involved in the conflict, if this is possible. This means the conflict is extended geographically for all involved trains. This method of adding additional trips is called *add n_3 trips*.

All of these methods can be used on their own to enrich minimal conflicts, or be combined. They are evaluated experimentally in Section 5.4.

5.3.3 An Iterative Algorithm to Resolve Infeasible PESP Instances

We now combine what is described in the previous subsections. We first describe the individual iterations of the algorithm. Next, we use this as a subroutine to describe the final algorithm to resolve all conflicts in a PESP instance.

Resolve a single conflict

In this section we state our algorithm to enrich, and resolve conflicts. A key observation for understanding the design of our algorithm is the observation that the computation time for a model like PESP-IP-Dep can increase rapidly as the model size increases, since PESP is NP-complete (Serafini and Ukovich, 1989). Hence we have to be careful not to increase computation time too much when adding additional activities, while still taking care of the solution quality. Therefore, if we cannot resolve the enriched conflict in a certain time limit, we resolve the minimal conflict graph instead, and then successively add activities. In doing so, we can use the objective value found for a subproblem as a lower bound for the larger problem, supported by the following proposition:

Proposition 5.1. *Suppose two graphs $G_1 \subseteq G_2$ are given, and the optimal objective values according to the corresponding MIP model PESP-IP-Dep are z_1^* and z_2^*. Then $z_1^* \leq z_2^*$.*

Proof. Since G_2 contains at least all activities of G_1, the model corresponding to G_1 is a relaxation of the model of G_2 and hence $z_1^* \leq z_2^*$. □

Using the objective value of a subproblem as a lower bound for the larger problem, can help the solver to determine optimality since a possibly good lower bound already is available. However, we can also provide the full solution instead of only the objective value. There is a good chance that this solution satisfies many constraints, and hence it can be used as a 'warm start' for the solver. Whether we supply this solution as a warm start or not is encoded in a Boolean parameter b_3.

We now describe how we encode the methods we use for enriching conflicts. Note that also the order in which the methods are executed influences the final conflict graph. For ease of notation, we encode a combination of the methods as follows: $n_1 n_2 b_1 b_2 n_3 b_3$, where n_i corresponds to an integer number and b_i to a boolean value ($i \in \{1, 2, 3\}$). All these correspond to the rules described before and are defined as follows:

n_1 This refers to the method *add previous conflicts*, with parameter n_1 (option 1).

n_2 This value denotes the neighbourhood depth for adding neighbouring activities (option 2).

b_1 If this is *true*, we add all interrelated activities (option 3).

b_2 If this is *true*, we add single track activities (option 4).

n_3 We apply the rule 'add n_3-trips', i.e., for each train in the conflict, we add n_3 trips before and after the trips involved in the conflict (option 5).

b_3 If this is *true*, we provide the solution to a subproblem as a 'warm start' for a larger problem.

When enriching a conflict, all these methods are executed in the order they appear in the encoding. As an example of an encoding, the code *00ff0f* refers to the method that always resolves a minimal conflict without adding further constraints.

Our algorithm is stated as Algorithm 5.1 on page 134. It is parametrized by the aforementioned encoding, which defines the *maximum level of additional activities.*

The subgraph corresponding to the 'maximum level of additional activities' is denoted by C^+, and its corresponding encoding is denoted by enc^*. Let $\mathcal{S}$ be a list, containing the subgraphs for C^+ that are generated by the encodings from the previous section in the following order (where the variable with the asterisk denote the value they have in encoding enc^*): For $n_2 \in \{0, \ldots, n_2^*\}$: for $b_2 \in \{false, b_2^*\}$: for $n_3 \in \{0, \ldots, n_3^*\}$: for $b_1 \in \{false, b_1^*\}$: generate the subgraph corresponding to the encoding $n_1^* n_2 b_1 b_2 n_3 b_3^*$ and name them $S_1, \ldots, S_\kappa$ in the order they are generated. So $S_i \in \mathcal{S}$ $(i = 1, \ldots, \kappa)$ and $\kappa = |\mathcal{S}|$. Furthermore, S_1 is the minimal conflict, and $S_\kappa = C^+$. The PESP-IP-Dep models corresponding to these subproblems can be solved consecutively. Each MIP model corresponding to S_i provides a lower bound for the model corresponding to S_{i+1}, if it is a subgraph, according to Proposition 5.1.

In this algorithm, we take as input a minimal conflict C. Next, we enrich this conflict (line 1). We then try to resolve the conflict in the resulting constraint graph by the CPF-Dep model (line 3). We allow t_1 minutes for this. The time limits in the algorithm are dynamic. That means the following: If the time limit is ρ minutes, we allow at least ρ minutes to solve the model. If in the last $\rho/4$ minutes of this allowed time improvements are found, either in the lower bound or the upper bound of the model, we continue optimization for another $\rho/4$ minutes, until no improvements are found any more. The rationale behind this is that in solving PESP, we can often find rapid improvements in the beginning, and then often find ourselves in one of the following situations: We can have found the best solution, and the lower bound (slowly) is updated to prove optimality; or we are 'stuck' for some time and both the lower and upper bounds do not change. We want to break away from being stuck, but also not break out too early if all we have to do is prove optimality.

If in these t_1 minutes the model is solved to optimality, we use the found solution as a solution to the conflict (line 5). If not, we start with the smallest subproblem, and solve it (line 11). The time limit t_2 is usually set high in order to ensure that an optimal solution to the minimal conflict is found. This solution is then used as a lower bound for resolving the next conflict graph for which we impose a time limit of t_3 minutes (line 13). If a time limit is exceeded or all subproblems are solved, the last best found solution is used to resolve the conflict (line 20).

Resolve a full PESP instance

Until now, we have seen how a single conflict can be resolved. Now we describe the final algorithm which, given a PESP instance, iteratively searches for conflicts,

Data: Conflict C, time limits t_1, t_2, t_3, weight vector $w \in \mathbb{R}^{2|A|}$, bound vector $\tau \in \mathbb{R}^{2|A|}$.

Result: One of the following: 1. Values for c_a^ℓ and c_a^u (for each $a \in A_C$) such that changing the activity bounds by these values leads to a feasible resolution of conflict C; 2. Proof that no resolution exists; 3. Time-limit exceeded;

Enrich conflict C, giving enriched conflict C^+;
Build conflict graph for C^+;
Solve PESP-IP-Dep model for C^+ in t_1 minutes (dynamic time limit);
if *optimal solution found to PESP-IP-Dep* **then**
  Process solution found to C^+ setting activity bounds as:
  $\ell_a \leftarrow \ell_a - c_a^\ell$
  $u_a \leftarrow u_a + c_a^u$;
else
  Find sub-problems $S_1, \ldots, S_\kappa$;
  Set $\mathcal{S} \leftarrow S_1$;
  Solve PESP-IP-Dep model for $\mathcal{S}$ in at most t_2 minutes (dynamic time limit);
  for $i = 2, \ldots, \kappa$ **do**
    Solve S_i in t_3 minutes (dynamic time limit). If b_3 is set to *true*, use solution to S_{i-j} as a warm start and lower bound, with $j \in \mathbb{N}_{>0}$ as small as possible such that $S_{i-j} \subseteq S_i$;
    if *optimal solution found for* S_i **then**
      Set $\mathcal{S} \leftarrow S_i$;
    else
      break;
    end
  end
  Process the solution found to $\mathcal{S}$ by changing activities, by setting:
  $\ell_a \leftarrow \ell_a - c_a^\ell$
  $u_a \leftarrow u_a + c_a^u$;
end

Algorithm 5.1: Resolving one conflict

resolves them by Algorithm 5.1 (or notifies us that no feasible solution exists) and, if it terminates, provides a conflict-free PESP-instance and a timetable.

Data: PESP instance $\mathcal{I}$, time limits t_1, t_2, t_3, TL, iteration limit IL, weight vector $w \in \mathbb{R}^{2|A|}$, bound vector $\tau \in \mathbb{R}^{2|A|}$.
Result: One of the following: 1. List of adapted PESP activities such that conflict is resolved together with a timetable; 2. Proof that no resolution exists; 3. Time-limit or iteration-limit exceeded;

1 Set $it\_count = 0$;
2 Search for a conflict C;
3 **while** *A conflict* C *is found and* $it\_count < IL$ *and elapsed time* $< TL$ **do**
4   Run Algorithm 5.1 on conflict C;
5   **if** C *cannot be resolved* **then**
6   Stop algorithm: it is not possible to modify PESP parameters within the prespecified bounds to obtain a feasible solution to PESP;
7   **else**
8   Search for a new conflict C in $\mathcal{I}$.;
9   **end**
10 **end**

Algorithm 5.2: Resolving a PESP instance

This algorithm finds a conflict in the PESP instance or certifies that no such conflict exists. Next, it resolves the conflict. Afterwards, the search for more conflicts is continued until no conflict exist any more. If there are no more conflicts, we find a timetable and post-optimize it. Note that if Algorithm 5.1 determines that a certain conflict cannot be resolved, this means that there is no feasible solution to PESP, even with the allowed modifications of the activity bounds.

Due to the time limits set in line 3, Algorithm 5.2 will terminate after a pre-specified time, even if the conflicts in the network are not resolved up to that point in time and no unresolved conflict is found.

Note that for fixed values of the modulo parameters p, PESP-IP-Ext is totally unimodular. Hence when all activity bounds ℓ_a and u_a and all entries of τ are integer, every time we run Algorithm 5.1 we widen at least one constraint interval by at least one unit. Since we do not allow to undo the changes in step 9, the algorithm terminates at the latest when all PESP activity bounds are maximally relaxed, that is after $\sum_{a \in A}(\tau_a^\ell + \tau_a^u)$ iterations.

That implies that if we ran the algorithm without time limits on an an instance of type PESP-IP-Ext, it would terminate in finite time (under the assumption that

RAM and memory of our computer are enough to resolve the occurring conflicts as specified in Algorithm 5.1).

For instances of type PESP-IP-Dep, that is, in particular, when we model changes in the trip times as adjustments of trip times instead of a relaxation of the activity interval, we can give no such guarantee since a trip time that is changed once from value r to r' may be changed back in a later iteration, if the corresponding activity occurs in a second conflict. So-induced cycles can be prevented to a certain extent by enlarging conflict graphs as described in the previous section. A second approach to avoid the repeated changing of trip times is described in the following.

Tabu search

In Algorithm 5.2, conflicts are resolved iteratively. If one conflict is resolved, we search for the next one and resolve it. One thing that might happen is that the resolution of one conflict in some iteration, creates a new conflict. If we undo the first change, it might resolve the second conflict but again generate the first one.

There are several possibilities to avoid this going back-and-forth. Note that this situation would have been avoided if more activities would have been taken into account in resolving the first conflict. However, as this increases the size of the problems to be solved, we use a different approach in addition. For each $a \in A$, we define κ_a as the number of times the slack variables for this activity have been changed. Then the objective coefficient, penalizing changes to this activity, is multiplied by g^{κ_a} $(g > 1)$ to make changes to this activity more expensive and at the same time avoiding models to become infeasible if this activity has to be changed back again. Throughout the experiments, we use $g = 2$.

Finding a timetable & post-optimisation

If the iteration limit or time limit in Algorithm 5.2 is not exceeded, the algorithm terminates with a conflict-free PESP instance. A dedicated PESP-solver can be used to solve such an instance (cf. Kümmling et al., 2015a; Schrijver and Steenbeek, 1993). Once we have this timetable, together with the original PESP instance, we can build the PESP-IP-Dep model and restrict the event times to be in an interval, given by the event time from the timetable plus or minus a deviation $\eta \in [0, T/2]$, where $\eta = 0$ corresponds to no freedom and $\eta = T/2$ to full freedom. So suppose event π is scheduled to be at .27, we then add the restriction that $\pi \in [27-\eta, 27+\eta]$.

We call this model PESP-IP-Dep(η). The reason to use a model based on PESP-IP rather than CPF, is that the event times are explicitly included in this model.

If we first solve the model with $\eta = 0$, we find the changes to the activities that are needed to make the found timetable feasible and we can determine the objective value that is achieved when making this PESP instance feasible. In a next step, we increase η to a positive value and resolve the model. Since the timetable was feasible for PESP-IP-Dep(0), it surely is for PESP-IP-Dep(η) where $\eta > 0$. Hence, better solutions might be found since the feasible space of this model increases in η. So reoptimizing leads to a new change in the activity bounds and a new objective value that is not higher than the objective value obtained with $\eta = 0$.

5.4 Experiments

In order to test our methodology in practice, and to analyse the impact of different methods to enrich conflicts, we have tested it on several instances which are provided by Netherlands Railways and are based on the Dutch railway network. They vary between a tiny instance and the complete Dutch network. We present a description of the instances, together with the computational results in this section. Computations are performed on a 64-bit PC with an Intel(R) Core(TM) i7-4700MQ processor with 3.40 GHz and 16.00 GB of RAM installed, operating under Windows 7 Professional. The implementation is done using JAVA 1.8, IBM ILOG CPLEX 12.6.2 (IBM, 2015) is used to solve the MIP models, allowing for 4 parallel threads.

In this chapter, several methods are shown to enrich conflicts. Different ways of enriching conflicts can result in different solutions to the model on how to change some of the PESP-activities. This in turn can influence the number of conflicts found in total. If a solution to some conflict is found that is not only feasible with respect to this specific conflict, but also in the larger context of the total PESP instance, the number of conflicts that must be resolved is likely to be small, but the computation time might be large. On the other hand, resolving many small conflicts might lead to a lot of iterations of the algorithm, but probably to a relatively short computation time. Therefore, in the experiments we tested many different methods of enriching conflicts, to determine a pattern in which methods work well and which do not.

For finding a conflict or a timetable, the CADANS solver (Odijk et al., 2002; Schrijver and Steenbeek, 1993) is used. This solver finds a minimal conflict or a timetable based on Constraint Programming. For an instance of the size of the Dutch railway network, the computation time is often only a few minutes (cf. Caimi et al., 2017, sec. 6.1). However, our methodology does not depend on the algorithm used for conflict detection and any other algorithm for conflict detection could be used instead.

To solve the conflicts, we used the approach described in Section 5.3.2 based on fixed trip times, since the considered instances contain fixed trip times. Instead of solving PESP-IP-Dep however, we used an equivalent formulation based on the cycle periodicity formulation which has been proven to be faster in practice, stated as CPF-Dep in Appendix 5.A. To solve CPF-Dep, a minimum weight spanning tree is found where the weights of the activities are the number of possibilities for the corresponding process time, i.e., $w_a = u_a - \ell_a + 1$ for activities $a \in A$. Based on this tree, an integral cycle basis is generated.

Throughout the post-optimization, we first solve the PESP-IP-Dep(η) model with $\eta = 0$, to determine the initial objective value. Next, we relax the model to $\eta = 5$ and reoptimize, to find the objective value for this model with more freedom.

5.4.1 Den Helder - Schagen (Hdr-Sgn)

The first instance we consider is a small single-track network between stations Den Helder and Schagen.

Instance description

Physical network. The network consists of four stations as is shown in Figure 5.3, where the layout of this instance is displayed. There is a bridge between Anna Paulowna and Den Helder Zuid, which is shown as a dashed line, that can be opened during some time of the day and hence no trains can pass during that time. This means that the passing times of trains at this bridge must be known in order to know when the bridge can be opened. Note that trains can pass each other only at the stations, since the intermediate tracks are all single track.

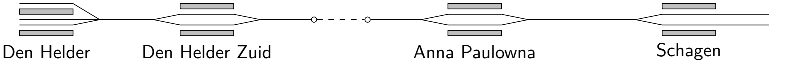


Figure 5.3: Den Helder - Schagen network.

Train services. On this network, two train lines are operated. Line 1 is an Intercity line, stopping only at Den Helder and Schagen and running in both directions, twice per hour. Line 2 is a so-called sprinter line, that runs once per hour and stops at all stations.

Activities. For service reasons, line 1 is required to drive in a 30-minute pattern, i.e. the trains leave the first station of their journey 30 minutes apart from each other. Furthermore, they should leave Den Helder at .29 or .59, because of connection times to other modes of transport.

In Table 5.1, an overview is given of the different PESP-activities that are present in this instance. The different types of activities are shown in the first column, and how often they occur in the instance is shown in the second column. A trip time activity refers to a situation where a train does not stop at some station, a trip-dwell activity refers to the situation where a train drives to a station and dwells there for some time. A synchronisation activity requires trains to drive in a given pattern (for example, 30 minutes apart). A fixation activity specifies a specific departure time. Safety activities all deal with headway times.

Objective coefficients and algorithmic limits. A trip(-dwell) activity involves a given planned trip time r, including a dwell time in case of trip-dwell activities. This trip time is used to define the maximum allowed changes to the lower (upper) bound of an activity bound as shown in column 3 (4). The objective coefficients for the changes to lower and upper bound are shown in the last two columns. A linear objective is used in this model.

In this instance, we do not allow to change safety activities, all other activities can be adjusted. The parameter sets are chosen together with NS planners.

Activity type	Number	Maximum allowed slack		Objective coefficient	
Trip time	20	$\min\{1, r\}$	2	$\max\{20 - 2r, 10\}$	$\max\{7 - r, 2\}$
Trip-dwell	4	$\min\{1, r\}$	2	$\max\{18 - 2r, 8\}$	$\max\{6 - r, 2\}$
Synchronisation	2	10	10	5	5
Fixation	2	5	5	5	5
Safety	156	0	0	0	0

Table 5.1: Bounds and coefficients for changes

As time limits, we used one hour, and the iteration limit was set to 500. Because this instance is small, these limits are actually never met.

Results of the methodology

When running the algorithm, a minimal conflict was found involving the Intercity lines in both directions and the sprinter line in one direction. In total 25 nodes and 38 activities were involved. Using $00ff0f$ as the maximum level of additional activities, a feasible timetable is found in two iterations.

The proposed solution is to decrease the trip time on one of the trips, i.e., to accelerate the train, and to change the frequency of the Intercity line to 29/31 instead of 30/30. The fixations are relaxed by one minute. The total computation time is 0.9 seconds, while finding a solution for the models to resolve a conflict took 0.2 seconds.

The most prominent reason why the resolution of the first conflict caused another conflict, is that the sprinter train was involved in one direction only. If both directions were taken into account, which can be achieved by enriching a conflict by the method $00ft0f$ for example, the solution was found in one iteration. On single track networks, every train that uses this track provides a large limitation on the changes that are feasible. The solution in this case is identical to the solution that was found in the case with the two iterations. Computation times are comparable as the total computation time now is 0.7 seconds and 0.2 seconds are used to resolve the conflict.

5.4.2 Rotterdam - Utrecht (Rtd–Ut)

Instance description

Physical network. The underlying network for this instance are the tracks between Gouda and Zwolle, with legs from Gouda to Rotterdam and The Hague, as well as from Zwolle to Leeuwarden and Gouda. The majority of this network is double track, except for the part between Gouda and Utrecht, which mainly has four tracks. The interested reader is referred to www.sporenplan.nl for more details about this network. Figure 4.2b also displays the location of these stations in the railway network.

Train services. The basis for the instance are the Intercity trains that share tracks between Rotterdam and Utrecht. Next to this, we added some additional trains. In total, the instance consists of the following trains (the numbers are added for references later on):

500 Intercity between Rotterdam and Groningen (1 time per hour).

12500 Intercity between Rotterdam and Leeuwarden (1 time per hour).

2000 Intercity between Rotterdam and Utrecht (2 times per hour).

2800 Intercity between The Hague and Utrecht (2 times per hour).

4000 Sprinter train between Rotterdam and Uitgeest (2 times per hour). This train shares infrastructure with the Intercity trains between Rotterdam and Gouda. Next, it has to cross the Intercity paths halfway between Gouda and Utrecht.

5600 Sprinter train between Utrecht and Zwolle.

Activities. Both lines 2000 and 2800 are synchronized to drive exactly 30 minutes apart from each other. Furthermore, lines 500, 12500 and 2800 are synchronised at Utrecht and Rotterdam to drive 15 minutes apart from each other.

Objective coefficients and algorithmic limits. In Table 5.2 the maximum allowed changes and the objective coefficients are shown, which were chosen based on expertise from NS planners. Furthermore, the number of activities of each type is displayed.

Activity type	Number	Maximum allowed slack		Objective coefficient	
Trip time	512	$\min\{1, t\}$	2	$\max\{20 - 2t, 10\}$	$\max\{7 - t, 2\}$
Trip-dwell	167	$\min\{1, t\}$	2	$\max\{18 - 2t, 8\}$	$\max\{6 - t, 2\}$
Frequency	93	10	10	30	30
Fixation	18	5	5	20	20
Connection	14	1	5	30	25
Safety	2257	0	0	0	0

Table 5.2: Bounds and coefficients for changes in Rtd–Ut

The PESP instance contains 699 events and 3061 actitivies. This instance contains many Intercity trains, sharing the same infrastructure. However, all these trains might have different driving characteristics as they use different train types. Therefore, the corresponding trip times can be different, which is undesirable because it consumes more capacity on these tracks and it easily gives rise to conflicts. Hence, solutions that redistribute buffer times are desired. In order to find these, the corresponding actitivies are added and the corresponding objective term gets a coefficient of value 10 (the γ in (5.13)).

For the algorithm, we set an iteration limit of 500 iterations and a time limit of 2 hours.

Results of the methodology

Using the encoding stated in Section 5.3.2, we tested all possible combinations of the following ways to enrichting a conflict:

- $n_1 \in \{0, 1, 2\}$, the number of previously found conflicts that are added.
- $n_2 \in \{0, 1, 2\}$, the neighbourhood depth.
- $n_3 \in \{0, 2, 4, 6, 8, 10\}$, the number of additional trips of the train lines that are involved.

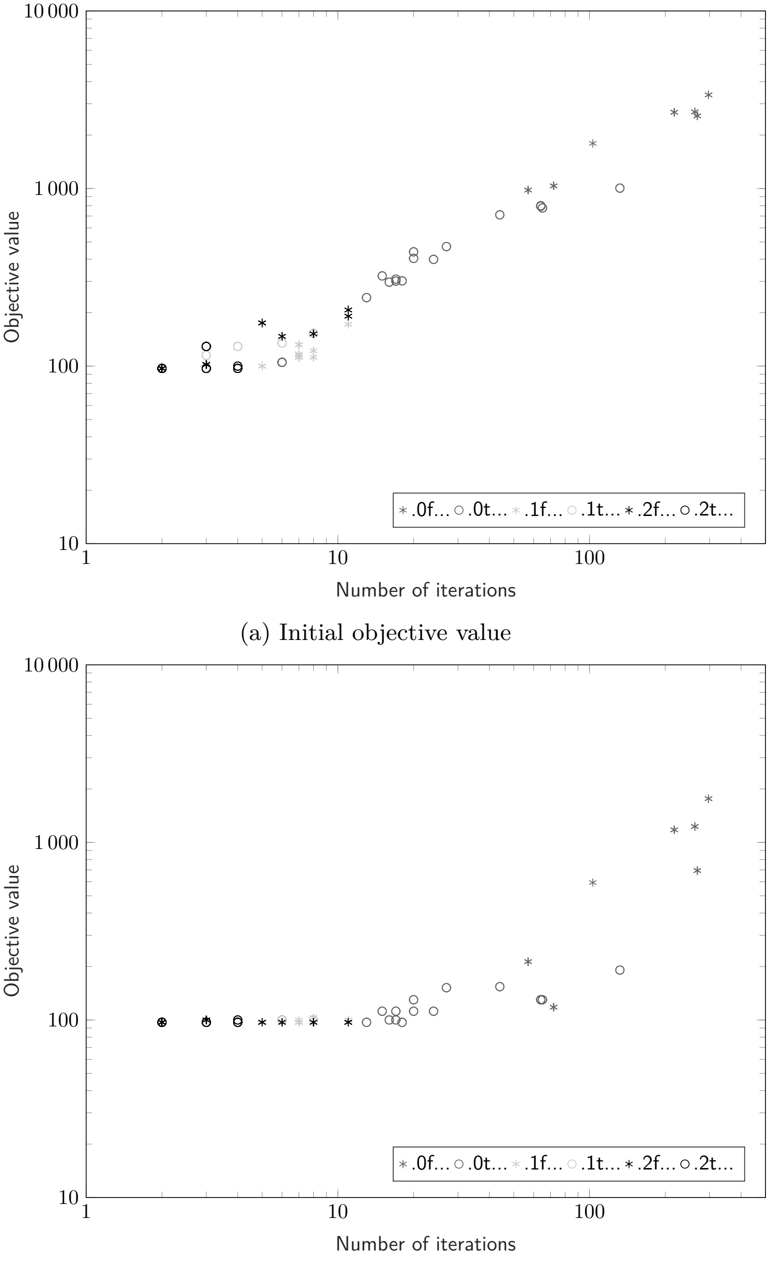
Furthermore, the true/false parameters could take both values, which correspond to adding the interrelated actitivies or not, adding all trains in single track parts or not and providing solutions of subproblems as a warm start or not.

This led to 432 possible methods of adding additional actitivies. Out of these 432 methods, 48 have led to the algorithm exceeding the iteration limit. All of these 48 methods had neighbourhood depth equal to zero and no interrelated actitivies were added.

