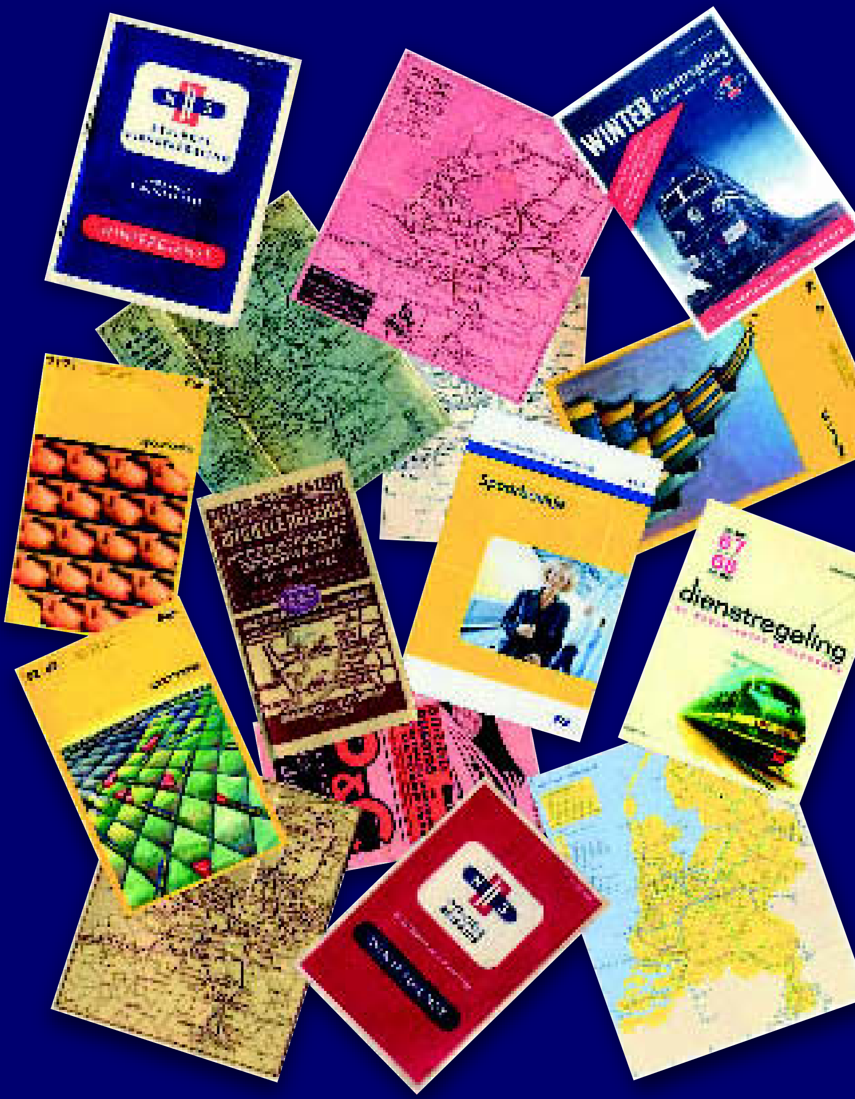
LEON W.P. PEETERS

Cyclic Railway Timetable Optimization



Cyclic Railway Timetable Optimization

Leon Peeters

Cyclic Railway Timetable Optimization

Optimalisatie van cyclische spoorwegdienstregelingen

Proefschrift

ter verkrijging van de graad van doctor aan de
Erasmus Universiteit Rotterdam
op gezag van de
Rector Magnificus
Prof.dr.ir. J.H. van Bemmel
en volgens besluit van het College voor Promoties.

De openbare verdediging zal plaatsvinden op
vrijdag 6 juni 2003 om 11.00 uur

door

Leon Willem Petrus Peeters

geboren te Tegelen

Promotiecommissie

Promotoren:	Prof.dr. L.G. Kroon
	Prof.dr.ir. J.A.E.E. van Nunen
Overige leden:	Prof.dr. S.L. van de Velde
	Prof.dr. A.P.M. Wagelmans
	Prof.dr.ing. I.A. Hansen

Cover illustration: Netherlands Railways timetable booklets (1922–2003)

Erasmus Research Institute of Management (ERIM)
Erasmus University Rotterdam
Internet: http://www.erim.nl
ERIM Electronic Series Portal: http://hdl.handle.net/1765/1

ERIM Ph.D series Research in Management 22
TRAIL Thesis series T3003/5, The Netherlands TRAIL Research School

ISBN 90–5892–042–9

Acknowledgements

As sometimes happens to trains as well, this thesis on railway timetabling has arrived a bit later than scheduled. During the last years, a number of people have acted as a sort of traffic control, and contributed to keeping the delay within limits. More importantly, they have made it a truly enjoyable ride for me.

First, I want to thank my promotor Leo Kroon. With seemingly endless patience, he read each draft of the thesis in detail, and always had suggestions for improvements. I genuinely enjoyed our Friday afternoon whiteboard brainstorm sessions in Rotterdam, and later, when I was in Konstanz, our email discussions on the mathematical details of the thesis. His mathematical intuition not only benefited the proofs in the thesis, it also proved most useful for maneuvering us through the heavy Utrecht-Rotterdam traffic in his red Alfa Romeo, when late for an appointment at the Erasmus University.

I clearly remember first meeting my other promotor Jo van Nunen. During the interview for the PhD position, he managed to lighten my nervosity by noting that he grew up close to my birth town, and hinting at the encounters he used to have with its female inhabitants at the annual funfair. On a more serious note, I am thankful for his remarks, which certainly contributed to improving the structure of the thesis.

From the start, I felt at home at the pleasant and friendly atmosphere of the Department of Decision and Information Sciences. During my first year, I shared a small office with Otto Koppius. Both with a background in mathematics, we dove into our first management courses. Most, I remember the almost unproductive amount of fun we had during our first year, and I share his concern on the separate offices. My second roommate, Michiel Vromans, was an invaluable source on the most exotic details of any railway related subject whatsoever. Loli Romero Morales provided an extensive introduction to Spanish culture, starting with croquetas, and with the current high point of a field trip to Sevilla. I further recall Jelle's numerous attempts to explain how simple things are, drawing diagrams for Chantal on my whiteboard, and evenings in the Smitse that frequently ended later than foreseen.