For the remaining methods, results are shown in Figures 5.4 and 5.5. The legend shows which method is used to enrich conflicts. For example, .1*t*... shows that first all actitivies are added that have a relation with the conflict, and next all interrelated actitivies are added. The dots indicate that the parameter for this method is left open, so in the figure, there are several dots corresponding to .1*t*..., all indicating a different setting where the parameter on the dots is varied.

In Figure 5.4, a plot is shown of the number of iterations that are performed versus the total objective value (Figure 5.4a) and the objective value obtained after post-optimization (Figure 5.4b). By using the three different types of gray, we distinguish between the results for adding neighbouring activities or not. Also, we distinguished between adding interrelated activities or not by using different symbols. As is clear from these figures, adding more activities leads to a smaller number of iterations and a better initial objective. Also, for the objective obtained after post-optimization, the same observation can be made. For neighbourhood depth 1 and 2, a solution with objective value 97 was obtained in nearly all methods. Also, a difference can be seen between the methods that add interrelated activities or not. In general, doing so leads to fewer iterations to find a feasible timetable.

In Figure 5.5, plots are shown for the total time in seconds versus the objective value. The same color schemes are used here as in the previous figure. As is clear, in general



(a) Initial objective value

(b) Improved objective value

Figure 5.4: Rtd–Ut - Number of iterations versus objective value

the medium-gray dots (neighbourhood depth 0) correspond to methods that lead to solutions being found rapidly, however, with a high objective value in general. Also, if these solutions are improved, they are still worse than the solutions corresponding to methods with a different neighbourhood depth. There are a few exceptions, having a really long computation time and a high objective value.

In general we can observe that resolving only the minimal conflicts does not work well. All the methods that have led to exceeding the iteration limit, hardly added any activities to the minimal conflict. Furthermore, we saw a tendency that increasing the neighbourhood depth leads to better solutions. However, this may come at the cost of an increase in computation time. A trade-off has to be made, where to stop adding additional activities. Adding too many leads to an increased problem size, without providing additional relevant information.

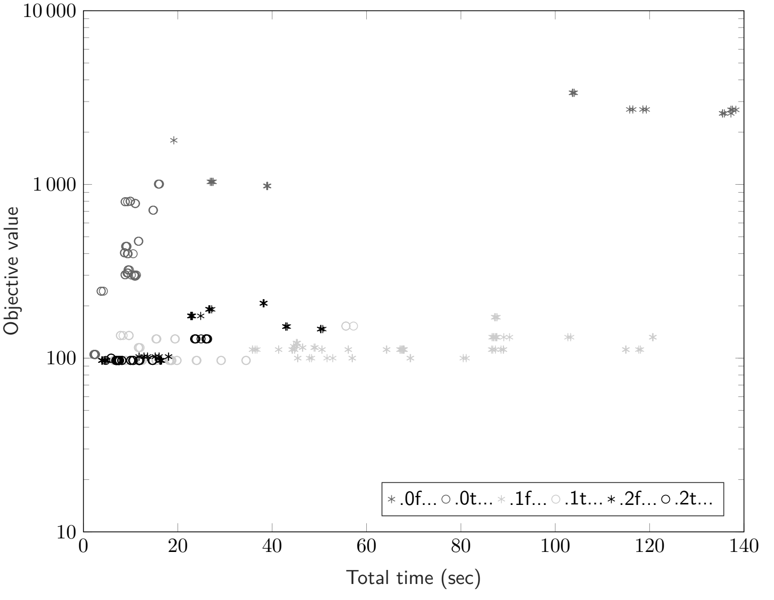
5.4.3 Dutch Network 2013 (NL2013)

Instance description

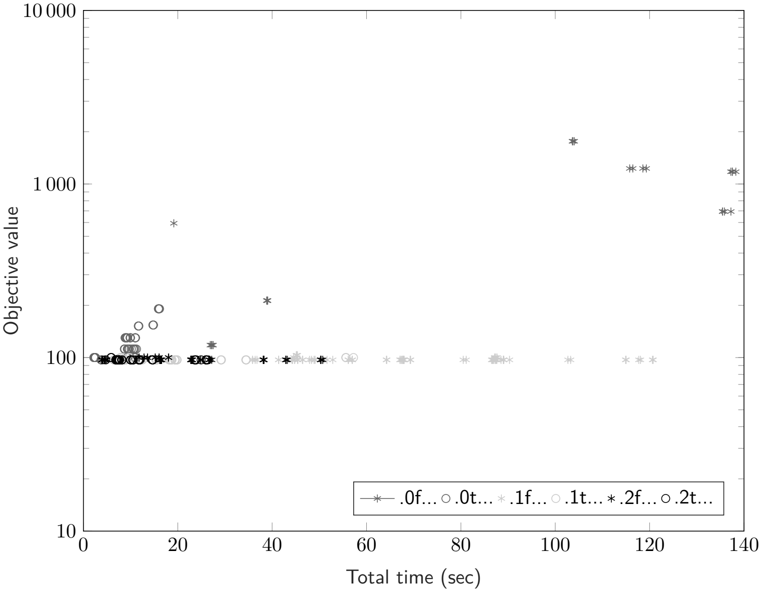
This instance was used as a basis to generate the full Dutch timetable of 2013, including transfer and synchronisation requirements. The timetabling instance is initially infeasible. In order to obtain the actual timetable in 2013 in the Netherlands, trip time and synchronisation requirements were changed manually.

The Dutch railway network is one of the most heavily used railway networks in Europe (Boston Consulting Group, 2017). There are many trains operated on this network, many of them sharing a piece of infrastructure in the network, thus generating many interdependencies in the network. Therefore, finding a feasible timetable for this instance is a challenging task. This is why we ran experiments on this network, to test the performance of our algorithmic framework. The only expected changes are on trip times and synchronisation requirements.

Objective coefficients and algorithmic limits. The allowed changes, the corresponding coefficients and the number of PESP-constraints for each type are shown in Table 5.3. They are similar to the previous case. Here, the constraints to redistribute buffer times between stations are not added to the IP. We have set a maximum of 500 iterations and 3 hours of computation time. The network consists of 9085 nodes and 75309 activities with 448 trains in total.



(a) Initial objective value



(b) Improved objective value

Figure 5.5: Rtd–Ut - Total time versus objective value

Activity type	Number	Maximum allowed slack		Objective coefficient	
Trip time	5670	$\min\{1, t\}$	2	$\max\{20 - 2t, 10\}$	$\max\{7 - t, 2\}$
Trip-dwell	2966	$\min\{1, t\}$	2	$\max\{18 - 2t, 8\}$	$\max\{6 - t, 2\}$
Frequency	291	10	10	30	30
Fixation	18	5	5	20	20
Connection	0	1	5	30	25
Other	66212	0	0	0	0

Table 5.3: Bounds and coefficients for changes in NL2013

Results of the methodology

For this instance again several methods are tested to enrich a conflict. We varied the parameters of the methods as follows:

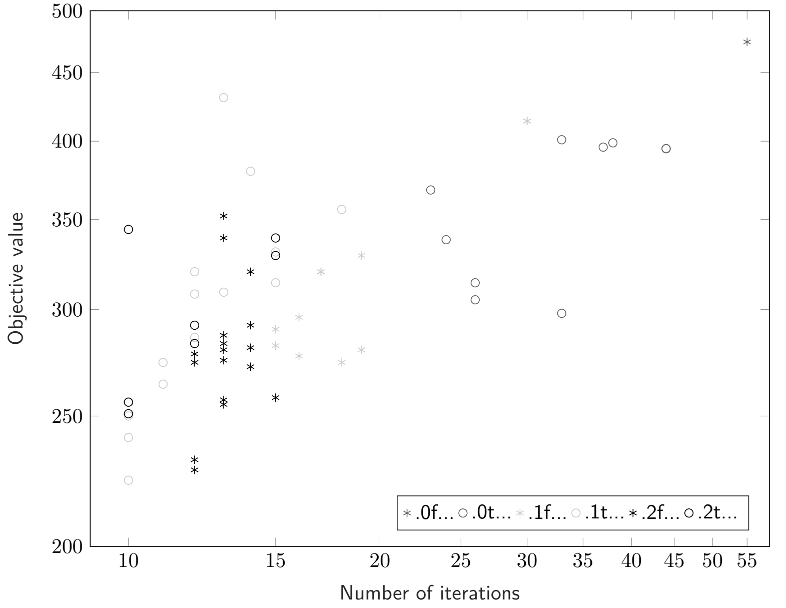
- $n_1 \in \{0, 1, 2\}$, the number of previously found conflicts that are added.
- $n_2 \in \{0, 1, 2\}$, the neighbourhood depth.
- $n_3 \in \{0, 2, 4\}$, the number of additional trips of the train lines that are involved.

Furthermore, the true/false parameters could take both values, which correspond to adding the interrelated activities or not, adding all trains in single track parts or not, and providing solutions of subproblems as a warm start or not.

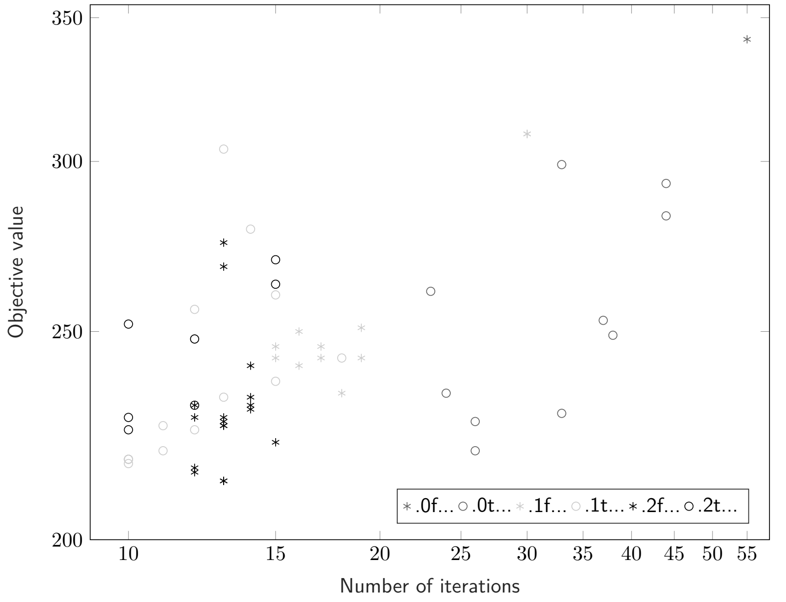
This led to 216 possible methods of adding additional activities. Out of these 216 methods, 1.8% exceeded the iteration limit, which are methods that had neighbourhood depth 0 and no interrelated activities added. 49.5% exceeded the overall time limit and 20.37% exceeded the time limit allowed for finding conflicts or a timetable in one of the iterations. We do not recognize a clear pattern in which combination of parameters causes violation of the time limit. The results for the remaining methods are shown in Figures 5.6 and 5.7.

Although the results are less clear than in the previous case, again we see that the methods that have neighbourhood depth 0 tend to lead to a higher number of iterations. Although the computation time in general is low, again the objective value is very high. For neighbourhood depth 1 and 2 the results are identical.

Another observation here is that almost all methods that have neighbourhood depth 0 and no addition of interrelated activities, did not lead to a feasible result within the given time and iteration limits.

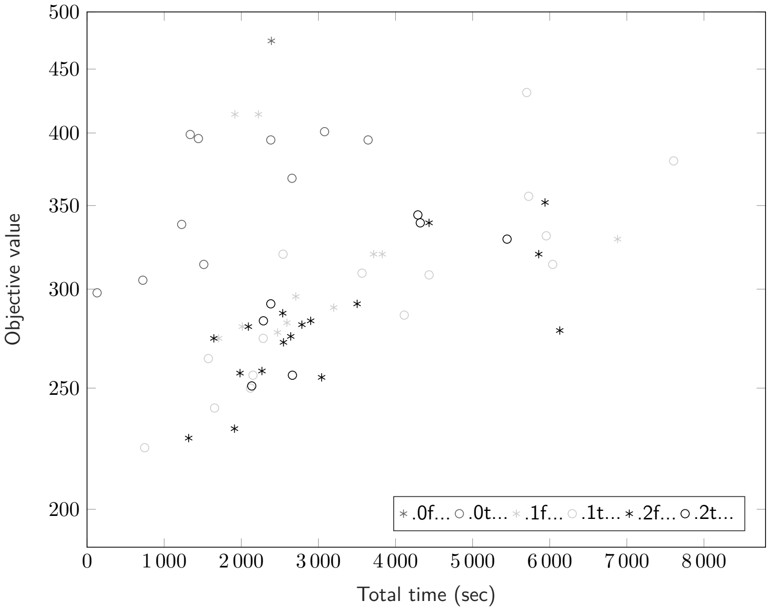


(a) Initial objective value

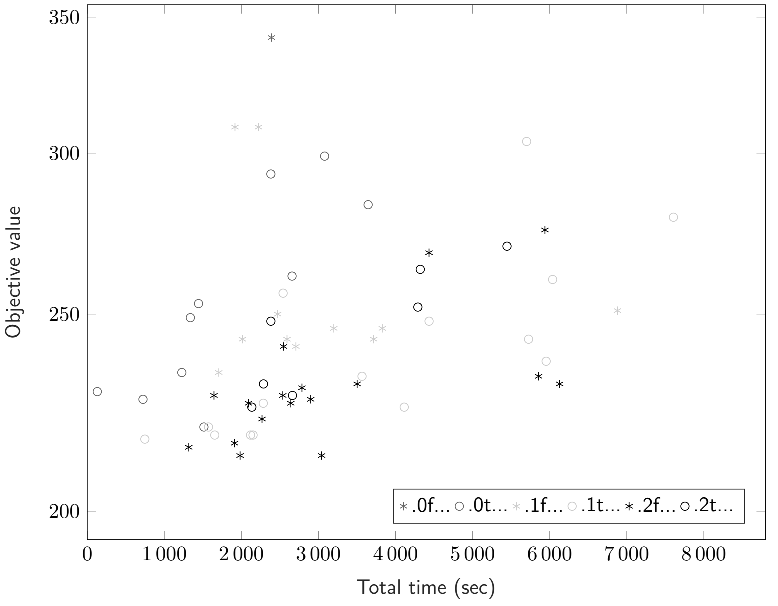


(b) Improved objective value

Figure 5.6: NL2013 - Number of iterations versus objective value



(a) Initial objective value



(b) Improved objective value

Figure 5.7: NL2013 - Total time versus objective value

5.5 Conclusion and Discussion

We developed a methodology to relax PESP activities to resolve infeasible PESP instances. This approach supplements current timetabling algorithms, which suffer from the fact that increased demand for capacity usage as well as quality requirements often lead to (on first sight) infeasible timetabling instances. We resolve conflicts in a PESP model with as few deviations as possible, based on predefined activity weights and bounds. Our approach is iterative in the sense that we find a conflict in the existing PESP instance, solve this conflict using a MIP model, and then search for the next conflict. In contrast to existing approaches on resolving infeasible PESP instances, our approach can deal with fixed trip times, an assumption that is often made in PESP instances arising from railway timetabling.

We find that in our iterative approach of finding and resolving conflicts it is important not to resolve only minimal conflicts, but to carefully add more activities from the timetabling instance to these minimal conflicts, in order to find good solutions as well as to improve computation times. We proposed several methods to enrich the minimal conflicts. From our computational results, we have seen that adding neighbouring activities in the constraint graph in general leads to better results. In our experiments based on parts of or the whole the Dutch railway network, feasible timetables are found in reasonable time in most cases.

Our approach requires the choice of many parameters, namely the allowed deviations of the original bounds and the weights in the objective function, which will steer the algorithm towards a solution. In practice, these parameters would need to be chosen based on the expertise and preferences of the railway operator - and could be adjusted based on the feedback of the operator when seeing the generated solution. It would even be possible to incorporate expert feedback (in the sense of resetting and fine-tuning parameters) after the resolution of each conflict.

Appendix

5.A Solving Conflicts by Extending the Cycle Periodicity Formulation

Analogously to the described extension of PESP-IP, we can extend the Cycle Periodicity Formulation for PESP to resolve conflicts. The cycle periodicity formulation is explained in Section 2.1.2. In makes use of a 'process time' of an activity $a \in A$, which we denote here as x_a, i.e.,

$$x_a = \pi_j - \pi_i + Tp_{ij}. \tag{5.15}$$

This appendix details how the extensions of PESP-IP can be modelled using the Cycle Periodicity Formulation.

5.A.1 Equivalent for PESP-IP-Ext

We can extend CPF in the same way as described in Section 5.3.2 for PESP-IP to resolve conflicts.

We assume a cycle basis $\mathcal{B}$ is given. The Cycle Periodicity Formulation equivalent to PESP-IP-Ext is denoted by CPF-Ext and can be stated as follows:

$$\begin{aligned}
\min \quad & \sum_{a \in A} w_a^l s_a^\ell + w_a^u s_a^u & & \text{(5.16a)}\\
\text{s.t.} \quad & \sum_{a \in \mathcal{C}+} x_a - \sum_{x \in \mathcal{C}-} x_a = Tq_\mathcal{C} & \forall \mathcal{C} \in \mathcal{B} \quad & \text{(5.16b)}\\
& \ell_a - s_a^\ell \leq x_a \leq u_a + s_a^u, & \forall\, a \in A \quad & \text{(5.16c)}\\
& a_\mathcal{C} \leq q_\mathcal{C} \leq b_\mathcal{C} & \forall \mathcal{C} \in \mathcal{B} \quad & \text{(5.16d)}\\
& s_a^\ell \in [0, \tau_a^\ell],\ s_a^u \in [0, \tau_a^u] & \forall\, a \in A \quad & \text{(5.16e)}\\
& x_a \in \mathbb{R} & \forall\, a \in A \quad & \text{(5.16f)}\\
& q_\mathcal{C} \in \mathbb{Z} & \forall \mathcal{C} \in \mathcal{B}. \quad & \text{(5.16g)}
\end{aligned}$$

The objective function should be the same and the activity bounds are altered in the same way, by adding and subtracting the right slack variables, which are the same as well, so Constraint (5.16a), (5.16c) and (5.16e)–(5.16g) are essentially the same. The sum of all process times should still be equal to a multiple of T as in Constraint (5.16b). Finally, the number of multiples of T in each cycle $\mathcal{C}$ can be bounded by (5.16d), where the bounds are calculated as:

$$a_\mathcal{C} = \left\lceil \frac{1}{T} \left(\sum_{a \in \mathcal{C}+} (\ell_a - \tau_a^l) - \sum_{a \in \mathcal{C}-} (u_a + \tau_a^u) \right) \right\rceil, \tag{5.17a}$$

$$b_\mathcal{C} = \left\lfloor \frac{1}{T} \left(\sum_{a \in \mathcal{C}+} (u_a - \tau_a^u) - \sum_{a \in \mathcal{C}-} (\ell_a - \tau_a^l) \right) \right\rfloor. \tag{5.17b}$$

Here, we assume worst case situations, i.e., we calculate the lowest and highest possible values, based on the maximum allow changes, in order to have a model that is not too restrictive. This shows that PESP-IP-Ext and CPF-Ext are equivalent.

5.A.2 Equivalent for PESP-IP-Dep

In order to get the CPF equivalent for PESP-IP-Dep, we use a similar approach. The model is referred to as CPF-Dep and stated as

$$
\begin{aligned}
\min \quad & \sum_{a\in A} w_a^l s_a^l + w_a^u s_a^u && && (5.18a)\\
\text{s.t.} \quad & \sum_{a\in\mathcal{C}^+} x_a - \sum_{a\in\mathcal{C}^-} x_a = Tq_{\mathcal{C}}, && \forall\, \mathcal{C}\in\mathcal{B} && (5.18b)\\
& a_{\mathcal{C}} \le q_{\mathcal{C}} \le b_{\mathcal{C}} && \forall\, \mathcal{C}\in\mathcal{B} && (5.18c)\\
& \ell_a - c_a^{\ell} \le x_a \le u_a + c_a^u && \forall\, a\in A && (5.18d)\\
& c_a^{\ell} = s_a^{\ell} + c_a^{\ell,imp}, \quad c_a^u = s_a^u + c_a^{u,imp} && \forall\, a\in A && (5.18e)\\
& c_a^{\ell,imp} = \left(\lambda_a^{\ell}\right)^{\mathsf{T}} s, \quad c_a^{u,imp} = \left(\lambda_a^{u}\right)^{\mathsf{T}} s && \forall\, a\in A && (5.18f)\\
& L_a = \ell_a - \tau_a^{\ell} - \max\{c_a^{\ell,imp}\} && \forall\, a\in A && (5.18g)\\
& U_a = u_a + \tau_a^{u} + \max\{c_a^{u,impl}\} && \forall\, a\in A && (5.18h)\\
& a_{\mathcal{C}} = \left\lceil \frac{1}{T}\left(\sum_{a\in\mathcal{C}^+} L_a - \sum_{a\in\mathcal{C}^-} U_a\right)\right\rceil && \forall\, \mathcal{C}\in\mathcal{B} && (5.18i)\\
& b_{\mathcal{C}} = \left\lfloor \frac{1}{T}\left(\sum_{a\in\mathcal{C}^+} U_a - \sum_{a\in\mathcal{C}^-} L_a\right)\right\rfloor && \forall\, \mathcal{C}\in\mathcal{B} && (5.18j)\\
& s_a^{\ell} \in [0,\tau_a^{\ell}], \quad s_a^u \in [0,\tau_a^u] && \forall\, a\in A, && (5.18k)\\
& x_a\in\mathbb{R}, \qquad c_a^{\ell}, c_a^u \in\mathbb{R}, \qquad \lambda_a^{\ell},\lambda_a^u \in \{0,\pm1\}^{2|A|} && \forall\, a\in A && (5.18l)\\
& q_{\mathcal{C}}\in\mathbb{Z} && \forall\, \mathcal{C}\in\mathcal{B}. && (5.18m)
\end{aligned}
$$

Here, constraints (5.18b) relate the processes in a cycle to an integer multiple of T. Constraints (5.18d) give bounds on the allowed process times and correspond to the bounds of a PESP activity. Constraints (5.18c) bound the integer variable for each cycle, denoting how many multiples of T can possibly occur in a cycle. Bounds on these variables are calculated by (5.18i) and (5.18j). It makes use of the smallest and largest possible bounds for each activity, which are calculated by (5.18g) and (5.18h), in order to avoid having a model that is too restrictive. Note that again PESP-IP-Dep and CPF-Dep are equivalent.

Chapter 6

An Adjustable Robust Optimization Approach for Periodic Timetabling*

6.1 Introduction

The timetable of a railway operator is the backbone of daily operations. It is therefore no surprise that forced deviations from the planned timetable can be extremely costly: A recent investigation of the Dutch government estimated the annual societal costs of disturbances in the Dutch railway system at 400 to 500 million euros (Kennisinstituut voor Mobiliteitsbeleid, 2019). These costs come from productivity loss, and from the uncertainty that passengers experience in the time their journey takes. Resilience against possible disturbances is therefore a crucial aspect of the operated timetable.

A railway timetable is an assignment of departure and arrival times for a given set of train services. The input to timetabling specifies this set of train services, including all locations where the train shall call. Our focus in this chapter is on generating (periodic) timetables, which have to be designed a few months before the actual operation.

*This chapter, up to minor modifications, is a direct copy of G.J. Polinder, T. Breugem, T. Dollevoet, and G. Maróti (2019): *An adjustable robust optimization approach for periodic timetabling.*

The actual operations often differ from what was planned. For example, trains may not be able to run at full speed or have to dwell longer at some stations. These are all *disturbances*, events that can influence the execution of a timetable by causing delays. In this chapter, we focus on small disturbances, leading to delays of up to a few minutes. Although the timetable was designed several months before the operations, it should still be operable with a limited amount of delays, no matter when and where the unpredictable disturbances occur. A robust timetable is one for which this is the case. In order to achieve robustness, a timetable allocates time supplements on top of the technically minimal travel times and dwell times. These supplements can absorb the unexpected increase of travel and dwell times and thereby limit the accumulation of delays.

In this chapter, we consider disturbances that reoccur in every cycle period during a day. Because their reappearance is the same for every cycle period of the timetable, we refer to them as *periodic* disturbances. Periodic disturbances are seen, for example, on autumn days when the fallen leaves lead to slippery rails, and thus to increased travel times. When designing the timetable to be operated in autumn, it is known that fallen leaves lead to increased travel times during some days. However, it is not known yet at which days this will happen. Similarly, the dwell times at some stations increase on sunny summer days when many people take the train to the beach. One can expect this to happen, say, in August, but it is unknown which days will be sunny. In both cases, a timetable is announced for a period of several weeks or months, while the periodic disturbances affect a few days only. The timetable must balance the service quality of the normal days and the delays of the disturbed days.

Achieving robustness generally conflicts with achieving efficiency in a timetable. In an efficient timetable, passengers have little waiting time, and can travel quickly from their origin to their desired destination. Robust solutions prefer larger time supplements since they offer more potential to absorb delays. However, the increased time supplements are added on undisturbed days, as well. That is, the passengers will experience longer travel times even if there are no disturbances on that day.

In this chapter we aim at finding a periodic timetable which is robust against small periodic disturbances. An adjustable robust periodic timetable is one that can easily be adapted to cope with such disturbances. In particular, the periodic disturbances can be incorporated without changing the structure of the timetable. Note that this leads to a complex optimization problem. The underlying periodic timetabling problem is NP-hard (Serafini and Ukovich, 1989) and known to cause major computational challenges in practice, as well. The robust extension adds to the complexity in that it needs to keep track of the consequences of many possible disturbances.

In this chapter, we formulate the Robust Periodic Timetabling Problem (RPTP), an adjustable robust formulation for the periodic timetabling problem. We propose a solution method for a parametrized uncertainty region, and we study the interplay between three values: (*i*) the severity of the disturbance; (*ii*) the total amount of timetable adjustments; and (*iii*) the efficiency of the undisturbed timetable. We apply our approach to practical instances from Netherlands Railways (NS), and show that good solutions can be obtained for challenging real-world instances.

To the best of our knowledge, this chapter contains one of the first approaches to incorporate periodic disturbances in the Periodic Event Scheduling Problem, and to combine this with standard robust optimization techniques. Furthermore, we are the first to analyse the efficiency and robustness of a timetable with respect to a recovery budget in the context of periodic timetabling in such detail.

The contribution of this chapter is twofold. Firstly, we propose an adjustable robust extension of the periodic timetabling problem. Second, we demonstrate the viability of adding adjustable robustness extensions to practically relevant optimization problems. In fact, periodic timetabling is an application where the nominal (i.e., non-robust) optimization problems pose a considerable challenge. Our robust extension turns out to be tractable by using the proper combination of modelling and solution techniques.

The remainder of this chapter is organized as follows. In Section 6.2 we formalize the RPTP. We give an overview of related work in Section 6.3. In Section 6.4 we propose a mathematical formulation for the problem. We introduce two assumptions in Section 6.5, on which we base our solution method in Section 6.6. We apply this method to practical instances from Netherlands Railways in Section 6.7. Finally, the chapter is concluded in Section 6.8.

6.2 Problem Description

The periodic timetabling problem is explained in Chapter 2. The key components for this problem are the set of events V and of the set of activities (constraints) A. Each constraint states a restriction of the time difference between two events, and is generally written as

$$\ell_{ij} \leq \pi_j - \pi_i + Tp_{ij} \leq u_{ij}. \tag{6.1}$$

For more information about these activities and periodic timetabling in general, we refer to Chapter 2. In this section, we focus on the incorporation of robustness in periodic timetabling and highlight the problem we deal with in this chapter.

The goal of the PTP is to find a feasible timetable. It is common practice, however, to include an objective which maximizes the efficiency or the robustness of the timetable. These two objectives are conflicting: Efficiency is achieved by minimizing the planned duration of trip and dwell activities, while robustness is achieved by adding time supplements on these activities. The duration of an activity is defined as the time difference between the two involved events, i.e., as $\pi_j - \pi_i + Tp_{ij}$ in (6.1).

In practice, operations often differ from the plan due to disturbances, thus rendering the planned timetable infeasible. We assume that the uncertainties manifest themselves in the lower bounds of the PESP-constraints (6.1). This assumption is in line with the literature (Goerigk, 2015; Goerigk and Schöbel, 2010; Liebchen et al., 2009). We note that uncertainties in the upper bounds can be easily incorporated in the models, and that they have no effect on the methodology proposed in this chapter.

Suppose that a driving activity is given that states that the time difference between arrival event π_2 and departure event π_1 should be at least 10 minutes, and at most 13 minutes. This constraint can be written as $\pi_2 - \pi_1 + Tp_{12} \in [10, 13]$, which is visualized in Figure 6.1 (where the cycle time is 60 minutes). The highlighted area

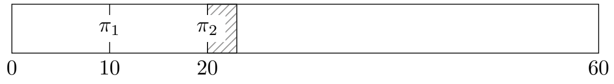


Figure 6.1: Example Periodic Constraint.

shows the possible event times π_2, given the scheduled time $\pi_1 = 10$. For this activity, scheduling π_2 at minute 20 is a feasible solution. In fact, this would be the unique optimal solution from an efficiency-perspective.

Now suppose that a disturbance affects this activity: The travel time will be at least 11 minutes due to slippery tracks. This implies that the feasibility interval changes from $[10, 13]$ to $[11, 13]$, as shown in Figure 6.2. Notice that the highlighted area shrunk by 1 minute. As a result, the originally scheduled pair of event times

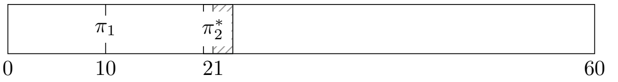


Figure 6.2: Example Periodic Constraint with Disturbance.

becomes infeasible. A possible solution is to shift event time $\pi_2 = 20$ by one minute to $\pi_2^* = 21$, thereby making the schedule feasible again. The shift of event time π_2 is known as the *recourse action* for the given disturbance.

Given an uncertainty set (i.e., a set of possible disturbances), we determine a timetable that maximizes the efficiency while we require that for *every* disturbance in the uncertainty set, at least one recourse action should exist such that the timetable remains feasible. This implies that two things have to be taken into account whilst constructing a robust timetable: (*i*) the set of possible disturbances, and (*ii*) the set of feasible recourse actions. The latter is restricted by introducing a *budget*. The budget limits the recourse actions: The total amount of adjustments must not exceed the budget. This assures that the adjustable timetable has the desirable property of not being too different from the nominal timetable.

The RPTP can now be stated as follows: Given an uncertainty set, and a recovery budget, create a feasible periodic timetable such that the efficiency is optimized, whilst assuring that, given the budget, there exists feasible timetable adjustments for each possible disturbance. A solution to the RPTP is specified by a nominal timetable, and a recourse action for every possible disturbance. By varying the uncertainty set and recovery budget, different 'good' timetables can be obtained.

6.3 Literature Review

When operating a periodic timetable in practice, disturbances and disruptions are faced on a daily basis. Disturbances are small perturbations to the operation of a timetable and generally cover small delays, while disruptions have much more influence and can cause long delays and train cancellations. Common examples of a disruption are the blockage of a track or failures of signs and switches. In the past, so called contingency plans were used to reschedule the trains (Ghaemi et al., 2017). There is a rich stream of literature on ad-hoc rescheduling of the timetable in case of a given disruption, see, for example, Ghaemi et al. (2018), Louwerse and

Huisman (2014), Meng and Zhou (2011), Veelenturf et al. (2016), Zhu and Goverde (2019, 2020). Also, much research has been devoted to rescheduling the timetable in case of known smaller disturbances, such as delays of several minutes, see D'Ariano et al. (2007), Pellegrini et al. (2015). In both cases, a real-time setting is considered, which means that the disturbance or disruption is known. Lamorgese and Mannino (2015) developed an algorithm to achieve automatic disturbance management, which is operational in Norway. We refer to Cacchiani et al. (2014) for an overview of recovery models for railway timetabling. In contrast, in robust timetabling, the occurrence of disturbances is considered already in the design of the timetable, i.e., in the planning phase.

The field of robust optimization has developed quickly in recent years. We refer to Ben-Tal et al. (2009b) and Bertsimas et al. (2011) for a detailed overview of robust optimization and associated techniques. Gorissen et al. (2015) also give a detailed overview of robust optimization techniques, thereby focusing on applicability in practice. Adjustable robustness is a specific technique in robust optimization, in which a two-stage problem is considered, where in the second stage an additional decision has to be made, according to some decision rule. This idea has been successfully applied to practical problems (see, e.g., Ben-Tal et al. (2005), Ben-Tal et al. (2009a)). Yanıkoğlu et al. (2018) give a survey on several techniques in adjustable robust optimization, thereby providing some examples of decision rules. Interesting work has been done in Iancu et al. (2013) and Zhen et al. (2018), and references therein, which prove optimality for classes of decision rules based on the underlying problem structure.

In recent years, robust timetabling has received much attention (see e.g., Cacchiani and Toth (2012), Goerigk and Schöbel (2010), Lusby et al. (2018) for detailed overviews). Although much interesting work has been done, there still seems to be a lack of consensus: As noted in Lusby et al. (2018), there is currently no clear and unique definition of robustness for timetabling. We therefore cover only a subset of the approaches in this overview which are most closely related to our research.

Most research regarding robust timetabling has focused on *aperiodic* timetabling. Fischetti and Monaci (2009) introduce light robustness, and apply this robustness concept to the aperiodic timetabling problem. The light robustness framework aims at finding a robust solution, without deviating too much from the optimal nominal objective value. The first papers that apply adjustable robustness in timetabling are Liebchen et al. (2009), Cicerone et al. (2009), and Cicerone et al. (2012). The adjustability of a solution relates to the existence of a rescheduling algorithm, which resolves possible infeasibilities as soon as the disturbance becomes known. Finally, Goerigk and Schöbel (2010) and Goerigk and Schöbel (2014) introduce the concept of Recovery-to-Optimality, which aims at minimizing the recovery cost of the solution (i.e., the cost incurred when making the disturbed solution feasible again).

One of the first approaches to increase the reliability of a *periodic* timetable that we are aware of is by Kroon et al. (2008). The authors apply a stochastic programming approach in order to improve the robustness of an input timetable, by adding time supplements at strategic places. Later on, this method is improved by Maróti (2017). Bešinović et al. (2016), Goverde et al. (2016) develop a robust timetable using a microscopic-macroscopic approach to timetabling and the robustness is measured as delay-settling time, which is obtained though Monte-Carlo simulation. Stochastic programming and adjustable robustness are closely related in that both approaches aim at finding a nominal solution that can be repaired in any disturbance scenario. The main conceptual difference is the objective. Stochastic programming models minimize the expected repair costs subject to a given probability distribution of the disruptions. In contrast, adjustable robustness considers worst-case repair costs; in particular, it does not assume any known probability distribution.

Another approach to increase the robustness of a timetable is presented by Dewilde et al. (2014). Here, a timetable is developed that minimizes the *expected* travel time of the passengers. To this end, the expected travel time is defined, for each edge in the event-activity network, as a function of the time supplement allocated to that edge. By minimizing the expected travel time, the trade-off between the planned travel time and the delays that occur in practice is automatically incorporated (see Sels et al., 2016). Another approach considering these time supplements is Sparing and Goverde (2017). Here, cycles in the event-activity network are considered, and critical circuits are sought (Goverde, 2007). Such a circuit determines the minimum cycle time that is possible, and hence the most buffer time in the timetable. More time supplements can be added in this case. If the cycle time decreases, the timetable becomes more robust. A related concept is found in Odijk et al. (2006), where a good order of trains is determined in order to maximize the buffer times between trains.

Liebchen et al. (2009) note that the adjustable robustness framework can also be applied to periodic timetabling. They observe that a choice has to be made regarding the (a)periodicity of the disturbances. They, however, do not provide any computational results for the periodic case. Liebchen et al. (2010) consider periodic timetabling, by rolling out a periodic instance to a non-periodic instance. Goerigk (2015) extends the Recovery-to-Optimality approach to the periodic case. For small cases, a mixed-integer programming approach is taken. For larger instances, for which this approach is intractable as the nominal problem already is notoriously difficult, a local search heuristic is applied. Within a provided time limit, this approach tries to improve the robustness of the current solution.

As is clearly shown in Lusby et al. (2018), current literature has devoted much attention to robust timetabling. The main focus of many of these approaches is on aperiodic timetabling. For periodic timetabling, only few approaches exist, and many of these focus on aperiodic disturbances. This, in general, means that a roll-out has to be done which can be evaluated over different scenarios. In the majority of the cases, a robust timetable is found based only on a limited number of scenarios (Kroon et al., 2008; Liebchen et al., 2010). These approaches suffer from the curse of dimensionality, i.e., they quickly become intractable if more scenarios are included. Therefore, in practical cases, only few scenarios can be considered. There is only one paper that we are aware of, that considers periodic timetabling with periodic disturbances (Goerigk, 2015), but it also deals with a finite number of scenarios and a heuristic is used for larger cases.

We add to the robust optimization literature by proposing a method to find a periodic timetable, that can be adjusted once a possible disturbance occurs. It deals with periodic disturbances, which has received only limited attention so far. Our model extends a known model, the PESP, for the fairly challenging Periodic Timetabling Problem.

6.4 Mathematical Model

In this section we present a mathematical formulation for the RPTP. We first develop the necessary notation and terminology in Section 6.4.1. We then present the mathematical formulation in Section 6.4.2, and we conclude by characterizing the set of admissible decision rules in Section 6.4.3.

6.4.1 Notation and Terminology

We consider a network of events V and activities A. The set $\bar{A} \subseteq A$ denotes the set of all activities related to the efficiency of the timetable (i.e., the set of activities related to trip, dwell, and transfer activities). Let w_{ij} denote the weight assigned to each of these activities. These weights specify the relative importance (e.g., based on the number of passengers) of each of the activities. Furthermore, let β denote the budget that is available for the timetable adjustments. These adjustments are measured at a subset $\bar{V} \subseteq V$ of the events (e.g., all arrivals at important stations, or at all stations where the trains stop).

Each activity $(i, j) \in A$ has an associated lower bound ℓ_{ij} and upper bound u_{ij}, restricting the time difference between events i and j according to (6.1). Furthermore, a subset $D \subseteq A$ of the activities is prone to disturbances. In the example of the fallen leaves, the set D contains all driving activities that are influenced by fallen leaves, thus, those that traverse regions with many trees. These disturbances manifest themselves in the lower bounds of the activities. Formally, we are given an uncertainty set Z of disturbances, which influence the subset of activities D. Let ζ_{ij} denote the effect of disturbance $\zeta \in Z$ on activity $(i, j) \in D$. It is assumed that ζ_{ij} is in $[0, 1]$. The lower bound of activity $(i, j) \in D$, given that the disturbance $\zeta \in Z$ occurs, is given by

$$\ell_{ij} + g_{ij}\zeta_{ij}. \tag{6.2}$$

Here the non-negative scalar g_{ij} regulates the impact of the disturbance on the lower bound. The found timetable should be robust against all disturbances in Z. That is, for every possible disturbance $\zeta \in Z$, there should exist timetable adjustments such that the solution is feasible for the 'disturbed' lower bounds given by (6.2). We assume that Z always includes a 'nominal disturbance', i.e., a disturbance ζ^0 with $\zeta^0_{ij} = 0$, for all $(i, j) \in D$. This assures that the timetable is feasible for the non-disturbed lower bounds.

6.4.2 Mathematical Formulation

We now formalize the RPTP. First, we introduce the following decision variables.

- π_i for all $i \in V$. The variables π_i specify the event time of event $i \in V$. Due to the periodicity of the timetable, it must hold that $\pi_i \in \{0, 1, \ldots, T - 1\}$ (see also Chapter 2).
- $y_{i\zeta}$ for all $i \in V$ and $\zeta \in Z$. The variables $y_{i\zeta}$ specify the recovery action (i.e., the shift in event time) for event $i \in V$, given scenario ζ. These variables are the essence of adjustable robustness: The values $y_{i\zeta}$ together indicate how a timetable π is adjusted in order to deal with scenario ζ.

- p_{ij} for all $(i,j) \in A$. The integer variables p_{ij} are used to model the modulo operator in the PESP constraints. In particular, it is used to model the number of cycle periods between events i and j (see also Chapter 2).

The RPTP, for a given budget parameter β, can now be expressed as follows.

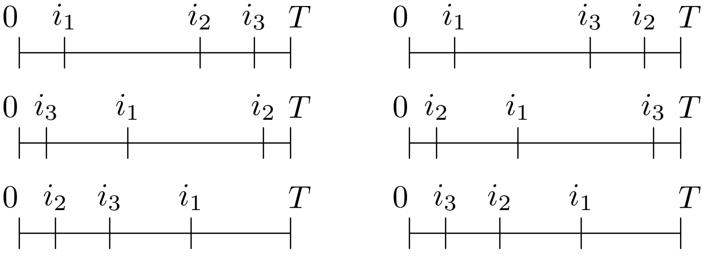
$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in \bar{A}} w_{ij}\left(\pi_j - \pi_i + Tp_{ij}\right) && && \text{(6.3a)}\\
\text{s.t.} \quad & \sum_{i\in \bar{V}} y_{i\zeta} \leq \beta && \forall\, \zeta \in Z && \text{(6.3b)}\\
& \pi_j + y_{j\zeta} - \pi_i - y_{i\zeta} + Tp_{ij} \geq \ell_{ij} + g_{ij}\zeta_{ij} && \forall\,(i,j)\in D,\ \zeta\in Z && \text{(6.3c)}\\
& \pi_j + y_{j\zeta} - \pi_i - y_{i\zeta} + Tp_{ij} \geq \ell_{ij} && \forall\,(i,j)\in A\setminus D,\ \zeta\in Z && \text{(6.3d)}\\
& \pi_j + y_{j\zeta} - \pi_i - y_{i\zeta} + Tp_{ij} \leq u_{ij} && \forall\,(i,j)\in A,\ \zeta\in Z && \text{(6.3e)}\\
& y_{i\zeta^0} = 0 && \forall\, i\in V && \text{(6.3f)}\\
& p_{ij} \in \mathbb{Z}_+ && \forall\,(i,j)\in A && \text{(6.3g)}\\
& \pi_i \in \{0,1,\ldots,T-1\} && \forall\, i\in V && \text{(6.3h)}\\
& y_{i\zeta} \in \mathbb{R}_+ && \forall\, i\in V,\ \zeta\in Z. && \text{(6.3i)}
\end{aligned}
$$

The objective (6.3a) expresses that we minimize a weighted sum of the duration of the activities. Generally, this is the minimization of the planned travel times of the passengers (i.e., we optimize the efficiency of the timetable). Constraints (6.3b) assure that the sum of the adjustments to the timetable is at most β, and (6.3c), (6.3d), and (6.3e) assure that the timetable can be made feasible for every possible scenario in Z. In particular, (6.3d) and (6.3e) represent (6.1) for the adjusted timetable. Here, $\pi_i + y_{i\zeta}$ and $\pi_j + y_{j\zeta}$ are the adjusted times when events i and j take place, given that disturbance ζ occurs. For transfer activities, for example, this ensures that the transfers are maintained for all disturbances. Constraints (6.3c) are similar to (6.3d) and take the increased lower bounds for disturbance ζ into account. Constraints (6.3f) enforce that no adjustments are made for the nominal (i.e. undisturbed) scenario, thereby assuring that (6.3a) is correctly specified. Finally, constraints (6.3g)–(6.3i) specify the domains of the decision variables. Note that (6.3) might have infinitely many constraints, depending on the uncertainty region Z.

We emphasize that we do not impose $\pi_i + y_{i\zeta} < T$ for events $i \in V$ and disturbances $\zeta \in Z$. If $\pi_i + y_{i\zeta} \geq T$ for some event i and disturbance ζ, it means that event i is actually rescheduled to $\pi_i + y_{i\zeta} - T$ if disturbance ζ occurs. Also observe that the decision variables p_{ij} are *not* adjustable. This implies that the recourse actions do not change the structure of the timetable. In the next section, we explain in detail what this means.

6.4.3 Analysis Decision Rules

Given a disturbance $\zeta \in Z$, the time when event $i \in V$ takes place is given by $\pi_i + y_{i\zeta}$. Here, the variables $y_{i\zeta}$ describe how the timetable should be adjusted based on the disturbance ζ. In contrast, the variables p_{ij} can *not* be changed in order to cope with the disturbance. In this section, we describe the consequences of the p-variables not being adjustable.



(a) The cyclic order is (i_1, i_2, i_3) (b) The cyclic order is (i_1, i_3, i_2)

Figure 6.3: Two cyclic orders of events $i_1, i_2, i_3 \in V$.

We first explain what we mean by the *cyclic* order of events. Consider three events $i_1, i_2, i_3 \in V$. There are two possible cyclic orderings of the events i_1, i_2, and i_3, as visualized in Figure 6.3. In Figure 6.3a, the cyclic order (i_1, i_2, i_3) is depicted. Here, after event i_1 has occurred, event i_2 takes place before event i_3. Note that this holds for all three time lines depicted in this figure. In particular, the cyclic orders (i_1, i_2, i_3), (i_2, i_3, i_1), and (i_3, i_1, i_2) are equal. Intuitively, this cyclic order means that after observing event i_1, we first observe event i_2 and then i_3. In Figure 6.3b, the cyclic order (i_1, i_3, i_2) is depicted. In this case, after the occurrence of event i_1, event i_3 occurs before event i_2. Observe that the cyclic order of three events is independent of the start time of a cycle and thus invariant under time shifts.

Consider now a set of three events $I = \{i_1, i_2, i_3\} \subseteq V$ that induces a clique in the PESP graph. This means that there is an activity between any two distinct events $i, i' \in I$. More formally, there exists $(i, i') \in A$ for all $i, i' \in I$ with $i \neq i'$. Three events form a clique, for example, if the events take place at the same platform or use a common part of the infrastructure. If $u_{ii'} < T$ for all these (i, i'), the nonadjustability of the p-variables implies that the cyclic order of these three events is not changed as a result of a disturbance. We now prove this claim formally.

Proposition 6.1. *Let events $i_1, i_2, i_3 \in V$ induce a clique: $(i, i') \in A$ for all $i, i' \in \{i_1, i_2, i_3\}$ with $i \neq i'$. If $u_{ii'} < T$ for all such (i, i'), then every feasible decision rule leaves the cyclic order of events i_1, i_2, and i_3 intact.*

Proof. Let a feasible solution $(\hat{\pi}, \hat{p}, \hat{y})$ for (6.3a) – (6.3i) be given. Assume, without loss of generality, that $\hat{\pi}_{i_1} \leq \hat{\pi}_{i_2} \leq \hat{\pi}_{i_3}$. It holds by definition that $\hat{\pi}_{i_3} \leq \hat{\pi}_{i_1} + T$. We will prove for every disturbance $\zeta \in Z$ that

$$\hat{\pi}_{i_1} + \hat{y}_{i_1\zeta} \leq \hat{\pi}_{i_2} + \hat{y}_{i_2\zeta} \leq \hat{\pi}_{i_3} + \hat{y}_{i_3\zeta} \leq \hat{\pi}_{i_1} + \hat{y}_{i_1\zeta} + T. \tag{6.4}$$

This implies that the cyclic order remains intact.

We first consider the inequality

$$\hat{\pi}_{i_1} + \hat{y}_{i_1\zeta} \leq \hat{\pi}_{i_2} + \hat{y}_{i_2\zeta}. \tag{6.5}$$

By our assumption that $\hat{\pi}_{i_1} \leq \hat{\pi}_{i_2}$, it holds that

$$0 \leq \hat{\pi}_{i_2} - \hat{\pi}_{i_1} = \hat{\pi}_{i_2} + \hat{y}_{i_2\zeta^0} - \hat{\pi}_{i_1} - \hat{y}_{i_1\zeta^0}. \tag{6.6}$$

We prove by contradiction that $\hat{p}_{i_1i_2} \leq 0$. In order to do so, assume that $\hat{p}_{i_1i_2} \geq 1$. We then obtain from the above that

$$\hat{\pi}_{i_2} + \hat{y}_{i_2\zeta^0} - \hat{\pi}_{i_1} - \hat{y}_{i_1\zeta^0} + \hat{p}_{i_1i_2}T \geq T. \tag{6.7}$$

Using that $u_{i_1i_2} < T$, this contradicts feasibility for ζ^0. We conclude that $\hat{p}_{i_1i_2} \leq 0$. To show that (6.5) is satisfied, assume on the contrary that (6.5) were not satisfied for a given $\zeta \in Z$. We then obtain

$$\begin{aligned}\hat{\pi}_{i_2} + \hat{y}_{i_2\zeta} - \hat{\pi}_{i_1} - \hat{y}_{i_1\zeta} + \hat{p}_{i_1i_2}T &\leq \hat{\pi}_{i_2} + \hat{y}_{i_2\zeta} - \hat{\pi}_{i_1} - \hat{y}_{i_1\zeta} \\ &< 0 \leq \ell_{i_1i_2} \leq \ell_{i_1i_2} + g_{i_1i_2}\zeta_{i_1i_2}.\end{aligned} \tag{6.8}$$

This contradicts feasibility for ζ. By contradiction, (6.5) must be satisfied. In a similar fashion, it follows from $\hat{\pi}_{i_2} \leq \hat{\pi}_{i_3}$ that

$$\hat{\pi}_{i_2} + \hat{y}_{i_2\zeta} \leq \hat{\pi}_{i_3} + \hat{y}_{i_3\zeta} \tag{6.9}$$

for all $\zeta \in Z$. It remains to prove that

$$\hat{\pi}_{i_3} + \hat{y}_{i_3\zeta} \leq \hat{\pi}_{i_1} + \hat{y}_{i_1\zeta} + T. \tag{6.10}$$

By our assumption that $\hat{\pi}_{i_3} \leq \hat{\pi}_{i_1} + T$, it holds that

$$0 \leq \hat{\pi}_{i_1} - \hat{\pi}_{i_3} + T = \hat{\pi}_{i_1} + \hat{y}_{i_1\zeta^0} - \hat{\pi}_{i_3} - \hat{y}_{i_3\zeta^0} + T. \tag{6.11}$$

It follows that $\hat{p}_{i_3i_1} \leq 1$. To see this, assume on the contrary that $\hat{p}_{i_3i_1} \geq 2$. We then obtain

$$\hat{\pi}_{i_1} + \hat{y}_{i_1\zeta^0} - \hat{\pi}_{i_3} - \hat{y}_{i_3\zeta^0} + \hat{p}_{i_3i_1}T \geq T. \tag{6.12}$$

Using that $u_{i_3i_1} < T$, this contradicts feasibility for ζ^0. We conclude that $\hat{p}_{i_3i_1} \leq 1$. Now, if (6.10) were not satisfied, then

$$\begin{aligned} \hat{\pi}_{i_1} + \hat{y}_{i_1\zeta} - \hat{\pi}_{i_3} - \hat{y}_{i_3\zeta} + \hat{p}_{i_3i_1}T &\leq \hat{\pi}_{i_1} + \hat{y}_{i_1\zeta} - \hat{\pi}_{i_3} - \hat{y}_{i_3\zeta} + T \\ &< 0 \leq \ell_{i_3i_1} \leq \ell_{i_3i_1} + g_{i_3i_1}\zeta_{i_3i_1}. \end{aligned} \tag{6.13}$$

This contradicts feasibility for ζ. By contradiction, (6.10) must be satisfied for all $\zeta \in Z$. It follows that the cyclic order of events is preserved in the solution $(\hat{\pi}, \hat{p}, \hat{y})$. $\square$

The decision rules that we obtain can easily be implemented in practice. Changing the cyclic order of events in order to deal with a disturbance might require dispatchers to set the routing of trains manually. The fact that the cyclic order of events does not change implies that no manual adjustments are necessary to operate the adjusted timetable. It also means that the timetable is adjusted by propagation and absorption of delays only.