Supported by her extensive tea collection, Iris Vis and I had many a discussion on automated guided vehicles. She further showed me that meat is not a necessary ingredient for a delicious dinner.

During the first two years of my PhD period, I spent quite some time at Railned. There, Jurjen, Dick, Joek, and Eric taught me a lot about the art of timetable construction. With Mieke van den Braber, who was doing a master's thesis internship at Railned, I collaborated on investigating cycle bases for cyclic railway timetabling. At NS Reizigers, I thank Peter Baas for his interest in my research, and for his clear explanations on timetable design.

Through a scholarship of the European Commission's AMORE network, I spent the last year of my PhD period visiting Dorothea Wagner at the University of Konstanz. I want to thank Dorothea for inviting me to Konstanz, and for giving me the opportunity to finish my research on cycle bases, and to complete writing the thesis. She and my other colleagues at the Algorithms and Data Structures group immediately made me feel at home in Konstanz. In particular, Frank helped me to find nice bars and a place to stay, Thomas introduced me to Linux, Sabine was a great cycle basis resource, Flavia made the best Italian coffee in Konstanz, Ulrik taught me about 3D visualization, Marco proved to be a great Santa Claus, Thomas a great snowboard tour guide, and Csaba provided the Hungarian touch.

At the first AMORE workshop, Christian Liebchen showed interest in the small triangles I was drawing on a small piece of paper during one of the presentations. The discussion that followed was the start of our collaboration on cyclic railway timetabling. He regularly infected me with his enthusiasm, shown in times of inspiration through email bombardments.

Through my PhD period, Patrick Meersmans and Jan-Willem Goossens have been my brothers in both research and drinks. Patrick provided me with shelter on my first nights in Rotterdam, and has been a friend and colleague ever since our Maastricht time. However, I still haven't gotten used to his habit of employing an electric razor as an alarm clock. Jan-Willem has been an incredibly valuable C++ resource, as well as a companion in outdoor mountain sports. Unfortunately, most of the brilliant ideas we together developed in Konstanz' bars had lost quite a bit of their brilliance by the next morning.

Finally, I thank Nathalie for her support, especially during the last months when I spent most of my time at the university. Patiently, she managed to be there for me in the spare moments between writing. Und das war wahrscheinlich nicht ganz einfach für jemanden mit deinem sensiblen Geduldsfaden! Weiterhin danke Ich der ganzen Zumsteinstrassen-WG für's Durchfuttern bei verschiendensten herrlichen Gerichten.

Het allerletste woord rich ik aan mien elders. Auk al heb ik mich de aafgelaupe jaore met logistieke planning bezig gehalde, ik bin altied onder de indruk gewaes van och logistieke expertise. Die toende zich beej mien diverse verhoezinge: van Baolder naor Bels, van Bels naor Mestreech, van Mestreech naor Leidschendam en truuk, daornao van Mestreech naor Rotterdam, en oeteindelik van Rotterdam via Boalder naor Konstanz. Waorlik hoegtepunte van zoewaal route- as auk inpak-optimalisatie. Maar eigelik steit die verhoezingshulp symbool veur och leefde en steun veur mich daor de jaore heer. Pap en mam, bedank!

Leon Peeters
Konstanz, April 2003

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis studies mathematical models and solution methods for constructing high quality cyclic railway timetables. In a cyclic timetable, train connections are operated regularly with respect to a cycle time. So, a train for a certain destination leaves a certain station at the same time every cycle time, say every half an hour, every hour, or every two hours. Cyclic timetables are mainly used for passenger railways, though cargo rail schedules are sometimes also cyclic to some extent. This thesis therefore focuses on passenger railway timetables.

The railways, and railway timetables in particular, have been a frequently discussed topic in the Netherlands the last years. A decade of steadily growing mobility has led to congested railways and highways. And, with an ongoing economic and demographic growth, the Dutch Ministry of Transportation expects the upward trend in mobility demand to continue in the near future (V&W, 2001)[1]. This is illustrated in Figure 1.1, which shows the expected growth in mobility for the Netherlands until 2020. As a reference point, the 1995 mobility level is set to 100, and estimates are shown for the mobility level in 2020 for three scenarios. The 'no policy' scenario represents a governmental laissez-faire approach to the mobility growth, 'current policy' represents a scenario in which the currently planned and decided upon governmental policies are carried out, and 'public transportation policy' represents the scenario in which the government stimulates the use of public transportation.

With market shares of 13% of the traveled kilometers, and 5% of the number of trips, public transportation takes a modest share in the total Dutch passenger transportation market. Car travel is still dominant in the Netherlands, with an important second place for bicycle travel and foot travel for short distances (V&W, 2001). However, public transportation should be seen as a complement to other

[1] Ministerie van Verkeer en Waterstaat, the Dutch Ministry of Transportation.

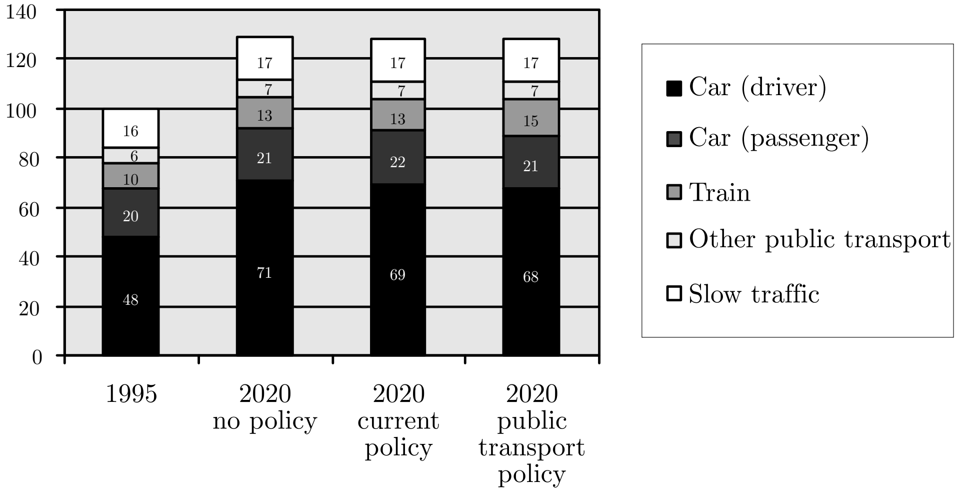


Figure 1.1: Expected mobility growth for the Netherlands (source: AVV website, 2002)