6.5 Modelling Assumptions

In this section, we discuss two assumptions we make, which are used in our solution approach for the RPTP. We first discuss the uncertainty region in detail in Section 6.5.1. Next, we discuss a specific decision rule we use in Section 6.5.2. The combination of these leads to our solution approach in Section 6.6.

6.5.1 Parametrized Uncertainty Region Z_α

We assume the uncertainty set Z as introduced in Section 6.4 to have a specific structure: We consider a parametrized uncertainty set Z_α, where α regulates the 'severity of disturbance' in the system. This is also known as a 'budget of uncertainty' (Bertsimas et al., 2011). To give a formal definition of the uncertainty set, Z_α is given by all ζ with $0 \leq \zeta_{ij} \leq 1$, for all $(i, j) \in D$, and

$$\sum_{(i,j)\in D} \zeta_{ij} \leq \alpha. \tag{6.14}$$

Note that this includes the uncertainty regions considered in Goerigk and Schöbel (2010) and Goerigk (2015). Furthermore, it generalizes the simplex and box uncertainty regions (Ben-Tal et al., 2009b).

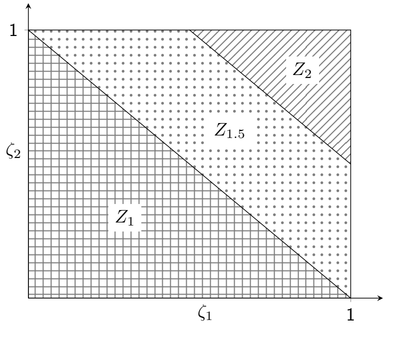


Figure 6.4: Uncertainty region Z_α for two activities. The highlighted areas indicate the uncertainty region for different values of α. Note that there is overlap between the different regions, i.e., $Z_1 \subseteq Z_{1.5} \subseteq Z_2$.

Figure 6.4 shows the uncertainty region Z_α for $|D| = 2$. The highlighted areas show the uncertainty region for different values of α. Note that the uncertainty region depends only on the ranges of the uncertain parameters. This means that part, *but not all*, of the correlation is expressed in the uncertainty region: Negatively correlated events will lead to linearly bounded uncertainty (i.e., the simplex region), whereas allowing general correlation (including perfect correlation) is captured by box uncertainty. In this sense, the control parameter α can be seen as a measure for both correlation between the disturbances and the 'size' of the disturbance: For two dimensions, $\alpha = 1$ corresponds to the simplex region, and $\alpha = 2$ corresponds to box uncertainty. This approach is more conservative than, e.g., stochastic programming, as only the support of the distribution of uncertain parameters is taken into account:

If the solution is feasible for the entire support, then it must be feasible for any distribution over this support.

We conclude by emphasizing that the scenarios ζ are real-valued vectors from the unit hyper-cube. Thus a typical ζ has strictly positive components only. In particular, the sets Z_1 and Z_2 are not confined to vectors with at most one or two non-zero components, respectively.

6.5.2 Linear Decision Rule

For the large-scale problems encountered in practice, it is generally impossible to solve (6.3) for arbitrary decision rules. We therefore restrict our attention to linear decision rules, in order to simplify (6.3). This can be seen as a heuristic approach: By only considering linear decision rules, we allow less flexibility when dealing with disturbances. Other, more complex decision rules are not considered. As a result, if we obtain a solution to (6.3), its objective value is an upper bound on the optimal solution value of (6.3).

In particular, we assume that the decision rule $y_{i\zeta}$ can be written as a linear function of ζ. That is, we introduce a decision variable $\delta^i_{kl} \in \mathbb{R}_+$ for each $i \in V$ and $(k, l) \in D$, and assume that $y_{i\zeta}$ can be written as

$$y_{i\zeta} = \sum_{(k,l)\in D} \delta^i_{kl}\zeta_{kl}, \tag{6.15}$$

for all $i \in V$. The functional form (6.15) is generally known as a *linear decision rule.* The linear decision rule assumes that the recovery action is a linear combination (with weights $\delta^i_{k\ell}$) of the disturbance vectors ζ. The weights express the extent by which a disturbance of an arc influences the realised event times: Each unit of disturbance of arc (k, ℓ) will require a recovery time of $\delta^i_{k\ell}$ at event i.

It is important to note that, although the linear decision rule is generally not an optimal policy, it is *always* an admissible recourse function. That is, every feasible linear decision rule is a feasible recourse function.

The resulting model is obtained by substituting the linear decision rule into the original model. That is, we use (6.15) to reformulate constraints (6.3b)–(6.3e). After rearranging terms, we obtain the new set of constraints

$$\sum_{i\in\bar{V}}\sum_{(k,l)\in D}\delta^i_{kl}\zeta_{kl} \leq \beta \qquad \forall\,\zeta\in Z \tag{6.16a}$$

$$\pi_j - \pi_i + Tp_{ij} \geq \ell_{ij} + g_{ij}\zeta_{ij} + \sum_{(k,l)\in D}\left(\delta^i_{kl} - \delta^j_{kl}\right)\zeta_{kl} \qquad \forall\,(i,j)\in D,\ \zeta\in Z \tag{6.16b}$$

$$\pi_j - \pi_i + Tp_{ij} \geq \ell_{ij} + \sum_{(k,l)\in D}\left(\delta^i_{kl} - \delta^j_{kl}\right)\zeta_{kl} \qquad \forall\,(i,j)\in A\setminus D,\ \zeta\in Z \tag{6.16c}$$

$$\pi_j - \pi_i + Tp_{ij} \leq u_{ij} + \sum_{(k,l)\in D}\left(\delta^i_{kl} - \delta^j_{kl}\right)\zeta_{kl} \qquad \forall\,(i,j)\in A,\ \zeta\in Z. \tag{6.16d}$$

The new model is obtained by replacing (6.3b)–(6.3e) with (6.16a)–(6.16d). Note that (6.3f) is trivially satisfied for (6.15).

6.6 Solution Approach

The key issue when modelling robust optimization problems, is dealing with the intractability of the original model. In the case of linear uncertainty over a polyhedral region, as introduced in Section 6.5, two methods are most common: The constraint is reformulated as a finite system of inequalities using LP duality (see e.g., Ben-Tal et al., 2009b), or constraints are separated iteratively using a cutting-plane method (also known as the adversarial approach). Bertsimas et al. (2016) give a detailed comparison of the reformulation technique and the cutting-plane method, where they show there is no clearly dominating method. We found the best-suited method for the RPTP to be a combination of the two. These two techniques that we use to solve the original model are outlined in this section, as well as a heuristic approach to find a benchmark solution.

6.6.1 Reformulation

In our approach, we only reformulate (6.16a). That means, (6.16a) is replaced by the system of inequalities

$$
\begin{aligned}
&\alpha\lambda + \sum_{(i,j)\in D} \mu_{ij} \leq \beta && && (6.17a)\\
&\lambda + \mu_{kl} \geq \sum_{i\in\bar{V}} \delta^i_{kl} && \forall\,(k,l)\in D && (6.17b)\\
&\mu_{ij} \in \mathbb{R}_+ && \forall\,(i,j)\in D && (6.17c)\\
&\lambda \in \mathbb{R}_+. && && (6.17d)
\end{aligned}
$$

The decision on whether to reformulate a constraint or not, can be made based on the uncertainty set. In the case of Z_α, the reformulation of one constraint requires a total of $|D|$ new constraints and $|D|+1$ new variables (see Appendix 6.A for a detailed overview of the reformulation). For the considered instances, D has roughly the same size as A. This implies that reformulating (6.16b), (6.16c), and (6.16d) results in roughly $|A|^2$ additional constraints. As a result, reformulating all of the constraints (6.16b)–(6.16d) might perform well for smaller instances, but would lead to an intractable model (from a practical point of view) for the larger instances. For this reason, we have used the cutting-plane approach for these sets of constraints.

6.6.2 Cutting-Plane Method

In order to discuss the cutting-plane method in more detail, we consider the formulation

$$
\begin{aligned}
\min \quad & \sum_{(i,j)\in\bar{A}} w_{ij}\left(\pi_j - \pi_i + Tp_{ij}\right) && && (6.18a)\\
\text{s.t.} \quad & \alpha\lambda + \sum_{(i,j)\in D} \mu_{ij} \leq \beta && && (6.18b)\\
& \lambda + \mu_{kl} \geq \sum_{i\in\bar{V}} \delta^i_{kl} && \forall\,(k,l)\in D && (6.18c)
\end{aligned}
$$

$$\pi_j - \pi_i + Tp_{ij} \geq \ell_{ij} + g_{ij}\zeta_{ij} + \sum_{(k,l)\in D} \left(\delta^i_{kl} - \delta^j_{kl}\right)\zeta_{kl} \quad \forall\,(i,j) \in D,\ \zeta \in L^{ij}_\alpha \tag{6.18d}$$

$$\pi_j - \pi_i + Tp_{ij} \geq \ell_{ij} + \sum_{(k,l)\in D} \left(\delta^i_{kl} - \delta^j_{kl}\right)\zeta_{kl} \quad \forall\,(i,j) \in A \setminus D,\ \zeta \in L^{ij}_\alpha \tag{6.18e}$$

$$\pi_j - \pi_i + Tp_{ij} \leq u_{ij} + \sum_{(k,l)\in D} \left(\delta^i_{kl} - \delta^j_{kl}\right)\zeta_{kl} \quad \forall\,(i,j) \in A,\ \zeta \in U^{ij}_\alpha \tag{6.18f}$$

$$p_{ij} \in \mathbb{Z}_+ \quad \forall\,(i,j) \in A \tag{6.18g}$$

$$\pi_i \in \{0, 1, \ldots, T-1\} \quad \forall\, i \in V \tag{6.18h}$$

$$\delta^i_{kl} \in \mathbb{R}_+ \quad \forall\, i \in V, (k,l) \in D \tag{6.18i}$$

$$\mu_{ij} \in \mathbb{R}_+ \quad \forall (i,j) \in D \tag{6.18j}$$

$$\lambda \in \mathbb{R}_+, \tag{6.18k}$$

where $L^{ij}_\alpha, U^{ij}_\alpha \subseteq Z_\alpha$ for all $(i,j) \in A$. The constraints (6.18d), (6.18e), and (6.18f) are enforced for only a subset of the scenarios. Initially, we have $L^{ij}_\alpha = U^{ij}_\alpha = \{\zeta^0\}$, and in each cutting-plane iteration, we extend these sets. For convenience in the description of the algorithm, we only consider (6.18d), noting that the methodology for (6.18e) and (6.18f) is identical.

Consider some fixed $(i,j) \in D$. Whenever a solution $(\hat{\pi}, \hat{p}, \hat{\delta})$ is found for the reduced set of constraints, we check whether (6.18d) is violated for some $\zeta \in Z_\alpha \setminus L^{ij}_\alpha$. If this is the case, then ζ is added to L^{ij}_α, and the model is solved again. This procedure is repeated until no more violations can be found. At that point, the found solution is feasible for all possible disturbances. Note that this algorithm is equivalent to applying Benders Decomposition on the *fully* re-formulated model.

To check whether a constraint (6.18d) is violated for some $\zeta \in Z_\alpha$, we determine a disturbance $\hat{\zeta} \in Z_\alpha$, given by

$$\hat{\zeta} = \arg\max_{\zeta \in Z_\alpha} \left\{ g_{ij}\zeta_{ij} + \sum_{(k,l)\in D} \left(\hat{\delta}^i_{kl} - \hat{\delta}^j_{kl}\right)\zeta_{kl} \right\}, \tag{6.19}$$

and add $\hat{\zeta}$ to L^{ij}_α whenever

$$\hat{\pi}_j - \hat{\pi}_i + T\hat{p}_{ij} < \ell_{ij} + g_{ij}\hat{\zeta}_{ij} + \sum_{(k,l)\in D} \left(\hat{\delta}^i_{kl} - \hat{\delta}^j_{kl}\right)\hat{\zeta}_{kl}. \tag{6.20}$$

Note that if the solution $(\hat{\pi}, \hat{p}, \hat{\delta})$ is feasible with respect to $\hat{\zeta}$, it is for all $\zeta \in Z_\alpha$.

An optimal solution for (6.19) can be determined efficiently as follows. For a given $(i, j) \in D$, the maximization problem (6.19) boils down to a linear knapsack problem over the activities D, with capacity α and cost coefficients $c_{ij} = g_{ij} + \hat{\delta}^i_{ij} - \hat{\delta}^j_{ij}$ for activity (i, j), and $c_{kl} = \hat{\delta}^i_{kl} - \hat{\delta}^j_{kl}$ for all activities $(k, l) \in D \backslash \{(i, j)\}$. For this problem, an optimal solution can be found efficiently by determining the $\lceil \alpha \rceil$ activities with largest nonnegative cost coefficient. As noted in Bertsimas et al. (2016), this can be done in $\mathcal{O}(|D| + \alpha \log \alpha)$ time, using an efficient partial sorting algorithm. Gorissen et al. (2015) note that it is not uncommon that the cutting-plane approach converges to optimality in only a small number of iterations.

6.6.3 Benchmark Solution

One way of avoiding delays in a timetable, is to absorb them as soon as possible. Based on this principle, we generate a benchmark solution by requiring that each activity has enough time supplements to absorb any possible disturbance on that activity. That is, we require that

$$\pi_j - \pi_i + Tp_{ij} \geq \ell_{ij} + \max_{\zeta \in Z_\alpha} g_{ij} \zeta_{ij}. \tag{6.21}$$

This imposes additional restrictions on the durations of activities and leads to solving a standard PESP model, where the efficiency is to be optimized. As an example, assume that all trip activities $(i, j) \in A$ can require up to one additional minute due to a disturbance. In that case, $g_{ij} = 1$ for all trip activities $(i, j) \in A$, and we require that the minimum duration of these activities is at least $\ell_{ij} + 1$. The only thing to be optimized is the additional duration of the trip and dwell time activities.

If a solution can be found, this solution is also feasible for (6.3), with a trivial linear decision rule (i.e., δ equals the zero-function: $\delta^i_{k\ell} = 0$ for all $i \in V$, $(k, \ell) \in D$), and no recovery budget. Such a benchmark solution may not exists if the worst-case disturbances are higher than what the restrictions of the underlying timetabling problem can handle. We note that our case studies never reach such a high level, therefore we always find benchmark solutions.

The benchmark solution can be used as a starting solution to be improved upon in the further optimization. Furthermore, it implements a simple and natural idea to cope with the uncertainty, therefore it is a suitable comparison point for the outcome of our robust optimization models. The benchmark solution captures the spirit of current railway practices. Most operators follow the UIC recommendations by setting the time supplements uniformly to 6-8% of the technically minimal travel times. Our benchmark solution slightly deviates from these recommendations because applying them strictly would lead to an infeasible timetabling problem.

To summarize our approach, we propose a solution method for the RPTP that builds upon a linear decision rule. Combining this with the uncertainty region Z_α, this decision rule gives an upper bound for the original problem in (6.3). The 'linearized' model is solved by combining reformulation with cutting planes: Constraints (6.18b) and (6.18c) are the dual reformulation of the budget constraint (6.16a), and Constraints (6.18d) – (6.18f) represent (6.16b)–(6.16d) for only a subset of the disturbances. A simple heuristic is used to generate a feasible solution. Next, the model is solved iteratively, identifying disturbances corresponding to violated cuts in each iteration, until all constraints are satisfied, and hence an optimal solution to (6.18) has been found.

6.7 Computational Experiments

We test the adjustable robust approach on real-life instances of Netherlands Railways (NS), the largest operator of passenger trains in the Netherlands. The instances feature the trains in the so-called 'Kop van Noord-Holland' region in a 1-hour period. Note that Kroon et al. (2008) report their computational results for the same region, although the approach and aim of that paper is quite different from ours.

The nominal periodic timetabling problem is formulated by using standard methods from the PESP literature, including various pre-processing steps and using cycle periodicity formulation inequalities (cf. Liebchen and Peeters (2009) and Chapter 2).

We study the relation between the three aspects of the timetable: recovery budget (β), robustness and efficiency. Robustness is expressed by the parameter α of the uncertainty set, ranging between zero and the number of disturbable arcs $|D|$. Note that D is considered input. In what follows we measure a solution's *efficiency loss* as the total amount of time supplements needed, defined as

$$\sum_{(i,j)\in\bar{A}} \pi_j - \pi_i + Tp_{ij} - \ell_{ij}. \tag{6.22}$$

That is, we consider all trip and dwell activities, and we add up the difference between the planned activity time and the minimal activity time. The best possible efficiency is achieved if the total time supplement is zero.

We consider two instances. Case 1 is a study on a medium-sized, yet non-trivial, sub-network of the region. It includes 20 services connecting 20 stations (Figure 6.5a). Case 1 has a PESP model with 589 nodes and 1668 activities. We assume that disturbances can occur on all dwell activities and most of the trip activities related to station Alkmaar (Amr): A trip activity can require up to 1 minute more time than in the nominal problem, a dwell activity can require up to 2 minutes more; there are 22 disturbable activities in total. Thus, $g_{ij} = 1$ if $(i,j) \in D$ represents a trip arriving at or departing from Alkmaar, and $g_{ij} = 2$ if $(i,j) \in D$ represents a dwell activity at Alkmaar. The robustness parameter α ranges from 0 to 22. Case 1 has a feasible solution to our model with maximal efficiency where the total travel time of the trains is $\sum_{(i,j)\in\bar{A}} \ell_{ij} = 622$.

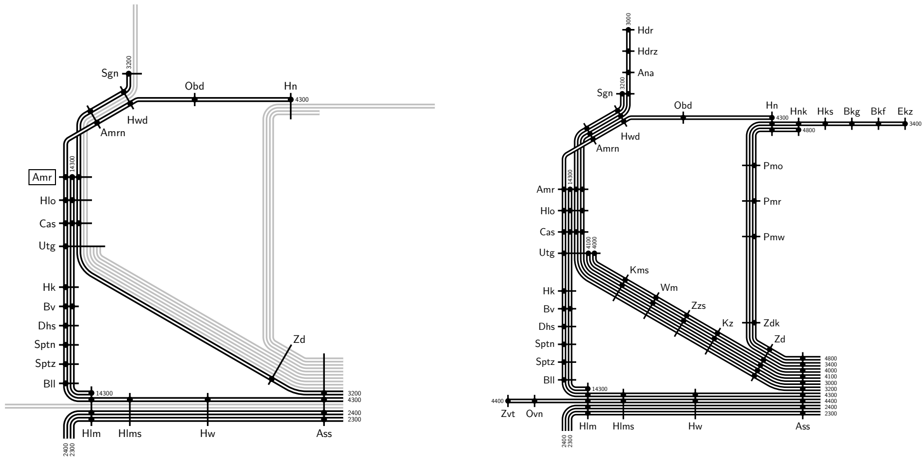
Case 2 is a large study that includes all 44 services of the region, connecting 38 stations (Figure 6.5b). It covers about 8% of all passenger services of NS. Disturbances of up to 1 minute can occur on all trip activities. Here, the set $D \subseteq A$ contains all driving activities and $g_{ij} = 1$ for all $(i,j) \in D$. The PESP model has 1321 nodes and 4862 activities of which 276 activities are disturbable. Case 2 has a feasible solution with maximal efficiency, too, where the total travel time of the trains is $\sum_{(i,j)\in\bar{A}} \ell_{ij} = 1406$.

We emphasize that our case studies are non-trivial timetabling problems. The nominal optimization problem itself takes 6.2 seconds for Case 1 and about 5 minutes for Case 2. In fact, Case 2 approaches the limits of what periodic timetable optimization models can handle in a non-robust setting. Adding just a few services to Case 2 raises the typical computation times dramatically: Near-optimality cannot be reached after several hours of CPU times, the optimality gaps often remain above 90%.

Our computations are carried out on a machine with an Intel Xeon E5-2650 v2 2.60Ghz processor and with 64 GB RAM. The integer programs are solved by Cplex 12.8.0 with 15 parallel threads (IBM, 2017). The cutting-planes are implemented via a Lazy Constraint Callback.

6.7.1 Kop van Noord Holland: Case 1

In this section, we discuss the computational results for Case 1. We first investigate the relation between the robustness and the recovery budget for a maximally efficient timetable, and then the trade-off between the efficiency, robustness, and recovery budget.



(a) The network of case 1.

(b) The network of case 2.

Figure 6.5: The railway networks considered in the case study.

Relation between robustness and recovery budget

Our first experiments aim at finding the relevant values of the recovery budget β. For each value $\alpha \in \{0, \ldots, 22\}$ we adjust our robust optimization model to determine the smallest possible recovery budget under the assumption that all time supplements are zero (that is, that the solution is maximally efficient).

In order to speed up these calculations, we observe that an optimal solution for a higher value of α is always a feasible solution for a lower value of α. Therefore we solved the problem in a decreasing order of α, using the best solution found so far as a starting solution by means of Cplex's MIP start feature.

The solid line in Figure 6.6 indicates the optimal budget for each value of α. Mind that $\alpha = 0$ is the nominal scenario (i.e., there is no disturbance), and $\alpha = 22$ is the scenario where each disturbable activity is indeed maximally disturbed. It turns out that each α admits a maximally efficient timetable. Therefore the obtained values indicate the largest amount of useful recovery budget for each α. We note that the recovery budget can rise to 360 minutes which is more than 57% of the total nominal travel times.

For each value of α, we have obtained a timetable that is maximally efficient and minimizes the required budget β. We now evaluate three of these timetables for all possible robustness values α. We consider the timetable found without considering

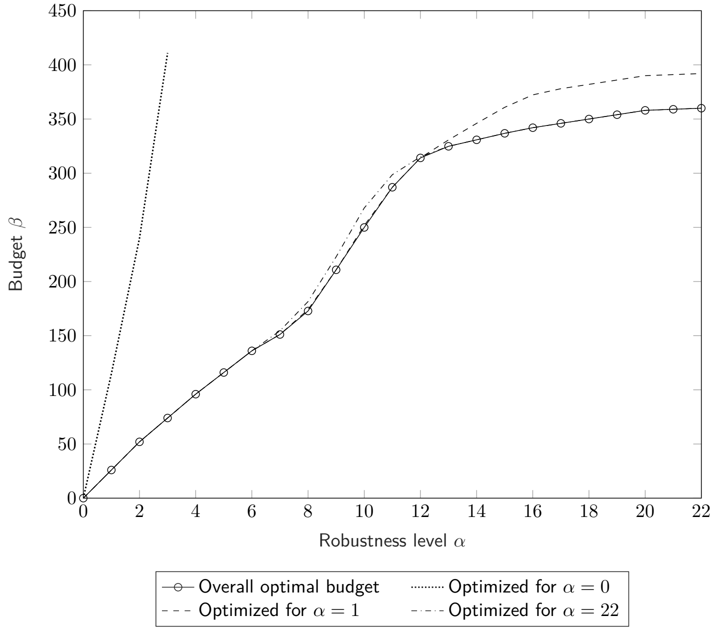


Figure 6.6: Trade-off between robustness and recovery budget. *Solid line*: The optimal budget for each α. *Other lines*: Maximally efficient timetables, optimized for one specific value of α and evaluated under various levels of disturbances.

robustness ($\alpha = 0$) and those found with a low and a high uncertainty level ($\alpha = 1$ and $\alpha = 22$, respectively). The dotted line in Figure 6.6 shows the budget needed to recover the timetable optimized for $\alpha = 0$. The recovery budget of this particular solution starts growing rapidly with α, and no recovery is possible for $\alpha > 3$.

The dashed and dash-dotted line in the figure show the optimal recovery costs of two timetables that are optimized for a low and a high level of robustness, respectively. The benefit of taking a suitable level of robustness into account is clearly visible in the figure. Optimizing a timetable for a small level of robustness leads to recovery costs for high values of α that are higher than the optimum. For $\alpha = 22$, the recovery costs are 392 which is roughly 9% from the optimal costs (360). If a high level of uncertainty is taken into account, the recovery costs are close to the optimum. Only for medium values of α the costs are slightly higher than the optimum. It is interesting that for none of the timetables that we have found the recovery budget equals the optimal budget for every value of α.

In this figure we see that maximally efficient solutions give rise to very different recovery costs. The result demonstrates our method's ability to find the most robust solution among all efficient solutions.

Trade-off between efficiency, robustness, and recovery budget

In the second set of experiments we apply the adjustable robust optimization model as described in Sections 6.4–6.6: We seek the lowest efficiency loss for given combinations of robustness level α and recovery budget β. We set a time limit of 4 CPU hours for each optimization run. Again, we carry out the computations, for each given β, in a decreasing order of α, and we use the previous round's solution as a feasible starting solution.

Our results for $\alpha \in \{0, \ldots, 22\}$ and $\beta \in \{0, 10, 20, 30, 50, 75, 100, 150\}$ are summarised in Figure 6.7. The vertical axes indicate the efficiency loss, both in absolute and relative terms. The left axis shows the amount of added time supplements, as defined in (6.22). The right axis displays the relative objective value increase with respect to a maximally efficient solution. Not all combinations are solved to optimality within the time limit; the dashed lines indicate the best found lower bounds. In fact, the computation times vary heavily, both for finding good solutions, and for proving optimality.

Figure 6.7 shows the intuitively clear relation between the recovery budget and the steepness of the curves. The time supplements trade delay propagation for delay absorption: A high budget allows for more propagation, and hence less time supplements are needed in the nominal timetable. The efficiency rapidly decreases for small β as the robustness level α increases, whereas large values of β admit a milder efficiency loss.

The curves for the individual budget values β reach their maximum height at surprisingly low values of α; this effect becomes increasingly stronger as β decreases. For example, the efficiency loss for $\beta = 50$ is identical for each $\alpha \geq 13$. The uncertainty parameter $\alpha = 13$ requires such an amount of time supplements that is sufficient for all scenarios, even with $\alpha = 22$. That is, there exists an optimal solution for $\alpha = 13$ which is optimal for $\alpha = 22$, as well.

More detailed information about the results can be gained from Figure 6.8. Here, heatmaps are shown for the solutions found with $\alpha \in \{1, 13, 22\}$ and $\beta \in \{0, 30, 100\}$.

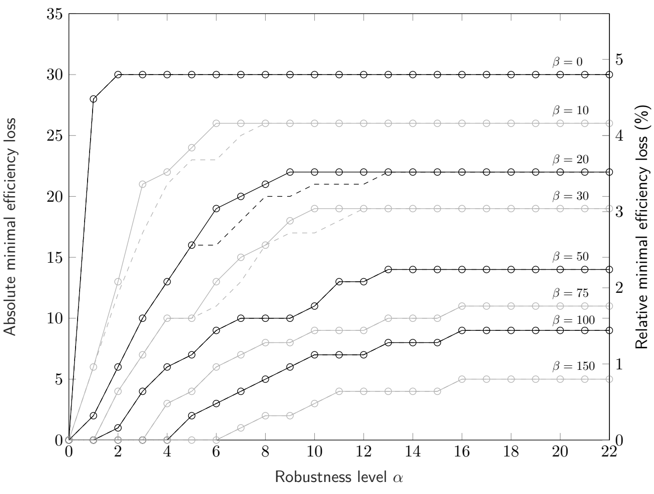


Figure 6.7: Trade-off between efficiency and robustness for various recovery budgets β.

The darker the color on the segments between stations, the more time supplements are added here.

Clearly, most of the time supplements are added around the station of Alkmaar (Amr), as is expected. Some are added at the tracks right next to Alkmaar, and some also a little bit further away, even though the disturbance could not directly affect these tracks. Secondly, we see that time supplements are added if there is no recovery budget available ($\beta = 0$) and that the solution hardly changes if more uncertainty is considered. If more budget is available, we see that a higher level of uncertainty is needed before time supplements are necessary. This confirms that our model indeed adds time supplements at reasonable locations and first uses the available recovery budget whenever possible.

The main benefit of our approach is the mere fact that it enables one to carry out detailed studies on the trade-off between robustness, efficiency and recovery budget. The decision makers gain a valuable insight into the consequences of sacrificing some efficiency for the sake of improved robustness.

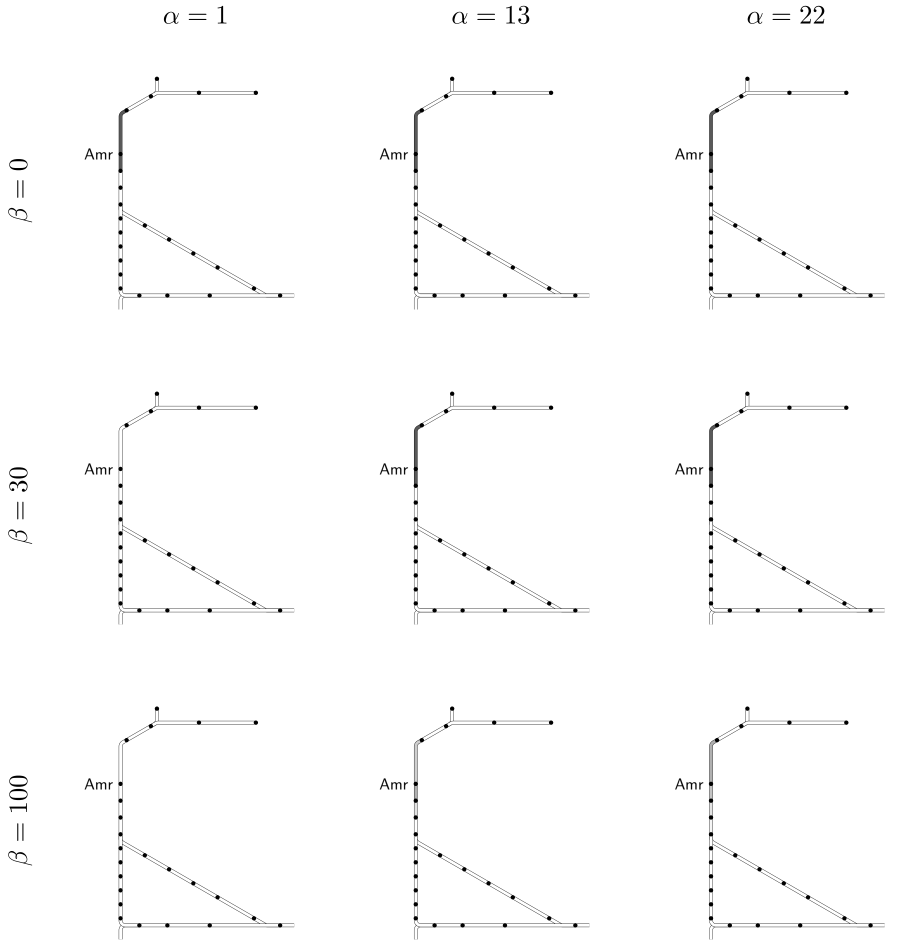


Figure 6.8: Graphical results for Case 1

6.7.2 Kop van Noord Holland: Case 2

Case 2 contains all services of the 'Kop van Noord Holland' region, and disturbances are allowed on each trip activity. This is a non-trivial study; the nominal timetabling problem, without any robustness consideration, needs about 5 minutes to be solved to optimality. We want to explore the capabilities and limitations of our adjustable robust approach by considering an underlying problem that is close to the limitations of the non-robust model.

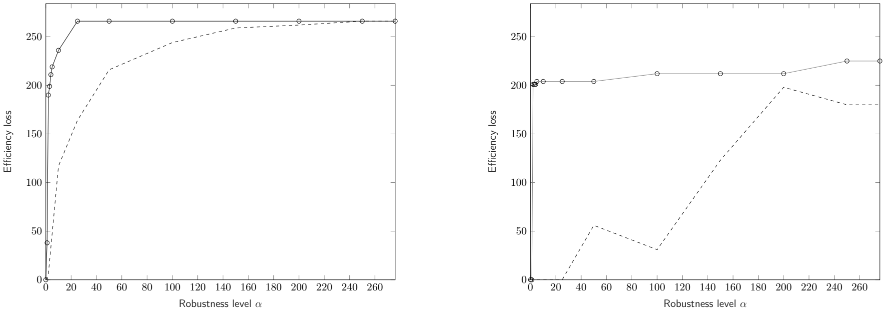
We consider $\alpha \in \{1, \ldots, 5, 10, 25, 50, 100, 150, 200, 250, 276\}$ as uncertainty parameters, recovery budgets $\beta \in \{10, 100\}$, and we minimize the efficiency loss for each combination of these parameters. Recall that $\alpha = 0$ is the nominal scenario, and $\alpha = 276$ is the scenario where each disturbable activity takes a disturbance of 1 minute. Each optimization run has a time limit of 5 hours.

The budgets of 10 and 100 minutes comprise 0.71% and 7.1% of the total train travel time of an optimally efficient solution, respectively. For a comparison, the timetables of NS feature travel times where the planned time supplement is 7-8% of the technically minimal travel times. We report our computational results with a wider range of budget values in Appendix 6.B.

We compare the model's outcome to the benchmark solution described in Section 6.6.3. This benchmark timetable adds (at least) 1 minute time supplement to each disturbable activity; its efficiency loss is 276 which is 19.6% of 1406, the total train travel time of an optimally efficient solution. We take various measures to reduce the solution time of the adjustable robust approach. We use the benchmark timetable as a feasible starting solution. Also, similarly to Section 6.7.1, we start the computations with $\alpha = 276$ and we proceed toward $\alpha = 1$ using the solutions of the previous rounds as a starting point.

Table 6.1 and Figure 6.9 summarize our results with the robust optimization approach using a linear decision rule in terms of the best objective value and the best lower bound of the MIP models. In addition, we report an upper bound on the optimality gap and the improvement upon the benchmark solution.

Our model is able to find feasible solutions that significantly improve the benchmark timetable. For $\alpha \geq 2$, our solutions have 3.6 to 31.2% less efficiency loss. The improvement is limited under the tight recovery budget $\beta = 10$. However, the more realistic budget $\beta = 100$ allows for an improvement of 18.5% even under the highest disturbance level, and for slightly larger improvements for lower disturbance levels.



(a) The case $\beta = 10$.

(b) The case $\beta = 100$.

Figure 6.9: Trade-off between efficiency and robustness in Case 2 for various recovery budgets β. *Solid line:* objective value; *dashed line:* lower bound.

For $\alpha = 1$ we find dramatic improvements; the choice of $\beta = 100$ admits a maximally efficient robust timetable. We note that the cases with $1 < \alpha < 275$ are intractable by scenario-based robust optimization approaches. Indeed, the uncertainty regions have, for integer α, at least $\binom{276}{\alpha}$ vertices; this results in more than 10^{10} scenarios for $\alpha = 5$, and more than 10^{81} scenarios for $\alpha = 150$.

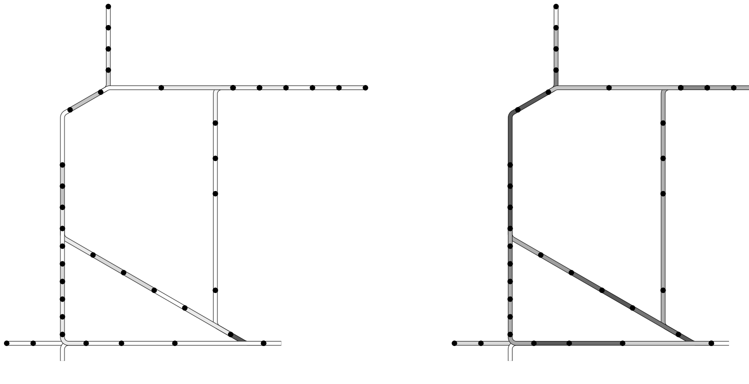
In addition, our model provides meaningful lower bounds on the efficiency loss (see again Figure 6.9). They provide a valuable insight to decision makers for evaluating the candidate timetables. The proven optimality gap is remarkably low for $\alpha \geq 100$ (with a budget of 10) and for $\alpha \geq 200$ (with a budget of 100). It turns out that the small improvement percentage 3.6% for $\beta = 10$ (see Table 6.1) is often paired with a lower bound that prevents substantially better solutions. We do admit, though, that the optimality gaps for $\alpha < 100$ are high. This is also reflected in the fact that the solutions for $\alpha = 2$ and $\alpha = 3$ have a lower objective value with $\beta = 10$ than with $\beta = 100$, even though a higher budget allows for a better solution. The superior solutions for $\alpha = 1$ suggest that the large gaps are likely caused by not finding near optimal solutions. For situations with moderate disturbance, better solutions might be found with a different solution methodology.

Figure 6.10 displays the heatmaps of the solutions that are found with two different combinations of α and β. In the case with small disturbances, we see that the majority of the time supplements are added at the west side of the network and at the tracks in the diagonal direction. The reason for this is that the majority of the trains use these tracks and thus a correct execution of the timetable is very important here. Furthermore, time supplements are added on the very north of the network, as

	$\beta = 10$				$\beta = 100$			
α	UB	*LB*	Opt.	Impr.	UB	*LB*	Opt.	Impr.
1	38	*2*	95%	86.2%	0	*0*	0%	100.0%
2	190	*6*	97%	31.2%	201	*0*	100%	27.2%
3	199	*13*	94%	27.9%	201	*0*	100%	27.2%
4	211	*33*	85%	23.6%	201	*0*	100%	27.2%
5	219	*40*	82%	20.7%	204	*0*	100%	26.1%
10	236	*117*	50%	14.5%	204	*0*	100%	26.1%
25	266	*164*	38%	3.6%	204	*0*	100%	26.1%
50	266	*216*	19%	3.6%	204	*56*	72%	26.1%
100	266	*244*	8%	3.6%	212	*31*	85%	23.2%
150	266	*259*	3%	3.6%	212	*123*	42%	23.2%
200	266	*262*	1%	3.6%	212	*198*	6%	23.2%
250	266	*266*	0%	3.6%	225	*180*	20%	18.5%
276	266	*266*	0%	3.6%	225	*180*	20%	18.5%

Table 6.1: Results for the 'Kop van Noord Holland' – Case 2.

there is only single track available here. Hence, the timetable is very sensitive to any disturbance and delays easily propagate to other trains using the same tracks. With an increased recovery budget and more uncertainty, time supplements are added throughout the network, but the largest increase is seen on the southern tracks.



(a) $\alpha = 1$, $\beta = 10$.

(b) $\alpha = 150$, $\beta = 100$.

Figure 6.10: Graphical results for Case 2

6.8 Conclusion

In this chapter, we proposed an adjustable robust extension of the Periodic Timetabling Problem (PTP): the Robust Periodic Timetabling Problem (RPTP). The RPTP is a challenging optimization problem, as the non-robust PTP is already known to be computationally difficult. We developed a solution method for the RPTP for a class of uncertainty regions. This class relates closely to uncertainty regions known in the robust optimization literature, and allows for a natural quantification of the robustness of a timetable.

The developed solution method consists of two parts. We first reduce the model size by assuming a linear decision rule, i.e., the recourse actions depend linearly on the disturbances. The size of the resulting model is only a fraction of the original model's size, albeit at a loss of freedom in the decision rule. As a result, this approach can be considered to be heuristic. The new model is solved using dual reformulation techniques and cutting-plane methods, both well-established techniques in the robust optimization literature.

We showed that the RPTP can be solved for practical-sized instances by applying the solution method to instances of Netherlands Railways (NS). In particular, we studied the interplay between the severity of the disturbances, the total amount of timetable adjustments, and the efficiency of the undisturbed timetable. Our experiments show that the RPTP remains computationally tractable: Good solutions for challenging instances can be found in reasonable time. We also saw, however, that for large scale instances with moderate disturbance there remains room for improvement.

Interesting further research would include the analysis of different, more complex, decision rules. In particular, it would be interesting to quantify the possible efficiency loss due to a linear decision rule. Note, however, that this would most likely greatly complicate any possible solution method. Furthermore, it would be interesting to consider a different class of uncertainty regions, based on, e.g., historical data.

Further research is needed to refine the solution methodology in order to find better solutions in situations with moderate disturbances.

Acknowledgments. We thank Prof. Dick den Hertog for his valuable comments and suggestions, which were helpful in improving this work.

Appendix

6.A Robust Counterpart

We now show how to derive the robust counterpart of (6.16a). We first reformulate the general constraint

$$a + \mathbf{b}^\top \boldsymbol{\zeta} \leq u \qquad \forall \boldsymbol{\zeta} \in Z, \tag{6.23}$$

with $a \in \mathbb{R}$, and $\mathbf{b}$ and $\boldsymbol{\zeta} \in \mathbb{R}^d$, following the methodology of Ben-Tal et al. (2009b). We assume that the uncertainty set Z is given by

$$Z = \{\boldsymbol{\zeta} \in \mathbb{R}^d : \mathbf{D}\boldsymbol{\zeta} \leq \mathbf{d}\}, \tag{6.24}$$

for some $\mathbf{D} \in \mathbb{R}^{p \times d}$ and $\mathbf{d} \in \mathbb{R}^p$. Constraint (6.23) can equivalently be written as

$$a + \max_{\boldsymbol{\zeta} \in \mathbb{R}^d} \{\mathbf{b}^\top \boldsymbol{\zeta} : \mathbf{D}\boldsymbol{\zeta} \leq \mathbf{d}\} \leq u. \tag{6.25}$$

By applying LP-duality, this transforms to

$$a + \min_{\boldsymbol{\theta} \in \mathbb{R}^p_+} \{\mathbf{d}^\top \boldsymbol{\theta} : \mathbf{D}^\top \boldsymbol{\theta} = \mathbf{b}\} \leq u. \tag{6.26}$$

It is easily seen that (6.26) can only be satisfied if we can find a $\boldsymbol{\theta}$ such that

$$a + \mathbf{d}^\top \boldsymbol{\theta} \leq u \tag{6.27a}$$

$$\mathbf{D}^\top \boldsymbol{\theta} = \mathbf{b} \tag{6.27b}$$

$$\boldsymbol{\theta} \in \mathbb{R}^p_+. \tag{6.27c}$$

The system of inequalities (6.27) is known as the *tractable reformulation* of (6.23).

We now consider the special case where $Z = Z_\alpha$. This implies that $\mathbf{D}^\top = [\mathbf{1}, -\mathbf{I}, \mathbf{I}]$, with $\mathbf{I}$ the identity matrix of dimension d, and $\mathbf{d}^\top = [\alpha, \mathbf{0}^\top, \mathbf{1}^\top]$. Hence, if we write $\boldsymbol{\theta}^\top = [\lambda, \boldsymbol{\phi}^\top, \boldsymbol{\mu}^\top]$, where $\lambda \in \mathbb{R}_+$ and $\boldsymbol{\phi}, \boldsymbol{\mu} \in \mathbb{R}^d_+$, and use the definition of $\mathbf{D}$ and $\mathbf{d}$, we obtain the constraints

$$a + \alpha\lambda + \mathbf{1}^\top \boldsymbol{\mu} \leq u \tag{6.28a}$$

$$\lambda\mathbf{1} - \boldsymbol{\phi} + \boldsymbol{\mu} = \mathbf{b} \tag{6.28b}$$

$$\lambda \in \mathbb{R}_+, \quad \boldsymbol{\phi}, \boldsymbol{\mu} \in \mathbb{R}^d_+. \tag{6.28c}$$

The $\boldsymbol{\phi}$ variables function solely as slack variables, and hence the model can be written in the more compact form

$$a + \alpha\lambda + \mathbf{1}^\mathsf{T}\boldsymbol{\mu} \leq u \tag{6.29a}$$

$$\lambda\mathbf{1} + \boldsymbol{\mu} \geq \mathbf{b} \tag{6.29b}$$

$$\lambda \in \mathbb{R}_+ \tag{6.29c}$$

$$\boldsymbol{\mu} \in \mathbb{R}_+^d. \tag{6.29d}$$

6.B Additional Numerical Results for Case 2

Table 6.2 summarizes our computational results on Case 2, using the recovery budget values $\{5, 10, 20, 50, 100, 200\}$.

	$\beta = 5$		$\beta = 10$		$\beta = 20$		$\beta = 50$		$\beta = 100$		$\beta = 200$	
α	UB	*LB*	UB	*LB*	UB	*LB*	UB	*LB*	UB	*LB*	UB	*LB*
1	103	*6*	38	*2*	11	*0*	0	*0*	0	*0*	2	*0*
2	213	*37*	190	*6*	202	*0*	213	*0*	201	*0*	117	*0*
3	228	*53*	199	*13*	202	*0*	213	*0*	201	*0*	178	*0*
4	236	*64*	211	*33*	211	*1*	213	*0*	201	*0*	178	*0*
5	237	*123*	219	*40*	212	*10*	213	*0*	204	*0*	178	*0*
10	271	*143*	236	*117*	220	*65*	213	*1*	204	*0*	178	*0*
25	271	*185*	266	*164*	256	*49*	213	*55*	204	*0*	181	*0*
50	271	*249*	266	*216*	256	*166*	232	*55*	204	*56*	183	*0*
100	271	*263*	266	*244*	256	*221*	232	*157*	212	*31*	183	*61*
150	271	*269*	266	*259*	256	*239*	233	*182*	212	*123*	192	*85*
200	271	*270*	266	*262*	256	*249*	233	*213*	212	*198*	192	*57*
250	271	*264*	266	*266*	256	*255*	233	*221*	225	*180*	218	*147*
276	271	*271*	266	*266*	258	*255*	234	*225*	225	*180*	217	*138*

Table 6.2: Results for the 'Kop van Noord Holland' – Case 2.

Chapter 7

Summary and Conclusions

In this thesis, we developed new models and algorithms to support the design of a railway timetable. In Chapter 3, we proposed a new optimisation approach for strategic timetabling, i.e., to develop a completely new timetable from scratch. This is extended in Chapter 4 and applied to timetabling situations in which infrastructure related restrictions are relevant. In Chapter 5, we proposed an iterative approach to modify bounds of PESP-activities, to allow the existence of a feasible solution to a PESP instance. Finally, in Chapter 6, we proposed a mathematical model that incorporates periodic disturbances in a periodic timetabling model, to compute a delay-resistant timetable. All methods are tested on real-world instances from Netherlands Railways (NS).

7.1 Main Findings

In Chapter 3, we introduced the Strategic Passenger Oriented Timetabling (SPOT) problem. This problem aims at finding a timetable pattern which is optimal for passengers, explicitly including adaption time into the perceived passenger travel time. In our approach to solve the SPOT problem, we formulated a quadratic integer program, thus integrating a timetabling model with a passenger routing problem. Due to the strategic nature of the problem at hand, we formulated the SPOT problem without including safety constraints, so that the underlying timetabling problem is relatively simple. However, the inclusion of adaption time in the model formulation leads to a quadratic objective, making the model harder to solve again. We linearised the resulting model, and we proposed and tested an approach for solving it. We improved with respect to the computation time by warm-starting the model with a heuristically achieved solution. We have shown in our case studies, how the solutions generated by the SPOT model can be used to learn about desirable patterns at key points of the network.

In Chapter 4, we proposed an approach to solve the tactical timetabling problem. Similar to Chapter 3, we focused on the quality of the timetable for the passengers. In order to find a feasible passenger-oriented timetable for challenging real world instances, for which the timetabling model in itself is already challenging, we used variants of two existing approaches. These two approaches are combined into an algorithmic framework. First, an ideal timetable is computed, thereby neglecting infrastructure related restrictions, as is done in Chapter 3. Next, through a Lagrangian heuristic, this timetable is modified to obtain a feasible timetable with respect to infrastructure. A feedback mechanism is used to improve the found solutions. We showed that for real-life instances, based on the network operated by NS, we obtained satisfying results. Furthermore, we showed that the provided feedback indeed leads to (overall) better timetables compared to the initially computed timetable.

In Chapter 5, we developed a methodology to relax PESP activities to resolve infeasible PESP instances. This approach supplements current timetabling algorithms operated in practice, which suffer from the fact that increased demand for capacity usage as well as quality requirements often lead to (on first sight) infeasible timetabling instances. We resolved conflicts in a PESP model with as few deviations as possible, based on predefined activity weights and bounds. Our approach is iterative in the sense that we find a conflict in the existing PESP instance, solve this conflict using a MIP model, and then search for the next conflict. In contrast to existing approaches on resolving infeasible PESP instances, our approach can deal with fixed trip times, an assumption that is often made in PESP instances arising from railway timetabling. We found that in our iterative approach of finding and resolving conflicts it is important not to resolve only minimal conflicts, but to carefully add more activities from the timetabling instance to these minimal conflicts, in order to find good solutions as well as to improve computation times. We proposed several methods to enrich the minimal conflicts. From our computational results, we concluded that adding neighbouring activities in the constraint graph in general leads to better results. In our experiments based on parts of or the whole the Dutch railway network, feasible timetables are found in reasonable time in most cases.