modes of transportation, not as a substitute. By bundling transportation flows, public transportation has traditionally played an important role in keeping urban areas accessible. Forecasts from the AVV[2] predict most future travel in the Netherlands to take place within, and to and from, the so-called Randstad area, the congested metropolitan area bounded by the four largest Dutch cities Amsterdam, Rotterdam, The Hague, and Utrecht (AVV, 2000). It is expected that exactly in this area a growth of public transportation will provide extra transportation capacity. Moreover, public transportation has the function of providing all people with transportation, thereby enabling everybody access to social and economic activities.

Railway transportation is the main mode of public transportation in the Netherlands, especially for medium to long distances. Bus and metro are primarily used for short to medium distances, or as feeder or distributor for the railway system (AVV, 2000). So, the performance of public transportation as a whole depends largely on the performance of the railway system.

The introduction of competition in the European railway market is a second development that puts the railways in the spotlight. In 1995, the state-owned NS[3] were split into an infrastructure managing part and a railway operating part. The infrastructure managing task units, such as Railned, were brought together in the governmental organization Prorail. The railway operating business units, such as

[2] Adviesdienst Verkeer en Vervoer, the advisory unit on traffic and transport of the V&W.

[3] Nederlandse Spoorwegen (Netherlands Railways).

the passenger railway operator NSR[4], are commercially operating organizations[5]. Since the introduction of competition, some new operators have entered the railway transportation market, such as the passenger railway operators Noordned and Syntus, and the cargo operators Shortlines, Railion, and ACTS.

Finally, the railways in the Netherlands have been criticized the past years because of a less than optimal operational performance, and because of related quality problems. With few investments in railway infrastructure in the 1970's and 1980's, a lot of delayed maintenance had built up when in the 1990's railway investments were resumed. Together with the increased mobility and the resulting highly intensive use of the infrastructure, this has led to infrastructure failures. These disruptions frequently resulted in delayed trains, and especially in peak hours such delays may be knocked on to other trains and propagated through the network. Besides these infrastructure issues, there have also been railway operator related problems. Too few trains were available to meet the increased demand for railway travel, which resulted in postponed maintenance and over-utilized trains. This, in turn, led to additional delays caused by the breakdowns of trains. Through a highly dependent crews schedule, delayed trains resulted in the late arrival of train crews, and thus in the delayed departure of the crew's subsequent train to be operated. Over-crowded trains, high pressure on the on-train crews, and the related crews strikes are other sources of criticism. Concluding, the current Dutch railway system uses its infrastructure intensively, and faces disruptions and punctuality problems.

1.1 Current Developments in Passenger Railways

The basic wishes of the modern railway customer are fairly simple: he or she wants to travel fast and comfortably for a reasonable price. Additionally, railway travel should be transparent and reliable, and should provide a choice of service level and comfort level.

The Dutch government aims at a 2020 railway system with high frequencies on the Randstad backbone network, and high velocities on long distance connections (V&W, 2001). High frequencies ensure good connections to pre-transportation and post-transportation, and to the regional transportation systems outside the Randstad. On long distance connections, there will be dedicated infrastructure for high-speed and traditional trains. The resulting homogenization of traffic velocities helps in solving the above mentioned capacity problems, because the railway infrastructure is utilized more efficiently. We refer to Section 2.4 for more details on the impact of

[4] Nederlandse Spoorwegen Reizigers (Netherlands Railways Travelers).

[5] At the time of writing, all shares of the business units are still owned by the Dutch government.

homogenizing railway traffic.

A study performed by the AVV found that public transportation forms an attractive mode of transportation when its travel times are competitive to the car travel times (AVV, 2000). In particular, public transportation travel times are competitive when highways are congested, and public transportation operates smoothly and at high frequencies, with short post-transportation times to, for example, office buildings and shopping centers. This implies that the strength of public transportation lies in the medium to long distance railway connections between urban centers. The same study identifies the travel within and around urban centers as a new and growing market for public transportation, in which the bulk of the number of moves takes place.

The NS product management department (NS, 2001) identifies the same key growth markets of national long distance travel, inter-urban medium distance travel, and urban travel for distances of ±40 km. International travel, though relatively small in size, is also seen as a key market, since it influences the development and image of the railway market as a whole. NS bases its product development for the period 2010–2020 as follows on these key markets, customer wishes, and the objectives of the Dutch government. Since travel time is a key criterium for choosing a mode of transportation, the train velocities and railway connections are differentiated. This differentiation is achieved by a high-speed network, an intercity network, and a sprinter network, which serve the markets of national travel, inter-urban travel, and urban travel, respectively. Each of the three networks has its formula, defining its velocity, comfort level, and service level. The three networks serve geographically different markets, and mostly use dedicated infrastructure. Therefore, the differentiation in velocity does not conflict with the above mentioned homogenization of traffic velocity. Stations take the role of switches between the networks. Finally, the basic principles of the new timetable follow the basic wishes of the customer.

The structure of the current Dutch railway timetable stems from the early 1970's, when the cyclic timetable with many connecting train services was introduced. All new developments since, such as the introduction of new train lines and new stations, have been incorporated into this basic structure. The future NS timetable concept aims at the following goals (NS, 2001)

- reducing travel times by higher velocities, briefer dwelling, faster acceleration and deceleration, and higher frequencies,
- increasing reliability and punctuality by a less interdependent timetable, and no splitting and combining of trains,
- improving the transparency of the timetable by evenly distributing the trains

over the hour.