Finally, in Chapter 6, we proposed an adjustable robust extension of the Periodic Timetabling Problem (PTP): the Robust Periodic Timetabling Problem (RPTP). The RPTP is a challenging optimization problem, as the non-robust PTP is already known to be computationally difficult, especially if also the service constraints are considered, which is the case in this chapter. We developed a solution method for the RPTP for a class of uncertainty regions. This class closely relates to uncertainty regions known in the robust optimization literature, and allows for a natural quantification of the robustness of a timetable. The developed solution method consists of two parts. We first reduce the model size by assuming a linear decision rule, i.e., the recourse actions depend linearly on the disturbances. The size of the resulting model is only a fraction of the original model's size, albeit at a loss of freedom in the decision rule. As a result, this approach can be considered to be heuristic. The new model is solved using dual reformulation techniques and cutting-plane methods, both well-established techniques in the robust optimization literature. We showed that the RPTP can be solved for practical-sized instances by applying the solution method to instances of NS. In particular, we studied the interplay between the severity of the disturbances, the total amount of timetable adjustments, and the efficiency of the undisturbed timetable. Our experiments show that the RPTP remains computationally tractable: Good solutions for challenging instances can be found in reasonable time. We also saw, however, that for large scale instances with moderate disturbance there remains room for improvement.

7.2 Practical Implications and Recommendations

The research presented in this thesis has been evaluated on practical instances of NS. Especially the Chapters 3–5 show interesting opportunities for further applications in practice and it would be recommended to design a Decision Support Systems integrating these approaches.

The approach in Chapters 3 and 4 considers the passengers as a starting point, i.e., based on passengers a timetable is designed. This leads to several interesting applications. The first of them is that the approach must be used to generate ideas for a new timetable. It should specifically be used as a Decision *Support* System, that can point out interesting characteristics of a newly computed timetable, that can be interesting for passengers. These characteristics should than be further investigated to find out why this characteristic is chosen and why it can be beneficial for the network. As an example, we have seen in practical tests performed by NS and by ourselves that in many solutions, trains are not synchronized. Instead, quite different solutions came out on some occasions. If two intercity trains and two local trains operate between two stations, currently they are often operated in an alternating manner, i.e., first an intercity train and then a local train. However, it turns out that first operating two intercity trains and then two local trains can be very good solutions as well. This is a suggestion the models can provide which should be considered to apply it in practice.

A second feature of this Decision Support System would be that it can help in guiding investments in infrastructure. Whereas traditionally the infrastructure network is considered as input to the line planning and timetabling problem, we can now reverse the order and start with a timetable (and a line plan). This implies that decisions on investments in infrastructure can now be guided by such a timetable, to obtain a better match between passenger demand and the service a network can offer. As an alternative, a hybrid approach can be chosen, in which part of the infrastructure is considered as input, and other parts are still flexible. The related safety restrictions can be included, either by the approach from Chapter 3, or using the iterative approach from Chapter 4. A comparison can be made to what extend safety restrictions influence the quality of the timetable for passengers, and thus be a measure to what extend an infrastructural investment is useful.

A third feature to be added to this system is to add the approach from Chapter 5. It may turn out that the tactical timetabling approach from Chapter 4 leads to a timetabling instance where many trains are cancelled. In that case, the approach from Chapter 5 can be used to detect conflicts in the instances and to propose solutions to resolve the conflict such that the tactical timetabling problem can find improved solutions with less cancelled trains.

7.3 Further Research

Optimizing a periodic timetable for perceived travel time of passengers, which includes adaption time, leads to a challenging optimisation problem. Especially if also headway restrictions are taken into account into the timetabling part of the model, as is done in Chapter 4. Although we proposed a heuristic optimisation approach to find reasonably good solutions, there is still a large optimality gap remaining, which needs to be reduced further. Currently, NS is further developing the approach from Chapter 3 to further close the optimality gap and reduce computation times.

Furthermore, they are deducing practical implications for the design of their railway network.
Our current model formulation in Chapter 3 (and hence also Chapter 4) relies on the assumption that passengers arrive uniformly distributed over the period for the definition of passenger groups and average waiting times. To include more detailed passenger information that may become available when entering the tactical or operational planning phase, different modelling approaches may be needed. Another property of the model is that one minute of additional perceived travel time for 50 passengers, weighs as much as 50 minutes for one passengers. It would be interesting to adapt the model to be able to incorporate other options here as well.

Reducing the optimality gaps from Chapter 3 will also benefit the approach proposed in Chapter 4. This chapter relies on the modelling approach from Chapter 3, as it is used in the proposed algorithmic framework. Interesting further research regarding this framework would include the further automatisation of the feedback procedure. Although this procedure is formalized in Section 4.4.3, it can still require manual inspection of the results in order to find a good feedback option.
Furthermore, it would be interesting to include station capacity in our models.

One of the key difficulties in Chapter 5 is to determine good values for the parameters of the algorithm, in order to find solutions that best represent solutions desired in real life applications. In practice, these parameters would need to be chosen based on the expertise and preferences of the railway operator - and could be adjusted based on the feedback of the operator when seeing the generated solution. It would even be possible to incorporate expert feedback (in the sense of resetting and fine-tuning parameters) after the resolution of each conflict.

In Chapter 6, it would be interesting to further analyse different, more complex, decision rules. In particular, it would be interesting to quantify the possible efficiency loss due to a linear decision rule. Note, however, that this would most likely greatly complicate any possible solution method. Furthermore, it would be interesting to consider a different class of uncertainty regions, based on, e.g., historical data.
Further research is needed to refine the solution methodology in order to find better solutions in situations with moderate disturbances.

References

Abbink, E. (2014). *Crew management in passenger rail transport* (PhD thesis). Erasmus University Rotterdam. Erasmus University Rotterdam.

Abbink, E., van den Berg, B., Kroon, L. & Salomon, M. (2004). Allocation of railway rolling stock for passenger trains. *Transportation Science*, *38*(1), 33–41. https://doi.org/10.1287/trsc.1030.0044

Ait-Ali, A., Lindberg, P.O., Eliasson, J., Nilsson, J.-E. & Peterson, A. (2020). A disaggregate bundle method for train timetabling problems. *Journal of Rail Transport Planning & Management*, in press.

Barrena, E., Canca, D., Coelho, L.C. & Laporte, G. (2014a). Exact formulations and algorithm for the train timetabling problem with dynamic demand. *Computers & Operations Research*, *44*, 66–74. https://doi.org/10.1016/j.cor.2013.11.003

Barrena, E., Canca, D., Coelho, L.C. & Laporte, G. (2014b). Single-line rail rapid transit timetabling under dynamic passenger demand. *Transportation Research Part B: Methodological*, *70*, 134–150. https://doi.org/10.1016/j.trb.2014.08.013

Belvaux, G., Boissin, N., Sutter, A. & Wolsey, L.A. (1998). Optimal placement of add /drop multiplexers static and dynamic models. *European Journal of Operational Research*, *108*(1), 26–35. https://doi.org/10.1016/S0377-2217(97)00021-0

Ben-Tal, A., Golany, B., Nemirovski, A. & Vial, J. (2005). Retailer-supplier flexible commitments contracts: A robust optimization approach. *Manufacturing & Service Operations Management*, *7*(3), 248–271.

Ben-Tal, A., Golany, B. & Shtern, S. (2009a). Robust multi-echelon multi-period inventory control. *European Journal of Operational Research*, *199*(3), 922–935.

Ben-Tal, A., El Ghaoui, L. & Nemirovski, A. (2009b). *Robust optimization*. Princeton University Press.

Bertsimas, D., Brown, D. & Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM review*, *53*(3), 464–501.

Bertsimas, D., Dunning, I. & Lubin, M. (2016). Reformulation versus cutting-planes for robust optimization. *Computational Management Science*, *13*(2), 195–217.

Bešinović, N., Goverde, R.M., Quaglietta, E. & Roberti, R. (2016). An integrated micro–macro approach to robust railway timetabling. *Transportation Research Part B: Methodological*, *87*, 14–32. https://doi.org/10.1016/j.trb.2016.02.004

Beter & Meer. (2014). Voorstel voor de operationele uitwerking van de lange termijn spooragenda [Online; Accessed on 2020-01-06].

Borndörfer, R., Hoppmann, H. & Karbstein, M. (2017). Passenger routing for periodic timetable optimization. *Public Transport*, *9*(1-2), 115–135. https://doi.org/10.1007/s12469-016-0132-0

Borndörfer, R., Klug, T., Lamorgese, L., Mannino, C., Reuther, M. & Schlechte, T. (Eds.). (2018). *Handbook of optimization in the railway industry* (Vol. 268). Springer. https://doi.org/10.1007/978-3-319-72153-8

Boston Consulting Group. (2017). The 2017 European Railway Performance Index.

Brännlund, U., Lindberg, P.O., Nou, A. & Nilsson, J.-E. (1998). Railway timetabling using lagrangian relaxation. *Transportation science*, *32*(4), 358–369.

Breugem, T. (2020). *Crew planning at Netherlands Railways* (PhD thesis). Erasmus University Rotterdam.

Burggraeve, S., Bull, S.H., Vansteenwegen, P. & Lusby, R.M. (2017). Integrating robust timetabling in line plan optimization for railway systems. *Transportation Research Part C: Emerging Technologies*, *77*, 134–160.

Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L. & Wagenaar, J. (2014). An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, *63*, 15–37.

Cacchiani, V., Caprara, A., Galli, L., Kroon, L. & Maróti, G. (2008a). Recoverable Robustness for Railway Rolling Stock Planning (M. Fischetti & P. Widmayer, Eds.). In M. Fischetti & P. Widmayer (Eds.), *8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'08)*, Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. https://doi.org/10.4230/OASIcs.ATMOS.2008.1590

Cacchiani, V., Caprara, A. & Toth, P. (2008b). A column generation approach to train timetabling on a corridor. *4OR*, *6*(2), 125–142.

Cacchiani, V., Caprara, A. & Toth, P. (2010). Scheduling extra freight trains on railway networks. *Transportation Research Part B: Methodological*, *44*(2), 215–231. https://doi.org/10.1016/j.trb.2009.07.007

Cacchiani, V. & Toth, P. (2012). Nominal and robust train timetabling problems. *European Journal of Operational Research*, *219*(3), 727–737. https://doi.org/10.1016/j.ejor.2011.11.003

Caimi, G., Kroon, L. & Liebchen, C. (2017). Models for railway timetable optimization: Applicability and applications in practice. *Journal of Rail Transport Planning & Management*, *6*(4), 285–312. https://doi.org/10.1016/j.jrtpm.2016.11.002

Caprara, A., Fischetti, M. & Toth, P. (2002). Modeling and solving the train timetabling problem. *Operations Research*, *50*(5), 851–861. https://doi.org/10.1287/opre.50.5.851.362

Cicerone, S., D'Angelo, G., Di Stefano, G., Frigioni, D. & Navarra, A. (2009). Recoverable robust timetabling for single delay: Complexity and polynomial algorithms for special cases. *Journal of Combinatorial Optimization*, *18*(3), 229–257. https://doi.org/10.1007/s10878-009-9247-4

Cicerone, S., Di Stefano, G., Schachtebeck, M. & Schöbel, A. (2012). Multi-stage recovery robustness for optimization problems: A new concept for planning under disturbances. *Information Sciences*, *190*, 107–126. https://doi.org/10.1016/j.ins.2011.12.010

D'Ariano, A., Pacciarelli, D. & Pranzo, M. (2007). A branch and bound algorithm for scheduling trains on a railway network. *European Journal of Operational Research*, *183*(2), 643–657.

De Keizer, B., Kouwenhoven, M. & Hofker, F. (2015). New insights in resistance to interchange [Current practices in transport: appraisal methods, policies and models - 42nd European Transport Conference Selected Proceedings]. *Transportation Research Procedia*, *8*, 72–79. https://doi.org/10.1016/j.trpro.2015.06.043

de Dios Ortúzar, J. & Willumsen, L.G. (2011). *Modelling transport*. John wiley & sons.

Deutschland-TAKT. (2019). TAKT website [Accessed on 2019-12-30].

Dewilde, T., Sels, P., Cattrysse, D. & Vansteenwegen, P. (2014). Improving the robustness in railway station areas. *European Journal of Operational Research*, *235*(1), 276–286. https://doi.org/10.1016/j.ejor.2013.10.062

Farina, F. (2019). *A heuristic for periodic train timetabling with integrated passenger routing* [Submitted to European Journal of Operational Research]. Submitted to European Journal of Operational Research.

Fioole, P.J., Kroon, L., Maróti, G. & Schrijver, A. (2006). A rolling stock circulation model for combining and splitting of passenger trains. *European Journal of Operational Research*, *174*, 1281–1297.

Fischetti, M. & Monaci, M. (2009). Light robustness. In R. Ahuja, R. Möhring & C. Zaroliagis (Eds.), *Robust and online large-scale optimization: Models and techniques for transportation systems* (pp. 61–84). Berlin, Heidelberg, Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-05465-5_3

Gattermann, P., Großmann, P., Nachtigall, K. & Schöbel, A. (2016). Integrating Passengers' Routes in Periodic Timetabling: A SAT approach (M. Goerigk & R. Werneck, Eds.). In M. Goerigk & R. Werneck (Eds.), *16th workshop on algorithmic approaches for transportation modelling, optimization, and systems (atmos 2016)*, Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum für Informatik. https://doi.org/10.4230/OASIcs.ATMOS.2016.3

Ghaemi, N., Cats, O. & Goverde, R.M. (2017). Railway disruption management challenges and possible solution directions. *Public Transport*, *9*, 343–364. https://doi.org/10.1007/s12469-017-0157-z

Ghaemi, N., Cats, O. & Goverde, R.M. (2018). Macroscopic multiple-station short-turning model in case of complete railway blockages. *Transportation Research Part C: Emerging Technologies*, *89*, 113–132. https://doi.org/10.1016/j.trc.2018.02.006

Goerigk, M. (2015). Exact and heuristic approaches to the robust periodic event scheduling problem. *Public Transport*, *7*(1), 101–119. https://doi.org/10.1007/s12469-014-0100-5

Goerigk, M. & Liebchen, C. (2017). An Improved Algorithm for the Periodic Timetabling Problem (G. D'Angelo & T. Dollevoet, Eds.). In G. D'Angelo & T. Dollevoet (Eds.), *17th workshop on algorithmic approaches for transportation modelling, optimization, and systems (atmos 2017)*, Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum für Informatik. https://doi.org/10.4230/OASIcs.ATMOS.2017.12

Goerigk, M. & Schöbel, A. (2010). An Empirical Analysis of Robustness Concepts for Timetabling (T. Erlebach & M. Lübbecke, Eds.). In T. Erlebach & M. Lübbecke (Eds.), *10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'10)*, Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum für Informatik. https://doi.org/10.4230/OASIcs.ATMOS.2010.100

Goerigk, M. & Schöbel, A. (2013). Improving the modulo simplex algorithm for large-scale periodic timetabling. *Computers & Operations Research*, *40*(5), 1363–1370. https://doi.org/10.1016/j.cor.2012.08.018

Goerigk, M. & Schöbel, A. (2014). Recovery-to-optimality: A new two-stage approach to robustness with an application to aperiodic timetabling. *Computers & Operations Research*, *52*, 1–15. https://doi.org/10.1016/j.cor.2014.06.025

Gorissen, B., Yanıkoğlu, İ. & den Hertog, D. (2015). A practical guide to robust optimization. *Omega*, *53*, 124–137. https://doi.org/10.1016/j.omega.2014.12.006

Goverde, R.M. (2007). Railway timetable stability analysis using max-plus system theory. *Transportation Research Part B: Methodological*, *41*(2), 179–201. https://doi.org/10.1016/j.trb.2006.02.003

Goverde, R.M., Bešinović, N., Binder, A., Cacchiani, V., Quaglietta, E., Roberti, R. & Toth, P. (2016). A three-level framework for performance-based railway timetabling. *Transportation Research Part C: Emerging Technologies*, *67*, 62–83. https://doi.org/10.1016/j.trc.2016.02.004

Großmann, P., Hölldobler, S., Manthey, N., Nachtigall, K., Opitz, J. & Steinke, P. (2012). Solving periodic event scheduling problems with SAT. In H. Jiang, W. Ding, M. Ali & X. Wu (Eds.), *Advanced research in applied artificial intelligence* (pp. 166–175). Springer Berlin Heidelberg.

Großmann, P., Opitz, J., Weiß, R. & Kümmling, M. (2015). On resolving infeasible periodic event networks, In *Conference on advanced systems in public transport*.

Hansen, I. & Pachl, J. (2014). *Railway timetabling & operations: Analysis, modelling, optimisation, simulation, performance evaluation*. DVV Media Group GmBH.

Hartleb, J. & Schmidt, M. (2019). Railway timetabling with integrated passenger distribution. *ERIM report series research in management Erasmus Research Institute of Management*, (ERS-2019-012-LIS). http://hdl.handle.net/1765/122487

Hartleb, J., Schmidt, M., Friedrich, M. & Huisman, D. (2019). A good or a bad timetable: Do different evaluation functions agree? *ERIM report series research in management Erasmus Research Institute of Management*, (ERS-2019-002-LIS). http://hdl.handle.net/1765/115831

Hooghiemstra, J.S., Kroon, L.G., Odijk, M., Salomon, M. & Zwaneveld, P.J. (1999). Decision support systems support the search for win-win solutions in railway network design. *INFORMS Journal on Applied Analytics*, *29*(2), 15–32. https://doi.org/10.1287/inte.29.2.15

Huisman, D. (2004). *Integrated and dynamic vehicle and crew scheduling* (PhD thesis). Erasmus University Rotterdam. Erasmus University Rotterdam. http://hdl.handle.net/1765/6779

Huisman, D., Kroon, L.G., Lentink, R.M. & Vromans, M.J. (2005). Operations research in passenger railway transportation. *Statistica Neerlandica*, *59*(4), 467–497.

Iancu, D., Sharma, M. & Sviridenko, M. (2013). Supermodularity and affine policies in dynamic robust optimization. *Operations Research*, *61*(4), 941–956.

IBM. (2015). IBM ILOG CPLEX Optimization Studio 12.6.2.

IBM. (2017). IBM ILOG CPLEX Optimization Studio 12.8.0.

IBM. (2019). IBM ILOG CPLEX Optimization Studio 12.9.0.

Kennisinstituut voor Mobiliteitsbeleid. (2019). *Mobiliteitsbeeld 2019* (tech. rep.). Ministerie van Infrastructuur en Waterstaat. Den Haag.

Kepaptsoglou, K. & Karlaftis, M. (2009). Transit route network design problem: Review. *Journal of Transportation Engineering*, *135*(8), 491–505. https://doi.org/10.1061/(ASCE)0733-947X(2009)135:8(491)

Kinder, M. (2008). *Models for periodic timetabling* (Diploma thesis). Technische Universität Berlin.

Kroon, L. (2015). Slides OML Course Public Transport.

Kroon, L., Huisman, D., Abbink, E., Fioole, P.J., Fischetti, M., Maróti, G., Schrijver, A., Steenbeek, A. & Ybema, R. (2009). The new Dutch timetable: The OR revolution. *Interfaces*, *39*(1), 6–17.

Kroon, L., Maróti, G., Retel Helmrich, M., Vromans, M. & Dekker, R. (2008). Stochastic improvement of cyclic railway timetables. *Transportation Research Part B: Methodological*, *42*(6), 553–570. https://doi.org/10.1016/j.trb.2007.11.002

Kroon, L. & Peeters, L. (2003). A variable trip time model for cyclic railway timetabling. *Transportation Science*, *37*(2), 198–212. https://doi.org/10.1287/trsc.37.2.198.15247

Kroon, L., Peeters, L.W., Wagenaar, J. & Zuidwijk, R. (2014). Flexible connections in pesp models for cyclic passenger railway timetabling. *Transportation Science*, *48*(1), 136–154. https://doi.org/10.1287/trsc.1120.0453

Kümmling, M., Großmann, P., Nachtigall, K., Opitz, J. & Weiß, R. (2015a). A state-of-the-art realization of cyclic railway timetable computation. *Public Transport*, *7*(3), 281–293. https://doi.org/10.1007/s12469-015-0108-5

Kümmling, M., Großmann, P., Opitz, J., Weiß, R. & Nachtigall, K. (2015b). Extraction of significant conflicts in periodic timetabling, In *Conference on advanced systems in public transport*.

Lamorgese, L. & Mannino, C. (2015). An exact decomposition approach for the real-time train dispatching problem. *Transportation Science*, *63*(1), 48–64. https://doi.org/10.1287/opre.2014.1327

Liebchen, C. (2003). Finding short integral cycle bases for cyclic timetabling. In G. Di Battista & U. Zwick (Eds.), *Algorithms - esa 2003* (pp. 715–726). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-39658-1_64

Liebchen, C. (2008). The first optimized railway timetable in practice. *Transportation Science*, *42*(4), 420–435. https://doi.org/10.1287/trsc.1080.0240

Liebchen, C., Lübbecke, M., Möhring, R. & Stiller, S. (2009). The concept of recoverable robustness, linear programming recovery, and railway applications. In R. Ahuja, R. Möhring & C. Zaroliagis (Eds.), *Robust and online large-scale optimization: Models and techniques for transportation systems* (pp. 1–27). Berlin, Heidelberg, Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-05465-5_1

Liebchen, C. & Möhring, R. (2007). The modeling power of the periodic event scheduling problem: Railway timetables – and beyond (F. Geraets, L. Kroon, A. Schöbel, D. Wagner & C. Zaroliagis, Eds.). In F. Geraets, L. Kroon, A. Schöbel, D. Wagner & C. Zaroliagis (Eds.), *Algorithmic methods for railway optimization: International dagstuhl workshop, dagstuhl castle, germany, june 20-25, 2004, 4th international workshop, atmos 2004, bergen, norway, september 16-17, 2004, revised selected papers*, Berlin, Heidelberg, Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-74247-0_1

Liebchen, C. & Peeters, L. (2009). Integral cycle bases for cyclic timetabling. *Discrete Optimization*, *6*(1), 98–109. https://doi.org/10.1016/j.disopt.2008.09.003

Liebchen, C., Proksch, M. & Wagner, F.H. (2008). Performance of algorithms for periodic timetable optimization. In M. Hickman, P. Mirchandani & S. Voß (Eds.), *Computer-aided systems in public transport* (pp. 151–180). Berlin, Heidelberg, Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-73312-6_8

Liebchen, C., Schachtebeck, M., Schöbel, A., Stiller, S. & Prigge, A. (2010). Computing delay resistant railway timetables. *Computers & Operations Research*, *37*(5), 857–868. https://doi.org/10.1016/j.cor.2009.03.022

Lin, Z. & Kwan, R. (2014). A two-phase approach for real-world train unit scheduling. *Public Transport*, *6*(1-2), 35–65. https://doi.org/10.1007/s12469-013-0073-9

Louwerse, I. & Huisman, D. (2014). Adjusting a railway timetable in case of partial or complete blockades. *European Journal of Operational Research*, *235*(3), 583–593.

Lübbe, J. (2009). *Passagierrouting und taktfahrplanoptimierung* (Diploma thesis). Technische Universität Berlin.

Lübbecke, M., Puchert, C., Schiewe, P. & Schöbel, A. (2018a). Integrating line planning, timetabling and vehicle scheduling, In *Conference on advanced systems in public transport.*

Lübbecke, M., Puchert, C., Schiewe, P. & Schöbel, A. (2018b). Integrating line planning, timetabling and vehicle scheduling. integer programming formulation and analysis, In *Conference on advanced systems in public transport.*

Lusby, R., Larsen, J. & Bull, S. (2018). A survey on robustness in railway planning. *European Journal of Operational Research*, *266*(1), 1–15. https://doi.org/10.1016/j.ejor.2017.07.044

Maróti, G. (2006). *Operations research models for railway rolling stock planning* (PhD thesis). Technische Universiteit Eindhoven.

Maróti, G. (2017). A branch-and-bound approach for robust railway timetabling. *Public Transport*, *9*, 73–94. https://doi.org/10.1007/s12469-016-0143-x

Martin-Iradi, B. & Røpke, S. (2019). A column-generation-based matheuristic for periodic train timetabling with integrated passenger routing.

Matos, G.P., Albino, L., Saldanha, R.L. & Morgado, E.M. (2017). Optimising cyclic timetables with a sat approach (E. Oliveira, J. Gama, Z. Vale & H. Lopes Cardoso, Eds.). In E. Oliveira, J. Gama, Z. Vale & H. Lopes Cardoso (Eds.), *Progress in artificial intelligence*, Cham, Springer International Publishing.

Matos, G.P., Albino, L.M., Saldanha, R.L. & Morgado, E.M. (2020). Solving periodic timetabling problems with SAT and machine learning. *Public Transport*. https://doi.org/10.1007/s12469-020-00244-y

Meng, L. & Zhou, X. (2011). Robust single-track train dispatching model under a dynamic and stochastic environment: A scenario-based rolling horizon solution approach. *Transportation Research Part B: Methodological*, *45*(7), 1080–1102.

Michaelis, M. & Schöbel, A. (2009). Integrating line planning, timetabling, and vehicle scheduling: A customer-oriented heuristic. *Public Transport*, *1*(3), 211.

Nachtigall, K. (1999). *Periodic network optimization and fixed interval timetables.* (Habilitation thesis). DLR Braunschweig.

Nachtigall, K. & Opitz, J. (2008). Solving Periodic Timetable Optimisation Problems by Modulo Simplex Calculations (M. Fischetti & P. Widmayer, Eds.). In M. Fischetti & P. Widmayer (Eds.), *8th workshop on algorithmic approaches for transportation modeling, optimization, and systems (atmos'08)*, Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum für Informatik. https://doi.org/10.4230/OASIcs.ATMOS.2008.1588

Niu, H., Zhou, X. & Gao, R. (2015). Train scheduling for minimizing passenger waiting time with time-dependent demand and skip-stop patterns: Nonlinear integer programming models with linear constraints. *Transportation Research Part B: Methodological*, *76*, 117–135.

NS. (2017). Website nederlandse spoorwegen [Accessed: 2017-11-03].

NS. (2019). NS jaarverslag. https://www.nsjaarverslag.nl

Odijk, M. (1996). A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research Part B: Methodological*, *30*(6), 455–464. https://doi.org/10.1016/0191-2615(96)00005-7

Odijk, M., Lentink, R. & Steenbeek, A. (2002). *CADANS/Conflex: Functionele beschrijving van het onderdeel Conflex* (tech. rep.). ORTEC Consultants BV, CWI. Gouda, Amsterdam.

Odijk, M., Lentink, R. & Steenbeek, A. (2003). *CADANS/Bepaal Speling: Functionele beschrijving en implementatie van het onderdeel Bepaal Speling* (tech. rep.). ORTEC Consultants BV, CWI. Gouda, Amsterdam.

Odijk, M., Romeijn, H.E. & van Maaren, H. (2006). Generation of classes of robust periodic railway timetables. *Computers & Operations Research*, *33*(8), 2283–2299. https://doi.org/10.1016/j.cor.2005.02.004

Pätzold, J., Schiewe, A., Schiewe, P. & Schöbel, A. (2017). Look-Ahead Approaches for Integrated Planning in Public Transportation (G. D'Angelo & T. Dollevoet, Eds.). In G. D'Angelo & T. Dollevoet (Eds.), *17th workshop on algorithmic approaches for transportation modelling, optimization, and systems (atmos 2017)*, Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum für Informatik. https://doi.org/10.4230/OASIcs.ATMOS.2017.17

Pätzold, J. & Schöbel, A. (2016). A Matching Approach for Periodic Timetabling (M. Goerigk & R. Werneck, Eds.). In M. Goerigk & R. Werneck (Eds.), *16th workshop on algorithmic approaches for transportation modelling, optimization, and systems (atmos 2016)*, Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum für Informatik. https://doi.org/10.4230/OASIcs.ATMOS.2016.1

Peeters, L. (2003). *Cyclic railway timetable optimization* (PhD thesis). Erasmus University Rotterdam. Erasmus University Rotterdam.

Pellegrini, P., Marlière, G., Pesenti, R. & Rodriguez, J. (2015). Recife-milp: An effective milp-based heuristic for the real-time railway traffic management problem. *IEEE Transactions on Intelligent Transportation Systems*, *16*(5), 2609–2619.

Polinder, G.J. (2015). *Resolving infeasibilities in the PESP model of the Dutch railway timetabling problem* (MSc thesis). Delft University of Technology.

Polinder, G.J., Cacchiani, V., Schmidt, M. & Huisman, D. (2020a). *An iterative heuristic for passenger-centric train timetabling with integrated adaption times* (ERS-2020-006-LIS). http://hdl.handle.net/1765/127816

Polinder, G.J., Schmidt, M. & Huisman, D. (2020b). *Timetabling for strategic passenger railway planning* (ERS-2020-001-LIS). http://hdl.handle.net/1765/123973

Polinder, G.J., Breugem, T., Dollevoet, T. & Maróti, G. (2019). An adjustable robust optimization approach for periodic timetabling. *Transportation Research Part B: Methodological*, *128*, 50–68. https://doi.org/10.1016/j.trb.2019.07.011

Polinder, G.J., Kroon, L., Aardal, K., Schmidt, M. & Molinaro, M. (2018). *Resolving infeasibilities in railway timetabling instances* (ERS-2018-002-LIS). http://hdl.handle.net/1765/104083

Radtke, A. (2014). Infrastructure modelling. In I. Hansen & J. Pachl (Eds.), *Railway timetable & operations*. DDV Media Group.

Robenek, T., Azadeh, S.S., Maknoon, Y. & Bierlaire, M. (2017). Hybrid cyclicity: Combining the benefits of cyclic and non-cyclic timetables. *Transportation*

Research Part C: Emerging Technologies, *75*, 228–253. https://doi.org/10.1016/j.trc.2016.12.015

Robenek, T., Maknoon, Y., Azadeh, S.S., Chen, J. & Bierlaire, M. (2016). Passenger centric train timetabling problem. *Transportation Research Part B: Methodological*, *89*, 107–126.

Rockafellar, R. (1998). *Network flows and monotropic optimization* [First published by John Wiley and Sons, Inc., in 1984]. Athena Scientific.

Schiewe, P. & Schöbel, A. (2018). *Periodic timetabling with integrated routing: An applicable approach* [Preprint nr. 17, Preprint-Serie des Instituts für Numerische und Angewandte Mathematik, Universitat Göttingen. http://num.math.uni-goettingen.de/preprints/files/2018-17.pdf]. Preprint nr. 17, Preprint-Serie des Instituts für Numerische und Angewandte Mathematik, Universitat Göttingen. http://num.math.uni-goettingen.de/preprints/files/2018-17.pdf.

Schmidt, M. & Schöbel, A. (2015a). Timetabling with passenger routing. *OR Spectrum*, *37*(1), 75–97. https://doi.org/10.1007/s00291-014-0360-0

Schmidt, M. & Schöbel, A. (2015b). Timetabling with passenger routing. *OR spectrum*, *37*(1), 75–97.

Schöbel, A. (2012). Line planning in public transportation: Models and methods. *OR Spectrum*, *34*, 491–510.

Schöbel, A. (2015). Integration of line planning and timetabling, some first approaches, In *Conference on advanced systems in public transport*.

Schöbel, A. (2017). An eigenmodel for iterative line planning, timetabling and vehicle scheduling in public transportation. *Transportation Research Part C: Emerging Technologies*, *74*, 348–365. https://doi.org/10.1016/j.trc.2016.11.018

Schrijver, A. & Steenbeek, A. (1993). *Dienstregelingontwikkeling voor railned* (tech. rep.) [In Dutch]. CWI. In Dutch. Amsterdam. https://homepages.cwi.nl/~lex/sdo.pdf

Sels, P., Dewilde, T., Cattrysse, D. & Vansteenwegen, P. (2016). Reducing the passenger travel time in practice by the automated construction of a robust railway timetable. *Transportation Research Part B: Methodological*, *84*, 124–156. https://doi.org/10.1016/j.trb.2015.12.007

Serafini, P. & Ukovich, W. (1989). A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, *2*(4), 550–581. https://doi.org/10.1137/0402049

Siebert, M. (2011). *Integration of routing and timetabling in public transportation* (MSc thesis). Georg-August-Universität Göttingen.

Siebert, M. & Goerigk, M. (2013). An experimental comparison of periodic timetabling models. *Computers & Operations Research*, *40*(10), 2251–2259. https://doi.org/10.1016/j.cor.2013.04.002

Sparing, D. & Goverde, R.M. (2017). A cycle time optimization model for generating stable periodic railway timetables. *Transportation Research Part B: Methodological*, *98*, 198–223. https://doi.org/10.1016/j.trb.2016.12.020

Veelenturf, L., Kidd, M., Cacchiani, V., Kroon, L. & Toth, P. (2016). A railway timetable rescheduling approach for handling large-scale disruptions. *Transportation Science*, *50*(3), 841–862. https://doi.org/10.1287/trsc.2015.0618

Veelenturf, L. (2014). *Disruption management in passenger railways - models for timetable, rolling stock and crew rescheduling* (PhD thesis). Erasmus University Rotterdam.

Wang, Y., Tang, T., Ning, B., van den Boom, T. & de Schutter, B. (2015). Passenger-demands-oriented train scheduling for an urban rail transit network. *Transportation Research Part C: Emerging Technologies*, *60*, 1–23. https://doi.org/10.1016/j.trc.2015.07.012

Warmerdam, J. (2004). *Specificaties trans toedeler* (tech. rep.). QQQ Delft.

Wolsey, L. (1998). *Integer programming*. Wiley.

Yan, F., Bešinović, N. & Goverde, R.M. (2019). Multi-objective periodic railway timetabling on dense heterogeneous railway corridors. *Transportation Research Part B: Methodological*, *125*, 52–75. https://doi.org/10.1016/j.trb.2019.05.002

Yan, F. & Goverde, R.M. (2019). Combined line planning and train timetabling for strongly heterogeneous railway lines with direct connections. *Transportation Research Part B: Methodological*, *127*, 20–46.

Yanıkoğlu, İ., Gorissen, B. & den Hertog, D. (2018). A survey of adjustable robust optimization. *European Journal of Operational Research*. https://doi.org/10.1016/j.ejor.2018.08.031

Yin, J., Yang, L., Tang, T., Gao, Z. & Ran, B. (2017). Dynamic passenger demand oriented metro train scheduling with energy-efficiency and waiting time minimization: Mixed-integer linear programming approaches. *Transportation Research Part B: Methodological*, *97*, 182–213.

Zhang, Y., Peng, Q., Yao, Y., Zhang, X. & Zhou, X. (2019). Solving cyclic train timetabling problem through model reformulation: Extended time-space network construct and alternating direction method of multipliers methods. *Transportation Research Part B: Methodological*, *128*, 344–379.

Zhen, J., den Hertog, D. & Sim, M. (2018). Adjustable robust optimization via Fourier–Motzkin elimination. *Operations Research*, *66*(4), 1086–1100.

Zhu, Y. & Goverde, R.M. (2019). Railway timetable rescheduling with flexible stopping and flexible short-turning during disruptions. *Transportation Research Part B: Methodological*, *123*, 149–181. https://doi.org/10.1016/j.trb.2019.02.015

Zhu, Y. & Goverde, R.M. (2020). Dynamic and robust timetable rescheduling for uncertain railway disruptions. *Journal of Rail Transport Planning & Management*, *15*, 100196. https://doi.org/10.1016/j.jrtpm.2020.100196

Zhu, Y., Mao, B., Bai, Y. & Chen, S. (2017). A bi-level model for single-line rail timetable design with consideration of demand and capacity. *Transportation Research Part C: Emerging Technologies*, *85*, 211–233.

Zwaneveld, P., Kroon, L., Romeijn, H., Salomon, M., Dauzère-Pérès, S. & van Hoesel, S. (1996). Routing trains through railway stations: Model formulation and algorithms. *Transportation Science*, *30*(3), 179–272. https://doi.org/10.1287/trsc.30.3.181

Nederlandse Samenvatting (Summary in Dutch)

Het feit dat op een gemiddelde werkdag ongeveer 750.000 reizigers in Nederland de trein gebruiken, laat zien dat de trein een belangrijk vervoersmiddel is. Met de toenemende aandacht voor milieubewust transport is de verwachting dat het vervoer via het spoor nog verder zal toenemen. Het is daarom van groot belang dat het vervoersaanbod op het spoor efficiënt is ingericht om reizigers zo snel mogelijk van hun herkomst naar hun bestemming te brengen. In dit proefschrift doen we onderzoek naar het opzetten van een vervoersnetwerk dat zo gunstig mogelijk is voor reizigers.

In Nederland wordt een cyclische dienstregeling gehanteerd. Dat wil zeggen dat de dienstregeling zich telkens na een bepaalde periode herhaalt; in Nederland is dat elk uur. De dienstregeling voor één uur noemen we het basisuurpatroon. Hoewel het mogelijk is dat op specifieke momenten (bijvoorbeeld tijdens de spits of de nacht) meer of minder treinen rijden dan in het basisuurpatroon zijn opgenomen, vormt dit patroon de basis voor de dienstregeling.

Voordat een dienstregeling kan worden gemaakt moet een lijnvoering bekend zijn. De lijnvoering geeft aan welke treinen zullen gaan rijden, hoe vaak per uur, wat hun route is en waar ze zullen stoppen. Bij het maken van een dienstregeling worden deze treinen ingepland. Dat moet op een zodanige manier worden gedaan dat aan een grote hoeveelheid restricties en wensen wordt voldaan. We noemen er een aantal:

- Veiligheidseisen verplichten dat er voldoende tijd moet worden ingepland tussen twee treinen die hetzelfde stukje spoor gebruiken.
- Om onnodig wachten door reizigers te voorkomen, moeten aansluitingen tussen treinen zo efficiënt mogelijk zijn.

- Naast goede aansluitingen is het voor reizigers belangrijk dat reisopties zo veel mogelijk verspreid zijn over de tijd.
- Ook moet een dienstregeling bestand zijn tegen kleine verstoringen.
- Tenslotte is het een reële mogelijkheid dat er dermate veel eisen aan dienstregelingen worden gesteld, dat er geen enkele dienstregeling bestaat die aan al die eisen voldoet. In dat geval zullen enkele eisen moeten worden versoepeld.

Al met al is het maken van een dienstregeling een erg ingewikkelde puzzel, waarbij de technieken van mathematische besliskunde van grote betekenis zijn. In dit proefschrift gebruiken we deze technieken om methoden te ontwikkelen die helpen bij het ontwerpen van een dienstregeling.

In hoofdstuk 3 wordt een methode ontwikkeld die een ideaal basisuurpatroon bepaalt. Dat wil zeggen: hoe zouden treinen idealiter moeten worden ingepland om reizigers zo snel mogelijk naar hun bestemming te brengen als we voldoende infrastructuur tot onze beschikking zouden hebben? We nemen dus aan dat treinen elkaar nooit kunnen hinderen, iedere trein zou op zijn eigen spoor kunnen rijden en in stations is altijd voldoende capaciteit beschikbaar om treinen te laten stoppen. Als gevolg van deze aanname kan elke willekeurige dienstregeling worden gerealiseerd, maar niet elke dienstregeling is even gunstig voor reizigers. Om de kwaliteit van een dienstregeling te beoordelen, gebruiken we als maat de ‘totale ervaren reistijd’ van alle reizigers samen. De ervaren reistijd van een reiziger is een gewogen som van verschillende tijden: de tijd in de trein, de tijd die de reiziger moet wachten op zijn vertrekstation, de tijd die besteed wordt aan een overstap, plus een ‘boete’ (in de vorm van tijd) voor iedere overstap die een reiziger moet maken. De ideale dienstregeling die we zoeken in dit hoofdstuk is dus een dienstregeling die de totale ervaren reistijd van alle reizigers samen minimaliseert.
Uit de resultaten in dit hoofdstuk blijkt dat het gunstig is om de vertrekmomenten van treinen te spreiden over de tijd, als we naar één reiziger kijken. Als er bijvoorbeeld twee treinen per uur vertrekken, is het gunstig om de tijd tussen deze twee treinen zo dicht mogelijk bij de 30 minuten te houden, in plaats van 20 en 40 minuten tijd tussen de vertrekmomenten. Hierdoor is de verwachte wachttijd op het herkomststation zo klein mogelijk, als deze reiziger op ieder willekeurig moment op het station kan arriveren. Houden we echter rekening met meerdere reizigers die allemaal andere herkomsten en bestemmingen hebben, dan blijkt dat deze spreiding vaak niet voor iedereen de beste oplossing is. Een perfecte spreiding is meestal niet mogelijk zonder op andere punten in de dienstregeling consessies te doen, bijvoorbeeld in halteer-

tijden: de tijd die een trein stilstaat op een station. Dit is met name het geval als het aantal treinen per uur op een traject wijzigt, en is het beste te illustreren aan de hand van een voorbeeld. Als er vanuit Groningen naar Utrecht twee treinen per uur vertrekken, dan is het voor reizigers vanuit Groningen het beste als deze treinen 30 minuten na elkaar vertrekken, zodat de verwachte wachttijd zo klein mogelijk is. Als er echter vanuit Zwolle nog een extra trein naar Utrecht gaat, zou het voor reizigers van Zwolle naar Utrecht het beste zijn als deze drie treinen vanuit Zwolle 20 minuten na elkaar vertrekken. Als alle treinen even snel rijden, gaan deze twee opties nooit goed samen, en moet er ergens een compromis gesloten worden. Onze methode maakt inzichtelijk wat vanuit het perspectief van alle reizigers samen een goed compromis is. Daaruit blijkt dat het regelmatig verstandiger is om treinen in een onregelmatig patroon te laten vertrekken vanaf stations om op andere locaties betere aansluitingen of vertrekpatronen te kunnen realiseren en voor alle reizigers samen een betere dienstregeling te verkrijgen. Evenzo kan het gunstig zijn om niet altijd dezelfde treinen op elkaar aan te laten sluiten, maar dit af te wisselen; bijvoorbeeld het ene kwartier de ene aansluiting en het andere kwartier een andere aansluiting.
De manier van plannen uit hoofdstuk 3 is met name geschikt om te gebruiken lang voordat de dienstregeling daadwerkelijk gaat worden uitgevoerd, omdat deze methode een ruwe structuur bepaalt en omdat er in dit stadium ook nog mogelijkheden zijn om infrastructuur aan te leggen.

In hoofdstuk 4 wordt het model uit hoofdstuk 3 uitgebreid: we nemen niet langer aan dat er voldoende infrastructuur is, maar houden rekening met de bestaande infrastructuur. Dit is daarmee een lastiger probleem, maar we hebben een iteratieve methode ontwikkeld die voor lijnvoeringen met bijna 100 treinen alsnog een goed basisuurpatroon kan maken. Daarnaast heeft deze methode als voordeel, in tegenstelling tot veel bestaande dienstregelingsalgoritmen, dat enkele treinen niet ingepland worden als de capaciteit op het spoor ontoereikend is. De methode die toegepast wordt gaat op zoek naar een dienstregeling die kan worden uitgevoerd op de bestaande infrastructuur en die zo dicht mogelijk bij een ideale dienstregeling blijft zoals berekend is in hoofdstuk 3.

De methoden die in hoofdstuk 5 en 6 zijn beschreven, zijn eveneens relevant om te komen tot een dienstregeling die gunstig is voor reizigers. De beschreven methoden kunnen circa een half jaar voor de daadwerkelijke uitvoering van de dienstregeling worden toegepast. In hoofdstuk 5 gaan we verder in op het probleem dat een dienstregeling aan te veel eisen moet voldoen, met als gevolg dat geen enkele dienstregeling mogelijk is. We onderzoeken hoe we aan zoveel mogelijk eisen kunnen voldoen en welke eisen moeten worden versoepeld om alsnog een dienstregeling te kunnen maken. Hiervoor wordt een heuristieke methode gebruikt die iteratief kleine conflicten oplost. Met kleine aanpassingen aan de eisen waar de dienstregeling aan moet voldoen blijkt het mogelijk een goede dienstregeling te maken. Een deel van de planningsproblemen kan opgelost worden door extra tijd die in de dienstregeling toegevoegd is om vertragingen te absorberen iets anders te verdelen.

In hoofdstuk 6 bespreken we hoe een dienstregeling kan worden gemaakt die robuust is tegen verstoringen. In een robuuste dienstregeling worden verstoringen zoveel mogelijk 'geabsorbeerd'. In dit hoofdstuk gaan we uit van periodieke verstoringen. Dit zijn verstoringen die elk uur op eenzelfde manier terugkomen, maar die niet elke dag optreden. Een voorbeeld hiervan is een dag dat er veel bladeren op de rails vallen, waardoor treinen langzamer moeten rijden. Op reguliere dagen moet de 'normale' dienstregeling worden gereden. Echter, op een dergelijke herfstdag moet de dienstregeling aangepast worden, zonder daarbij een hele andere dienstregeling uit te voeren. Onze methode berekent een dienstregeling die op reguliere dagen uitgevoerd kan worden, maar hij bepaalt ook hoe deze dienstregeling aangepast kan worden in het geval van kleine verstoringen. Op een aantal locaties in het netwerk krijgen treinen dan een bepaalde hoeveelheid extra tijd om de eventuele verstoringen zo goed mogelijk op te vangen.

Concluderend worden in dit proefschrift verschillende methoden voorgesteld die nuttig zijn bij het ontwerpen van een dienstregeling waarbij de primaire focus ligt op de reiziger. Met name de eerste twee hoofdstukken beginnen vanuit het oogpunt van de reiziger in plaats van, zoals in veel bestaande modellen wordt gedaan, vanuit het oogpunt van de bestaande infrastructuur. Met de methoden die in dit proefschrift zijn voorgesteld kunnen spoorvervoerders nog betere dienstregelingen ontwerpen om zo goed mogelijk aan te sluiten op de vervoersvraag van de reiziger.

About the author

Gert-Jaap was born on January 22, 1991 in Zwolle, the Netherlands. He received his bachelor's and master's degree in Applied Mathematics from Delft University of Technology, with a specialization on Operations Research and Optimization. His graduation study involved an applied project at Netherlands Railways, to improve current periodic timetabling algorithms. He continued studying algorithmic approaches to design periodic timetables as a PhD-candidate at the department of Technology of Operations Management of Rotterdam School of Management. This is done in close cooperation with the department of Process quality and Innovation (π) of Netherlands Railways, and under the supervision of Prof. Leo Kroon, Prof. Dennis Huisman and Dr. Marie Schmidt. As part of a collaborative project, Gert-Jaap spent two months in Bologna, Italy, to work with Dr. Valentina Cacchiani. He has presented his work on several international scientific conferences like TRISTAN, CASPT and INFORMS. He came in third in the 2019 INFORMS RAS Student Paper Competition. Furthermore, part of his work is published in scientific journals such as *Transportation Research Part B: Methodological*.

Currently, Gert-Jaap works as an Intelligent Systems Designer at ARVOO Imaging Products in Harmelen, the Netherlands.

Portfolio

Publications

International Journals

G.J. Polinder, T. Breugem, T. Dollevoet, G. Maróti (2019). "An adjustable robust optimization approach for periodic timetabling". *Transportation Research Part B: Methodological* 128, pp. 50-68.

Peer reviewed conference proceedings

G.J. Polinder, L.G. Kroon, M.E. Schmidt, K.I. Aardal, M. Molinaro (2016). "Resolving infeasibilities in railway timetabling instances". In *TRISTAN IX: Triennial Symposium on Transportation Analysis, Aruba, The Netherlands, June 13–17, 2016.*

G.J. Polinder, T. Breugem, T. Dollevoet (2017). "Robust Periodic Timetabling". In *RailLille 2017: Proceedings of the 7th International Conference on Railway Operations Modelling and Analysis (pp. 120–130), Lille, France, April 4–7, 2017.*

G.J. Polinder, M.E. Schmidt, D. Huisman (2018). "Incorporating passenger waiting times in timetabling". In *CASPT 2018: Conference on Advanced Systems in Public Transport, Brisbane, Australia, July 22–25, 2018.*

G.J. Polinder, M.E. Schmidt, D. Huisman (2019). "A new approach to periodic railway timetabling". In *RailNorrköping 2019: Proceedings of the 8th International Conference on Railway Operations Modelling and Analysis (pp. 1365–1381), Norrköping, Sweden, June 17–20, 2019.*

Technical Reports

G.J. Polinder, L.G. Kroon, K.I. Aardal, M.E. Schmidt, M. Molinaro (2018). *Resolving infeasibilities in railway timetabling instances.*
Tech. rep. ERS-2018-002-LIS. http://hdl.handle.net/1765/104083.
G.J. Polinder, M.E. Schmidt, D. Huisman (2020). *Timetabling for strategic passenger railway planning.*
Tech. rep. ERS-2020-001-LIS. http://hdl.handle.net/1765/123973.
G.J. Polinder, V. Cacchiani, M.E. Schmidt, D. Huisman (2020). *An iterative heuristic for passenger-centric train timetabling with integrated adaption times.*
Tech. rep. ERS-2020-006-LIS. http://hdl.handle.net/1765/127816.

Teaching

Master Thesis Supply Chain Management, Rotterdam School of Management. Supervision. 2015-2019.
Mathematics, Rotterdam School of Management. Tutorial lecturer and webcast lecturer. 2015-2020.
Distribution Networks, Rotterdam School of Management. Workshop lecturer AIMMS. 2016-2020.

PhD Courses

Algorithms and Complexity
Convex Analysis for Optimization
Integer Programming Methods
Interior Point Methods
Networks and Polyhedra
Robust Optimization
Stochastic Programming
Column Generation
Public Transport
Transport Logistics Modelling
Constraint Programming
Publishing Strategy
Scientific Integrity
Research Proposal Writing
English (CPE certificate)

Conferences Attended

INFORMS Annual Meeting 2019, Seattle, USA.
RailNorrköping 2019, Norrköping, Sweden.
CASPT 2018, Brisbane, Australia.
EURO-ALIO 2018, Bologna, Italy.
TRAIL Conference 2017, 2018, 2019, Utrecht, The Netherlands.
Workshop on timetabling (FOR2083) 2017, Göttingen, Germany
RailLille 2017, Lille, France.
EURO 2016, Poznan, Poland.
TRISTAN IX 2016, Aruba.
LNMB conference 2016, 2017, 2018, 2019, Lunteren, The Netherlands.
CASPT 2015, Rotterdam, The Netherlands.