These goals, together with the railway infrastructure capacity problems caused by the growing mobility, require fundamental changes in the current railway timetable.

1.2 Aim and Relevance of the Thesis

The aim of this thesis is to investigate mathematical models and solution methods for constructing high quality cyclic railway timetables. As such, this research provides a valuable contribution to existing mathematical timetabling methods, which tend to focus on the problem of constructing a timetable without explicitly taking an objective into account. Moreover, in the practical construction of cyclic railway timetables, objectives are clearly of great importance.

The main research question studied in this thesis is the following:

> *How can mathematical models and solution methods support the construction of high quality cyclic railway timetables?*

To that end, we investigate important criteria for assessing the quality of a timetable, and how these criteria can be incorporated into mathematical models. In particular, we consider the objectives of minimizing passenger travel times, maximizing timetable robustness, and minimizing the required number of rolling stock compositions. Moreover, we study over-specified timetable requirements, by allowing a penalized violation of the requirements. In such a case, the objective is to minimize the violation of the initially specified requirements. The construction of high quality cyclic railway timetables is a complex and time consuming process. Therefore, reducing the computation times of the proposed solution methods is also a research topic of the thesis. Summarizing, the main research question breaks down into the following sub-questions:

> *What are the criteria for assessing the quality of a timetable, and how can they be modeled?*

> *What adjustments need to be made to over-specified defined timetable requirements in order to obtain a feasible timetable, and how can these adjustments be modeled?*

> *How can the models arising from the previous two questions be solved in a reasonable amount of time?*

1.2.1 Social Relevance

The background setting discussed at the start of this chapter directly indicates the social relevance of studying cyclic railway timetables. Railway timetabling models and algorithms offer a method for improving the quality of the railway timetable, and thus of the railway system as a whole. Clearly, railway passengers benefit from quality improvements in the railway system. Moreover, an improved railway system may offer current users of private transportation a viable alternative mode of transportation. Therefore, this research may contribute to solving the earlier sketched mobility problems. Next, we discuss the social impact of the thesis in more detail.

Railway infrastructure investments involve tremendous amounts of public money, and have a long life cycle. Tools that assist the government in making infrastructure investment decisions are therefore highly valuable. Railway timetabling models provide such a tool for evaluating various alternatives for infrastructure investments. Capacity, robustness, and stability are common evaluation criteria in such a railway infrastructure study.

The opening of the European railway market to competition has lead to multiple operators, who are using the railway network simultaneously. The allocation of railway capacity to the various operators can be seen as the construction of a combined timetable for the entire railway system. An optimization method for railway timetabling offers infrastructure managers a tool to construct such a combined timetable. The combined timetable should be optimal for some overall criterium reflecting the benefits of the timetable to society, for example, fast travel or stability. Here, the possibility to quantify trade-offs between various criteria plays an important part.

Concluding, railway system improvements may stem from the above described decision aid for governmental infrastructure investments, or from assisting in the assignment of railway capacity in a competitive railway market. The next section describes how the thesis may improve the strategic and tactical planning processes of railway companies. In a competitive railway market, such improvements are expected to lead to a quality improvement of the railway system.

1.2.2 Managerial Relevance

By developing fast railway timetabling methods, the thesis offers railway managers a tool for faster and better founded tactical and strategic decision making. At the tactical level, timetable decision making involves investigating whether major changes or minor adjustments to the current timetable are feasible. So, a theoretical indication can be obtained of the effects of such changes or adjustments on the core timetable, without actually implementing them. At the strategic level, the research findings assist in exploring new timetable concepts for a 15 to 25 year time horizon.

A reduction of the time span for constructing a timetable enables railway managers to react faster and more accurately to developments in the transportation market. Good timetabling models and algorithms yield such a reduction of the timetabling time span. As an example, consider the DONS[6] timetabling decision support system, that NS started to develop in 1992. DONS supports the planning of a cyclic core timetable for strategic and tactical purposes. The introduction of DONS at NSR and Railned[7] reduced the lead time for constructing a cyclic timetable for the whole of the Netherlands significantly (see Hooghiemstra, 1996, Hooghiemstra et al., 1999). This lead time reduction is for the construction of a fairly *rough* timetable for tactical and strategic studies. The construction of a *detailed* timetable for the whole of the Netherlands still takes a considerable amount of time.

Moreover, the lead time reduction offers the opportunity to carry out scenario studies. Such scenario studies are hardly feasible when the construction of a single timetable already takes several weeks. Also, timetable optimization offers quantitative decision support for scenario studies, since it allows railway managers to quantify the trade-off between different criteria, such as passenger satisfaction and cost efficiency.

Finally, the timetable planning process is closely connected to other railway planning processes, such as train line planning, rolling stock scheduling, crew scheduling, and infrastructure planning. Timetable construction serves as an evaluation tool for some of these processes, and for other processes the timetable is one of the inputs. As such, any improvements in the timetable planning process also impact these other processes. We refer to Section 2.3 for an in-depth discussion of the dependence between timetable planning and other railway planning processes.

1.3 An Introduction to Cyclic Railway Timetables

In 1931, NS presented the cyclic railway timetable concept, with a cycle time of one hour. The cyclic nature of the 2001/2002 NSR timetable is illustrated in Table 1.1. This table shows the timetable for the direct intercity connection between the Dutch cities of The Hague and Venlo, which is operated by the 1900 intercity train, see Figure 1.2. The rows in the table show that the connection is operated every hour at exactly the same minute. The first train of the day is the only exception, leaving

[6] Designer Of Network Schedules.

[7] The development of DONS was started by the then state-owned NS. Since the split of NS, DONS has been in use at both the passenger operator NSR and the capacity manager Railned.

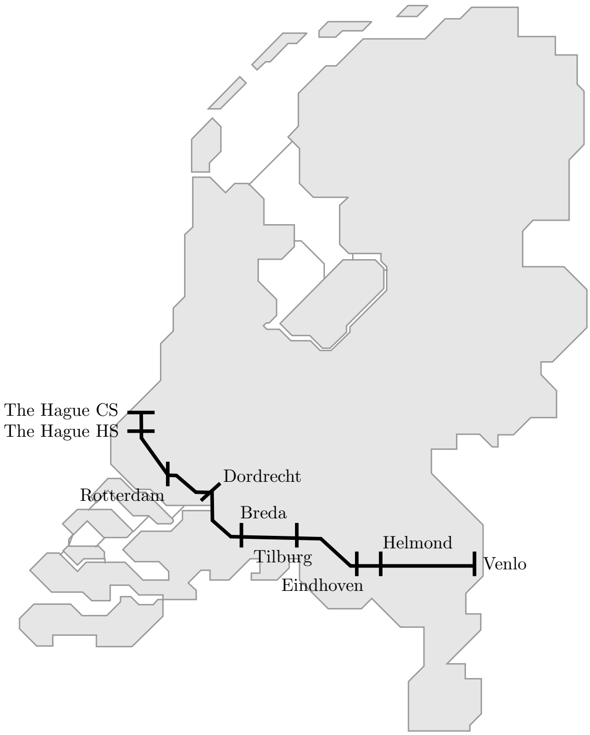


Figure 1.2: The 1900 train from The Hague to Venlo

The Hague CS	-	06:21	07:21	...	20:21	21:21	22:21
The Hague HS	-	06:25	07:25	...	20:25	21:25	22:25
Rotterdam	05:41	06:45	07:45	...	20:45	21:45	22:45
Dordrecht	06:01	07:01	08:01	...	21:01	22:01	23:01
Breda	06:19	07:19	08:19	...	21:19	22:19	23:19
Tilburg	06:33	07:33	08:33	...	21:33	22:33	23:33
Eindhoven	07:01	08:01	09:01	...	22:01	23:01	23:58*
Helmond	07:11	08:11	09:11	...	22:11	23:11	-
Venlo	07:32*	08:32*	09:32*	...	22:32*	23:32*	-

Table 1.1: The 2001/2002 weekday timetable for the 1900 intercity train from The Hague to Venlo (*indicates an arrival time)

from Rotterdam rather than The Hague, at 05:41, four minutes earlier than its cyclic departure time. The last train does conform to the cyclic departure times, but already ends in Eindhoven, not in Venlo.

Since the introduction of cyclic timetables in the Netherlands more than seventy years ago, many other European countries have adopted the concept. Nowadays, cyclic timetables are operated in Austria, Belgium, Denmark, Germany, Great Britain, Norway, and Switzerland. Moreover, many bus and metro systems operate cyclic timetables, though often with smaller cycle times than railway systems do.

Cyclic timetables have several advantages. Since they are transparent to the customer, there is no need for passengers to memorize complex timetables for their regular connections. It suffices to memorize the minutes of the hour at which the trains leave. Moreover, if the cyclic concept is strictly adhered to, passengers face no gaps in train services on parts of the day when transportation demand is low.

From a planning point of view, cyclic timetables have the advantage that one only needs to consider one cycle period. A crucial extra requirement, then, is that the situation at the end of the cycle period matches the situation at the start of the period. If the start and end situations match, the basis of a full-day timetable can be obtained by copying the cyclic timetable for all relevant hours of the day. This copy procedure yields a full-day timetable in which the departure times, in minutes, are the same for each train connection and for each cycle period. Clearly, this basic full-day timetable still needs some adjustments, for example, for rush hour traffic or late evening traffic.

A final and related advantage is that cyclic timetables can be represented compactly, that is, only a single cycle period needs to be shown. Cyclic timetables are therefore relatively easy to handle within the railway company itself. A railway operator can focus its planning on one cycle period, not only for the timetable itself, but also for depending plans, such as rolling stock planning and crew planning. However, this does not imply that the depending plans themselves are also cyclic.

At the other end of the timetable spectrum lies the a-cyclic timetable, which is sometimes referred to as the market-led timetable (Ford and Haydock, 1992). In an a-cyclic timetable, a railway operator fine-tunes the timetable to the demand for transportation. Such a timetable typically contains many trains during the morning and afternoon rush hours, and few trains in low traffic hours. Further, an a-cyclic timetable tends to offer a quite dense service on Mondays and Fridays, because of post-weekend and pre-weekend traffic, whereas the weekend timetable itself may contain many gaps. Extremely fine-tuning a timetable to market demand may result in a complex timetable that is hard to consult and to memorize. Moreover, the a-cyclic timetable concept requires the planning of a timetable for the entire day, for each day of the week. This typically results in very large planning problems.

On the other hand, cyclic timetables may yield higher costs than a-cyclic ones. In a cyclic timetable, the late evening hours offer the same train service as the rest of the day. But the occupation degree of the late evening trains is typically much less than during the rest of the day. Therefore, a cyclic timetable usually takes market demand into account to a certain extent. Typically, high capacity trains are used during rush hours, that is, longer and/or double-deck trains. And in low-traffic hours, trains may consist of less carriages. Moreover, a hybrid timetable concept is possible, with a cyclic core timetable, in which extra trains are operated during rush hours, and from which trains are removed during low-traffic hours.

1.4 Cyclic Railway Timetabling: A Literature Review

This section reviews previously published research on cyclic railway timetabling. For an overview of mathematical models for railway transportation in general, we refer to Assad (1980). A more recent overview, focusing on discrete optimization models for railway transportation, is described by Bussieck et al. (1997). We do not review the literature on a-cyclic railway scheduling, but refer the interested reader to the survey by Cordeau et al. (1998), and to the recent contributions by Brännlund et al. (1998), Caprara et al. (2001), and Caprara et al. (2002).

Most authors that study cyclic railway timetabling problems use models that are based on the Periodic Event Scheduling Problem (PESP), as introduced by Serafini and Ukovich (1989a). The PESP considers the problem of scheduling a set of periodically recurring events under periodic time window constraints. More specifically, the PESP aims at determining the time instants at which the periodic events are to take place. The periodic restrictions are imposed on pairs of events, and constrain the time interval between the two events to some time window. The basic PESP only deals with the problem of finding a feasible schedule, and does not take an objective function into account. Chapter 4 describes the PESP in detail.