The ERIM PhD Series

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

Dissertations last four years

Ahmadi, S., *A motivational perspective to decision-making and behavior in organizations*,
Promotors: Prof. J.J.P. Jansen & Dr. T.J.M. Mom,
EPS-2019-477-S&E, https://repub.eur.nl/pub/116727

Akemu, O., *Corporate Responses to Social Issues: Essays in Social Entrepreneurship and Corporate Social Responsibility*, Promotors: Prof. G.M. Whiteman & Dr. S.P. Kennedy,
EPS-2017-392-ORG, https://repub.eur.nl/pub/95768

Albuquerque de Sousa, J.A., *International stock markets: Essays on the determinants and consequences of financial market development*,
Promotors: Prof. M.A. van Dijk & Prof. P.A.G. van Bergeijk,
EPS-2019-465-F&A, https://repub.eur.nl/pub/115988

Alserda, G.A.G., *Choices in Pension Management*,
Promotors: Prof. S.G. van der Lecq & Dr. O.W. Steenbeek,
EPS-2017-432-F&A, https://repub.eur.nl/pub/103496

Arampatzi, E., *Subjective Well-Being in Times of Crises: Evidence on the Wider Impact of Economic Crises and Turmoil on Subjective Well-Being*,
Promotors: Prof. H.R. Commandeur, Prof. F. van Oort & Dr. M.J. Burger,
EPS-2018-459-S&E, https://repub.eur.nl/pub/111830

Arslan, A.M., *Operational Strategies for On-demand Delivery Services*,
Promotors: Prof. R.A. Zuidwijk & Dr. N.A.H. Agatz,
EPS-2019-481-LIS, https://repub.eur.nl/pub/126463

Avci, E., *Surveillance of Complex Auction Markets: a Market Policy Analytics Approach*,
Promotors: Prof. W. Ketter, Prof. H.W.G.M. van Heck & Prof. D.W. Bunn,
EPS-2018-426-LIS, https://repub.eur.nl/pub/106286

Balen, T.H. van, *Challenges of Early Stage Entrepreneurs: the Roles of Vision Communication and Team Membership Change*,
Promotors: Prof. J.C.M. van den Ende & Dr. M. Tarakci,
EPS-2019-468-LIS, https://repub.eur.nl/pub/115654

Bernoster, I., *Essays at the Intersection of Psychology, Biology, and Entrepreneurship*,
Promotors: Prof. A.R. Thurik, Prof. I.H.A. Franken & Prof. P.J.F. Groenen,
EPS-2018-463-S&E, https://repub.eur.nl/pub/113907

Blagoeva, R.R., *The Hard Power Of Soft Power: A behavioral strategy perspective on how power, reputation, and status affect firms*,
Promotors: Prof. J.J.P. Jansen & Prof. T.J.M. Mom,
EPS-2020-495-S&E, https://repub.eur.nl/pub/127681

Bouman, P., *Passengers, Crowding and Complexity: Models for Passenger Oriented Public Transport*, Promotors: Prof. L.G. Kroon, Prof. A. Schöbel & Prof. P.H.M. Vervest,
EPS-2017-420-LIS, https://repub.eur.nl/pub/100767

Breugem, T., *Crew Planning at Netherlands Railways: Improving Fairness, Attractiveness, and Efficiency*, Promotors: Prof. D. Huisman & Dr. T.A.B. Dollevoet,
EPS-2020494-LIS, https://repub.eur.nl/pub/124016

Bunderen, L. van, *Tug-of-War: Why and when teams get embroiled in power struggles*,
Promotors: Prof. D.L. van Knippenberg & Dr. L. Greer,
EPS-2018-446-ORG, https://repub.eur.nl/pub/105346

Burg, G.J.J. van den, *Algorithms for Multiclass Classification and Regularized Regression*,
Promotors: Prof. P.J.F. Groenen & Dr. A. Alfons,
EPS-2018-442-MKT, https://repub.eur.nl/pub/103929

Chammas, G., *Portfolio concentration*, Promotor: Prof. J. Spronk,
EPS-2017-410-F&E, https://repub.eur.nl/pub/94975

Chan, H.Y., *Decoding the consumer's brain: Neural representations of consumer experience*, Promotors: Prof. A. Smidts & Dr. M.A.S. Boksem, EPS-2019-493-MKT, https://repub.eur.nl/pub/124931

Dalmeijer, K., *Time Window Assignment in Distribution Networks*, Promotors: Prof. A.P.M. Wagelmans & Dr. R. Spliet, EPS-2019-486-LIS, https://repub.eur.nl/pub/120773

Dennerlein, T., *Empowering Leadership and Employees' Achievement Motivations: the Role of Self-Efficacy and Goal Orientations in the Empowering Leadership Process*, Promotors: Prof. D.L. van Knippenberg & Dr. J. Dietz, EPS-2017-414-ORG, https://repub.eur.nl/pub/98438

Dolgova, E., *On Getting Along and Getting Ahead: How Personality Affects Social Network Dynamics*, Promotors: Prof. P.P.M.A.R. Heugens & Prof. M.C. Schippers, EPS-2019-455-S&E, https://repub.eur.nl/pub/119150

Duijzer, L.E., *Mathematical Optimization in Vaccine Allocation*, Promotors: Prof. R. Dekker & Dr. W.L. van Jaarsveld, EPS-2017-430-LIS, https://repub.eur.nl/pub/101487

Eijlers, E., *Emotional Experience and Advertising Effectiveness: on the use of EEG in marketing*, Promotors: Prof. A. Smidts & Prof. M.A.S. Boksem, EPS-2019-487-MKT, https://repub.eur.nl/pub/124053

El Nayal, O.S.A.N., *Firms and the State: An Examination of Corporate Political Activity and the Business-Government Interface*, Promotors: Prof. J. van Oosterhout & Dr. M. van Essen, EPS-2018-469-S&E, https://repub.eur.nl/pub/114683

Feng, Y., *The Effectiveness of Corporate Governance Mechanisms and Leadership Structure: Impacts on strategic change and firm performance*, Promotors: Prof. F.A.J. van den Bosch, Prof. H.W. Volberda & Dr. J.S. Sidhu, EPS-2017-389-S&E, https://repub.eur.nl/pub/98470

Frick, T.W., *The Implications of Advertising Personalization for Firms, Consumer, and Ad Platfroms*, Promotors: Prof. T. Li & Prof. H.W.G.M. van Heck, EPS-2018-452-LIS, https://repub.eur.nl/pub/110314

Fytraki, A.T., *Behavioral Effects in Consumer Evaluations of Recommendation Systems*, Promotors: Prof. B.G.C. Dellaert & Prof. T. Li, EPS-2018-427-MKT, https://repub.eur.nl/pub/110457

Gai, J., *Contextualized Consumers: Theories and Evidence on Consumer Ethics, Product Recommendations, and Self-Control*, Promotors: Prof. S. Puntoni & Prof. S.T.L. Sweldens, EPS-2020-498-MKT, https://repub.eur.nl/pub/127680

Ghazizadeh, P., *Empirical Studies on the Role of Financial Information in Asset and Capital Markets*, Promotors: Prof. A. de Jong & Prof. E. Peek, EPS-2019-470-F&A, https://repub.eur.nl/pub/114023

Giurge, L., *A Test of Time; A temporal and dynamic approach to power and ethics*, Promotors: Prof. M.H. van Dijke & Prof. D. De Cremer, EPS-2017-412-ORG, https://repub.eur.nl/pub/98451

Gobena, L., *Towards Integrating Antecedents of Voluntary Tax Compliance*, Promotors: Prof. M.H. van Dijke & Dr. P. Verboon, EPS-2017-436-ORG, https://repub.eur.nl/pub/103276

Groot, W.A., *Assessing Asset Pricing Anomalies*, Promotors: Prof. M.J.C.M. Verbeek & Prof. J.H. van Binsbergen, EPS-2017-437-F&A, https://repub.eur.nl/pub/103490

Hanselaar, R.M., *Raising Capital: On pricing, liquidity and incentives*, Promotors: Prof. M.A. van Dijk & Prof. P.G.J. Roosenboom, EPS-2018-429-F&A, https://repub.eur.nl/pub/113274

Harms, J.A., *Essays on the Behavioral Economics of Social Preferences and Bounded Rationality*, Promotors: Prof. H.R. Commandeur & Dr. K.E.H. Maas, EPS-2018-457-S&E, https://repub.eur.nl/pub/108831

Hendriks, G., *Multinational Enterprises and Limits to International Growth: Links between Domestic and Foreign Activities in a Firm's Portfolio*, Promotors: Prof. P.P.M.A.R. Heugens & Dr. A.H.L. Slangen, EPS-2019-464-S&E, https://repub.eur.nl/pub/114981

Hengelaar, G.A., *The Proactive Incumbent: Holy grail or hidden gem? Investigating whether the Dutch electricity sector can overcome the incumbent's curse and lead the sustainability transition*, Promotors: Prof. R.J.M. van Tulder & Dr. K. Dittrich, EPS-2018-438-ORG, https://repub.eur.nl/pub/102953

Jacobs, B.J.D., *Marketing Analytics for High-Dimensional Assortments*, Promotors: Prof. A.C.D. Donkers & Prof. D. Fok, EPS-2017-445-MKT, https://repub.eur.nl/pub/103497

Jia, F., *The Value of Happiness in Entrepreneurship*, Promotors: Prof. D.L. van Knippenberg & Dr. Y. Zhang, EPS-2019-479-ORG, https://repub.eur.nl/pub/115990

Kahlen, M.T., *Virtual Power Plants of Electric Vehicles in Sustainable Smart Electricity Markets*, Promotors: Prof. W. Ketter & Prof. A. Gupta,
EPS-2017-431-LIS, https://repub.eur.nl/pub/100844

Kampen, S. van, *The Cross-sectional and Time-series Dynamics of Corporate Finance: Empirical evidence from financially constrained firms*,
Promotors: Prof. L. Norden & Prof. P.G.J. Roosenboom,
EPS-2018-440-F&A, https://repub.eur.nl/pub/105245

Karali, E., *Investigating Routines and Dynamic Capabilities for Change and Innovation*,
Promotors: Prof. H.W. Volberda, Prof. H.R. Commandeur & Dr. J.S. Sidhu,
EPS-2018-454-S&E, https://repub.eur.nl/pub/106274

Keko. E, *Essays on Innovation Generation in Incumbent Firms*,
Promotors: Prof. S. Stremersch & Dr. N.M.A. Camacho,
EPS-2017-419-MKT, https://repub.eur.nl/pub/100841

Kerkkamp, R.B.O., *Optimisation Models for Supply Chain Coordination under Information Asymmetry*, Promotors: Prof. A.P.M. Wagelmans & Dr. W. van den Heuvel,
EPS-2018-462-LIS, https://repub.eur.nl/pub/109770

Khattab, J., *Make Minorities Great Again: a contribution to workplace equity by identifying and addressing constraints and privileges*,
Promotors: Prof. D.L. van Knippenberg & Dr. A. Nederveen Pieterse,
EPS-2017-421-ORG, https://repub.eur.nl/pub/99311

Kim, T.Y., *Data-driven Warehouse Management in Global Supply Chains*,
Promotors: Prof. R. Dekker & Dr. C. Heij,
EPS-2018-449-LIS, https://repub.eur.nl/pub/109103

Klitsie, E.J., *Strategic Renewal in Institutional Contexts: The paradox of embedded agency*,
Promotors: Prof. H.W. Volberda & Dr. S. Ansari,
EPS-2018-444-S&E, https://repub.eur.nl/pub/106275

Koolen, D., *Market Risks and Strategies in Power Systems Integrating Renewable Energy*,
Promotors: Prof. W. Ketter & Prof. R. Huisman,
EPS-2019-467-LIS, https://repub.eur.nl/pub/115655

Kong, L., *Essays on Financial Coordination*,
Promotors: Prof. M.J.C.M. Verbeek, Dr. D.G.J. Bongaerts & Dr. M.A. van Achter,
EPS-2019-433-F&A, https://repub.eur.nl/pub/114516

Kyosev, G.S., *Essays on Factor Investing*,
Promotors: Prof. M.J.C.M. Verbeek & Dr. J.J. Huij,
EPS-2019-474-F&A, https://repub.eur.nl/pub/116463

Lamballais Tessensohn, T., *Optimizing the Performance of Robotic Mobile Fulfillment Systems*, Promotors: Prof. M.B.M. de Koster, Prof. R. Dekker & Dr. D. Roy, EPS-2019-411-LIS, https://repub.eur.nl/pub/116477

Leung, W.L., *How Technology Shapes Consumption: Implications for Identity and Judgement*, Promotors: Prof. S. Puntoni & Dr. G. Paolacci, EPS-2019-485-MKT, https://repub.eur.nl/pub/117432

Li, X., *Dynamic Decision Making under Supply Chain Competition*, Promotors: Prof. M.B.M. de Koster, Prof. R. Dekker & Prof. R.A. Zuidwijk, EPS-2018-466-LIS, https://repub.eur.nl/pub/114028

Liu, N., *Behavioral Biases in Interpersonal Contexts*, Promotors: Prof. A. Baillon & Prof. H. Bleichrodt, EPS-2017-408-MKT, https://repub.eur.nl/pub/95487

Maas, A.J.J., *Organizations and their external context: Impressions across time and space*, Promotors: Prof. P.P.M.A.R. Heugens & Prof. T.H. Reus, EPS-2019-478-S&E, https://repub.eur.nl/pub/116480

Maira, E., *Consumers and Producers*, Promotors: Prof. S. Puntoni & Prof. C. Fuchs, EPS-2018-439-MKT, https://repub.eur.nl/pub/104387

Mirzaei, M., *Advanced Storage and Retrieval Policies in Automated Warehouses*, Promotors: Prof. M.B.M. de Koster & Dr. N. Zaerpour, EPS-2020-490-LIS, https://repub.eur.nl/pub/125975

Nair, K.P., *Strengthening Corporate Leadership Research: The relevance of biological explanations*, Promotors: Prof. J. van Oosterhout & Prof. P.P.M.A.R. Heugens, EPS-2019-480-S&E, https://repub.eur.nl/pub/120023

Nullmeier, F.M.E., *Effective contracting of uncertain performance outcomes: Allocating responsibility for performance outcomes to align goals across supply chain actors*, Promotors: Prof. J.Y.F. Wynstra & Prof. E.M. van Raaij, EPS-2019-484-LIS, https://repub.eur.nl/pub/118723

Okbay, A., *Essays on Genetics and the Social Sciences*, Promotors: Prof. A.R. Thurik, Prof. Ph.D. Koellinger & Prof. P.J.F. Groenen, EPS-2017-413-S&E, https://repub.eur.nl/pub/95489

Peng, X., *Innovation, Member Sorting, and Evaluation of Agricultural Cooperatives*, Promotor: Prof. G.W.J. Hendriks, EPS-2017-409-ORG, https://repub.eur.nl/pub/94976

Petruchenya, A., *Essays on Cooperatives: Emergence, Retained Earnings, and Market Shares*, Promotors: Prof. G.W.J. Hendriks & Dr. Y. Zhang, EPS-2018-447-ORG, https://repub.eur.nl/pub/105243

Plessis, C. du, *Influencers: The Role of Social Influence in Marketing*, Promotors: Prof. S. Puntoni & Prof. S.T.L.R. Sweldens, EPS-2017-425-MKT, https://repub.eur.nl/pub/103265

Pocock, M., *Status Inequalities in Business Exchange Relations in Luxury Markets*, Promotors: Prof. C.B.M. van Riel & Dr. G.A.J.M. Berens, EPS-2017-346-ORG, https://repub.eur.nl/pub/98647

Pozharliev, R., *Social Neuromarketing: The role of social context in measuring advertising effectiveness*, Promotors: Prof. W.J.M.I. Verbeke & Prof. J.W. van Strien, EPS-2017-402-MKT, https://repub.eur.nl/pub/95528

Qian, Z., *Time-Varying Integration and Portfolio Choices in the European Capital Markets*, Promotors: Prof. W.F.C. Verschoor, Prof. R.C.J. Zwinkels & Prof. M.A. Pieterse-Bloem, EPS-2020-488-F&A, https://repub.eur.nl/pub/124984

Reh, S.G., *A Temporal Perspective on Social Comparisons in Organizations*, Promotors: Prof. S.R. Giessner, Prof. N. van Quaquebeke & Dr. C. Troster, EPS-2018-471-ORG, https://repub.eur.nl/pub/114522

Riessen, B. van, *Optimal Transportation Plans and Portfolios for Synchromodal Container Networks*, Promotors: Prof. R. Dekker & Prof. R.R. Negenborn, EPS-2018-448-LIS, https://repub.eur.nl/pub/105248

Romochkina, I.V., *When Interests Collide: Understanding and modeling interests alignment using fair pricing in the context of interorganizational information systems*, Promotors: Prof. R.A. Zuidwijk & Prof. P.J. van Baalen, EPS-2020-451-LIS, https://repub.eur.nl/pub/127244

Schie, R.J.G. van, *Planning for Retirement: Save More or Retire Later?*, Promotors: Prof. B.G.C. Dellaert & Prof. A.C.D. Donkers, EOS-2017-415-MKT, https://repub.eur.nl/pub/100846

Schouten, K.I.M., *Semantics-driven Aspect-based Sentiment Analysis*, Promotors: Prof. F.M.G. de Jong, Prof. R. Dekker & Dr. F. Frasincar, EPS-2018-453-LIS, https://repub.eur.nl/pub/112161

Sihag, V., *The Effectiveness of Organizational Controls: A meta-analytic review and an investigation in NPD outsourcing*, Promotors: Prof. J.C.M. van den Ende & Dr. S.A. Rijsdijk, EPS-2019-476-LIS, https://repub.eur.nl/pub/115931

Smolka, K.M., *Essays on Entrepreneurial Cognition, Institution Building and Industry Emergence*, Promotors: Prof. P.P.M.A.R. Heugens & Prof. J.P. Cornelissen, EPS-2019-483-S&E, https://repub.eur.nl/pub/118760

Straeter, L.M., *Interpersonal Consumer Decision Making*, Promotors: Prof. S.M.J. van Osselaer & Dr. I.E. de Hooge, EPS-2017-423-MKT, https://repub.eur.nl/pub/100819

Stuppy, A., *Essays on Product Quality*, Promotors: Prof. S.M.J. van Osselaer & Dr. N.L. Mead, EPS-2018-461-MKT, https://repub.eur.nl/pub/111375

Subaşi, B., *Demographic Dissimilarity, Information Access and Individual Performance*, Promotors: Prof. D.L. van Knippenberg & Dr. W.P. van Ginkel, EPS-2017-422-ORG, https://repub.eur.nl/pub/103495

Suurmond, R., *In Pursuit of Supplier Knowledge: Leveraging capabilities and dividing responsibilities in product and service contexts*, Promotors: Prof. J.Y.F. Wynstra & Prof. J. Dul, EPS-2018-475-LIS, https://repub.eur.nl/pub/115138

Toxopeus, H.S., *Financing sustainable innovation: From a principal-agent to a collective action perspective*, Promotors: Prof. H.R. Commandeur & Dr. K.E.H. Maas, EPS-2019-458-S&E, https://repub.eur.nl/pub/114018

Turturea, R., *Overcoming Resource Constraints: The Role of Creative Resourcing and Equity Crowdfunding in Financing Entrepreneurial Ventures*, Promotors: Prof. P.P.M.A.R. Heugens, Prof. J.J.P. Jansen & Dr. I. Verheul, EPS-2019-472-S&E, https://repub.eur.nl/pub/112859

Valboni, R., *Building Organizational (Dis-)Abilities: The impact of learning on the performance of mergers and acquisitions*, Promotors: Prof. T.H. Reus & Dr. A.H.L. Slangen, EPS-2020-407-S&E, https://repub.eur.nl/pub/125226

Vandic, D., *Intelligent Information Systems for Web Product Search*, Promotors: Prof. U. Kaymak & Dr. F. Frasincar, EPS-2017-405-LIS, https://repub.eur.nl/pub/95490

Verbeek, R.W.M., *Essays on Empirical Asset Pricing*, Promotors: Prof. M.A. van Dijk & Dr. M. Szymanowska, EPS-2017-441-F&A, https://repub.eur.nl/pub/102977

Visser, T.R., *Vehicle Routing and Time Slot Management in Online Retailing*, Promotors: Prof. A.P.M. Wagelmans & Dr. R. Spliet, EPS-2019-482-LIS, https://repub.eur.nl/pub/120772

Vlaming, R. de., *Linear Mixed Models in Statistical Genetics*,
Promotors: Prof. A.R. Thurik, Prof. P.J.F. Groenen & Prof. Ph.D. Koellinger,
EPS-2017-416-S&E, https://repub.eur.nl/pub/100428

Vries, H. de, *Evidence-Based Optimization in Humanitarian Logistics*,
Promotors: Prof. A.P.M. Wagelmans & Prof. J.J. van de Klundert,
EPS-2017-435-LIS, https://repub.eur.nl/pub/102771

Wang, R., *Corporate Environmentalism in China*,
Promotors: Prof. P.P.M.A.R. Heugens & Dr. F. Wijen,
EPS-2017-417-S&E, https://repub.eur.nl/pub/99987

Wang, R., *Those Who Move Stock Prices*,
Promotors: Prof. P. Verwijmeren & Prof. S. van Bekkum,
EPS-2019-491-F&A, https://repub.eur.nl/pub/129057

Wasesa, M., *Agent-based inter-organizational systems in advanced logistics operations*,
Promotors: Prof. H.W.G.M. van Heck, Prof. R.A. Zuidwijk & Dr. A.W. Stam,
EPS-2017-LIS-424, https://repub.eur.nl/pub/100527

Wessels, C., *Flexible Working Practices: How Employees Can Reap the Benefits for Engagement and Performance*,
Promotors: Prof. H.W.G.M. van Heck, Prof. P.J. van Baalen & Prof. M.C. Schippers,
EPS-2017-418-LIS, https://repub.eur.nl/

Wiegmann, P.M., *Setting the Stage for Innovation: Balancing Diverse Interests through Standardisation*, Promotors: Prof. H.J. de Vries & Prof. K. Blind,
EPS-2019-473-LIS, https://repub.eur.nl/pub/114519

Wijaya, H.R., *Praise the Lord!: Infusing Values and Emotions into Neo-Institutional Theory*,
Promotors: Prof. P.P.M.A.R. Heugens & Prof. J.P. Cornelissen,
EPS-2019-450-S&E, https://repub.eur.nl/pub/115973

Williams, A.N., *Make Our Planet Great Again: A Systems Perspective of Corporate Sustainability*, Promotors: Prof. G.M. Whiteman & Dr. S. Kennedy,
EPS-2018-456-ORG, https://repub.eur.nl/pub/111032

Witte, C.T., *Bloody Business: Multinational investment in an increasingly conflict-afflicted world*, Promotors: Prof. H.P.G. Pennings, Prof. H.R. Commandeur & Dr. M.J. Burger,
EPS-2018-443-S&E, https://repub.eur.nl/pub/104027

Ye, Q.C., *Multi-objective Optimization Methods for Allocation and Prediction*,
Promotors: Prof. R. Dekker & Dr. Y. Zhang,
EPS-2019-460-LIS, https://repub.eur.nl/pub/116462

Yuan, Y., *The Emergence of Team Creativity: a social network perspective*,
Promotors: Prof. D.L. van Knippenberg & Dr. D.A. Stam,
EPS-2017-434-ORG, https://repub.eur.nl/pub/100847

Zhang, Q., *Financing and Regulatory Frictions in Mergers and Acquisitions*,
Promotors: Prof. P.G.J. Roosenboom & Prof. A. de Jong,
EPS-2018-428-F&A, https://repub.eur.nl/pub/103871

A railway timetable states at what times trains arrive or depart at certain locations in the railway network. The design of a timetable is an extremely complex puzzle, and years of investigations are necessary to design a timetable from scratch. Amongst several other aspects, planners should take the travel demand, connections between trains, capacity on the tracks and in the train, and daily disturbances into account when designing a timetable. This can easily lead to the situation that there are too many restrictions that a timetable has to satisfy, such that no longer a timetable can exist satisfying all these restrictions.

In the first part of this thesis, methods are developed that can support the strategic, long-term design of a timetable. Timetables are computed that match with travel demand as good as possible, without taking infrastructure capacity into account. Using these ideal timetables, one can make clear whether regular departure patterns are useful or not, and how this relates to the expected travel time of passengers. Another method tries to find a timetable that can be operated on a given infrastructure network, and that is as similar as possible to the ideal timetable.

The second part of this thesis is oriented towards short-term timetabling. First of all, a method is developed that deals with the situation in which there are too many restrictions that a timetable has to satisfy. It finds relaxations to these restrictions, such that a feasible timetable can exist. This can be used when additional trains are scheduled into an existing network. Another approach aims at designing a timetable that is robust against minor disturbances that can occur in the real-life operation. This helps in deciding where to add time supplements in the network to absorb delays.

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

ERIM PhD Series
Research in Management

Erasmus University Rotterdam (EUR)
Erasmus Research Institute of Management
Mandeville (T) Building
Burgemeester Oudlaan 50
3062 PA Rotterdam, The Netherlands

P.O. Box 1738
3000 DR Rotterdam, The Netherlands
T +31 10 408 1182
E info@erim.eur.nl
W www.erim.eur.nl