Voorhoeve (1993) first considered a PESP based model for the cyclic railway timetabling problem for NS. His model takes into account the main requirements of Dutch timetables, such as train movement characteristics, safe use of the infrastructure, connections between trains, and limits on dwell times.

Schrijver and Steenbeek (1993, 1994) improved Voorhoeve's model and algorithm by developing a constraint programming based algorithm. Their CADANS algorithm solves the problem of finding a feasible timetable for a set of PESP constraints. The algorithm generally performs well on most Dutch real-life railway timetabling instances, though some instances require very large computation times. They also considered the post-optimization of an obtained feasible timetable. To that end, the train sequences of the obtained timetable are fixed, after which a locally optimal timetable is computed.

Nachtigall (1994, 1996a) also uses PESP constraints to model the cyclic behavior of railway timetables, and takes the objective of minimizing passenger waiting times into account. He transforms the problem into one that is formulated in terms of cycles of PESP constraints. Moreover, he introduces a multi-criteria objective for cyclic railway timetable optimization. He particularly explores bi-criteria objectives, namely the cost of infrastructure versus the benefit of improving the synchronization in the timetable, and rolling stock circulation versus passenger waiting time. In a related paper (Nachtigall, 1996b), cyclic timetables consisting of train lines with different cycle times are studied.

Nachtigall and Voget (1996) also consider the problem of minimizing passenger waiting times in a cyclic railway timetable. They heuristically generate an initial timetable, using ideas from a practical planner's manual timetable construction procedure. This initial timetable is then improved using a genetic algorithm. In a later paper, Nachtigall and Voget (1997) consider the bi-criterium objective of infrastructure investments and passenger waiting time improvements.

Odijk (1996, 1997) uses the PESP model for cyclic railway timetabling, but not for the sole purpose of constructing a timetable. Rather, his aim is to compose specifications for the extension of the infrastructure within and around a station. He proposes to generate a family of timetables for the station. By inspecting such a family of timetables and identifying the features they have in common, he composes specifications for the infrastructure extension.

Liebchen (1998) compares some of the above described methods by applying them to instances from the Berlin underground and urban rail. He considers the work by Nachtigall (1999), Odijk (1997), and Nachtigall and Voget (1996), with different options for the parameters for each of these methods.

Goverde (1999) proposes to use the transformed model by Nachtigall (1999), with the objective function of constructing stable and reliable timetables. So, his objective function favors timetables that are relatively insensitive to minor operating disturbances. To that end, he investigates objective functions, which take into account, to some extent, the stochastic nature of railway operations.

A combination of railway timetabling and railway line planning is studied by Lindner (2000). His research considers constructing a cost optimal train schedule, which is a timetable that minimizes the cost of the corresponding rolling stock plan. For the railway timetabling part of his model, he uses the PESP formulation. He developed a modification of the Serafini and Ukovich (1989a) algorithm, and a Branch&Cut

algorithm for solving the PESP.

Weigand (1983) does not use a PESP model, but considers cycle constraints similar to those in the model by Nachtigall (1999), for minimizing the waiting times of passengers at stations. His solution method picks a spanning tree of the graph induced by the arrival and departure times of the trains, and computes an optimal timetable for that tree. Then, the algorithm iteratively moves to a new spanning tree in order to improve the best found timetable.

Domschke (1989) uses a mathematical formulation that is similar to the Quadratic Assignment Problem (see Burkard et al., 1998, Çela, 1998) to model cyclic railway timetables. His quadratic objective function for minimizing passenger waiting time associates a weight with fixing the departure or arrival times for pairs of trains, and thus with the waiting time for passengers who transfer between the involved trains. Daduna and Voß (1995) further studied this model.

Lichtenegger (1990) considers the problem of the integrated fixed interval timetable, a special type of cyclic timetable. In an integrated fixed interval timetable, used in Austria and Switzerland, trains visit the large stations at fixed and regular time instants with regard to a cycle time. For example, with a cycle time of 60 minutes, trains only visit a large station at time instants 0, 15, 30, and 45. The idea is that at these time instants passengers can change trains, since most trains are then present at a station. Lichtenegger's aim is to minimize the infrastructure investments required to operate an integrated fixed interval timetable. He formulates this problem as a mixed integer program.

Finally, Goverde and Soto y Koelemeijer (2000) consider the problem of evaluating the performance of a cyclic railway timetable. They take a cyclic timetable as input, and use max-plus algebra to determine several performance indicators of the timetable. These performance indicators include the critical circuits in the railway network, the stability margin of the timetable, and the propagation of delays in case of a disruption.

1.5 Outline of the Thesis

The remainder of the thesis is structured as follows. Chapter 2 describes the practical aspects of railway timetable planning. It characterizes the relation between timetable planning and the other planning processes of a railway operator. The impact of timetabling for railway infrastructure planning is illustrated, and the DONS timetabling decision support system is described briefly.

Chapter 3 presents a mathematical model for cyclic railway timetabling. We first state our assumptions and the considered objectives. Then, periodic constraints are introduced as a basic tool for modeling cyclic timetables. We illustrate the modeling power of periodic constraints through extensive examples of practical situations. Next, these examples are formalized in an integer programming model. We then present a more compact constraint graph representation for the integer program, and formalize the objective functions. The chapter closes with a discussion of the structure of railway timetabling constraint graphs.

Chapter 4 relates the model of Chapter 3 to the existing literature on the PESP (Serafini and Ukovich, 1989a). We derive some useful properties of the PESP, and pay special attention to the sequencing of events in the PESP.

Based on the derived properties, Chapter 5 describes a transformation of the PESP, which builds upon the work by Nachtigall (1999). We derive theoretical results and properties for the transformed model. The transformed model is closely related to cycles in graphs. Therefore, part of the chapter is devoted to a study of bases of the cycle space of a graph.

Chapter 6 investigates extensions of the basic railway timetabling model from Chapter 3. These extensions allow for incorporating variable trip times, flexible train connections, and station capacity restrictions into the model. We also present a rolling stock minimization objective function, based on the ideas presented in Chapter 5. Moreover, the chapter considers the relation between mathematical solution methods, and the way in which planners use them in practice. By taking into account how planners construct railway timetables in practice, we simplify parts of our solution methods. Such simplifications lead to heuristic solution methods, that do not necessarily find an optimal solution. However, both the practical and the computational complexity of the timetabling problem justify the use of heuristics.

Chapter 7 reports on the computational experiments that were carried out to test the developed model and solution methods. The test instances are real-life Dutch timetabling problems that were obtained from the DONS system.

Finally, the conclusions of, and a reflection on, the results achieved in the thesis are presented in Chapter 8.

Chapter 2

Planning Railway Timetables

This chapter explores several practical aspects of timetable planning. It so gives an impression of the practical environment in which a railway timetable is constructed, of its input factors, and of the implications of a timetable for the railway system as a whole.

Section 2.1 considers the practical process of railway timetabling, and the various matters that are taken into account in this process. Section 2.2 discusses the process of timetable planning at NSR, the largest Dutch passenger railway operator. Next, Section 2.3 describes several other planning problems that need to be solved for a railway system to function well. These problems include estimating the travel demand, planning what train connections will be operated, scheduling, maintenance, and shunting the railway rolling stock, and crew scheduling and rostering. Timetabling also plays an important part in the railway infrastructure planning process, which is the subject of Section 2.4. Finally, Section 2.5 gives a brief overview of the DONS decision support system for railway timetabling. The DONS system is a practical example of how mathematics and computer science can support the tactical and strategic design of railway timetables.

2.1 Constructing Railway Timetables in Practice

Although intelligent decision support systems for railway timetabling, such as the DONS system described in Section 2.5, do exist, they are generally only used for strategic and tactical studies. Due to its complex nature, the construction of the actual timetable as it is operated in practice, is still mainly a human planning process, which is supported by computer aided design tools.

Traditionally, planners use two types of graphs as the main tools for constructing a timetable. The so-called *time-space diagram* graphically represents the train movements that take place in between stations, on the tracks. One axis displays time, and the other axis depicts space. Throughout this thesis, time is placed on the vertical axis, and space on the horizontal axis.

Figure 2.1 is an example of a time-space diagram of the 2001/2002 timetable for the track between the stations of Rotterdam and Utrecht. The character codes on the horizontal space axis represent stations between Rotterdam and Utrecht, and the numbers on the vertical axis stand for the minutes in an hour. Each line in the time-space diagram depicts a train, which is indicated by the number next to the line. Lines with a positive inclination correspond to trains from Rotterdam to Utrecht, and lines with a negative inclination to trains from Utrecht to Rotterdam. Flat lines indicate fast trains, because those trains cover a large distance in a short time, and steep lines indicate slow trains. When a train dwells for some time at a station, this gives a vertical line at that station, because time moves on while the location of the train remains unchanged. Intersecting lines indicate that the two corresponding trains meet at the point of intersection. This is clearly only allowed at stations, or if the trains are using different tracks.

The time-space diagram does not show any detailed information on the situation inside a station. The *platform-occupation chart* graphically zooms in to the details within a station. Here, time is set on the horizontal axis, and the vertical axis corresponds to the platforms. A platform track is depicted by a line, and a line next to a platform track represents a train occupying that platform track for the corresponding time.

Figure 2.2 provides part of the platform-occupation chart for station Utrecht in the 2001/2002 timetable. Each of the displayed platform tracks has an a-side and a b-side. These sides correspond to the front part and back part of the platform track, which can be used by different trains. The number codes in the graph represent trains, as in the time-space diagram. If both the a-side and the b-side of a platform track are occupied by the same train, this gives blocks in the chart rather than lines. So, each of the trains at platform track 5 occupies the entire platform. Associated with each train occupying a platform track, one sees two character codes. These represent the origin and the destination station of the train.

In practice, a timetable is constructed by specifying, for each train, a time-space path through the railway network, which is drawn in the time-space diagrams, and by specifying the platform tracks that the train occupies, which are drawn in the platform-occupation charts. Generally, a timetable is not constructed from scratch. Rather, adjustments are made to an existing timetable, typically, to the timetable of the previous year. Trains are added to the existing timetable, deleted from it, or the

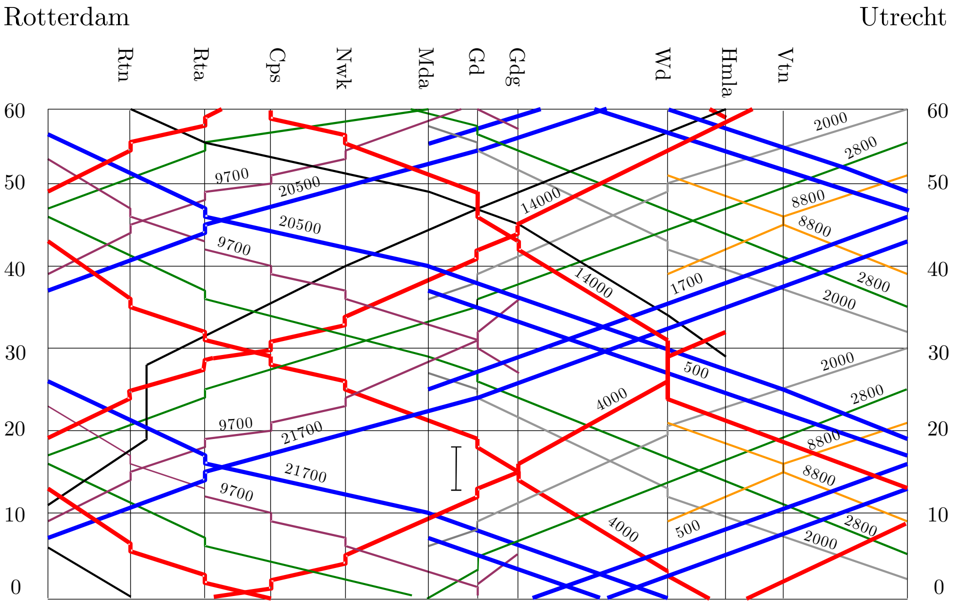


Figure 2.1: Time-space diagram for the track between Rotterdam and Utrecht.

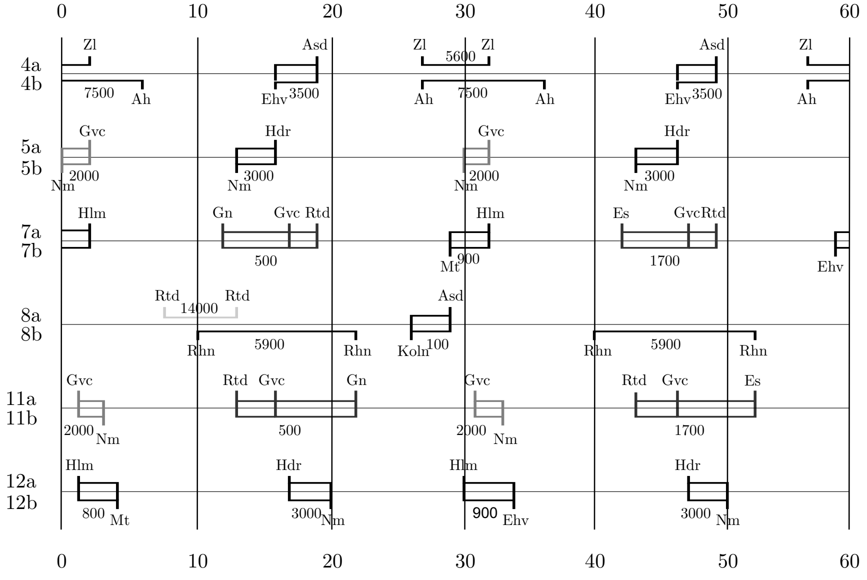


Figure 2.2: Platform occupation chart.

schedule of an already existing train is adjusted.

During the process of adding, deleting, and adjusting the schedules of trains, one usually runs into problems at a certain moment in time. It may not be possible to schedule a new train as desired, because there is no capacity on the track or in the station, or because capacity is only available when a connection can not be realized. In such a case, some of the already scheduled trains have to be rescheduled. This can be achieved by shifting an already scheduled train to some earlier or later point in time, by adjusting the planned trip time of a train, or by relocating a train to a different platform. Cycling through this process of scheduling trains, and backtracking on previously made choices in case of a ‘dead end’, one may eventually arrive at a complete timetable. The timetable construction process is quite complex, and it may take a team of planners several months to create a complete timetable. Below, we describe the main aspects that are taken into account in the construction of a timetable.

2.1.1 Adding Recovery Times to Trip Times

The time it takes a train to cover a certain distance while traveling at the maximum allowed velocity is called the *technically minimum trip time.* When constructing a timetable, a certain *recovery time* is added to the technically minimum trip time. Traditionally, about 7% recovery time is added to the technically minimum trip time. So, the train is scheduled at a lower than maximum velocity. Therefore, small delays can be compensated for by having the train travel at a higher than scheduled velocity.

2.1.2 The Block Safety System

Clearly, the railway safety system also has to be taken into account. Each track section is divided into so-called *blocks*, and each block may be used by at most one train at any time. At the start of a block, a signal light indicates the status of the block to approaching trains. A red light indicates that the block is occupied, so an approaching train may not enter the block. A yellow light indicates that the block is free, but that the next block is occupied. So, on a yellow light, an approaching train can enter the block, but has to start decelerating because the next signal light is red. A green light indicates that the block is free and can be entered. Figure 2.3 illustrates these three states. The yellow signal light for block 1 indicates that the first train can enter block 1, and that the signal light for the next block is red. This red light for block 2 indicates that block 2 is occupied by the second train. Block 3 is empty, and the block after that too, so the signal lights for both block 3 and the next block are green. In practice, more signal light states are actually used, such as flashing signals,

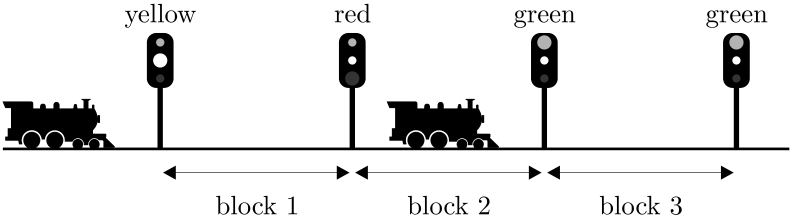


Figure 2.3: The railway safety block system

or signal lights in combination with a number indicating the maximum velocity at which the block can be entered.

Typically, the length of a block is between 1000 and 2000 meter. Therefore, taking every single block into account would require very precise and detailed time-space diagrams. Instead, planners usually choose a less detailed perspective, and model the blocks by so-called *minimum headway times* that separate any two trains using the same track. The idea is that it takes a train, on average, the headway time to cover a block. So, as long as the headway time is respected in the timetable, it should in practice be possible for all trains to be operated while respecting the safety system.

2.1.3 Connections between Trains

If a *passenger transfer connection* between two trains is desired, then both trains should be present at the connecting station at about the same time. Moreover, the trains are preferably assigned to opposite sides of the same platform, which creates a so-called *cross-platform connection.*

If two trains are to be combined to travel on as one single train, they need to be present at the station at *exactly* the same time. This yields a so-called *combination connection.*

2.1.4 Turn Around Times at Termini

One also has to pay attention to the so-called *turn around time* between arrival at a terminus, and the departure time of the train servicing the opposite route. If the turn around time is sufficiently long for the train to be cleaned, and for possible shunting operations to be carried out, then both journeys can be operated by the same train composition. Moreover, the turn around time should not be too long, because long turn around times decrease the efficiency of the train composition utilization. The actual planning of the train compositions takes place at a later point in time. These turn around times already indicate the interaction between timetable planning and

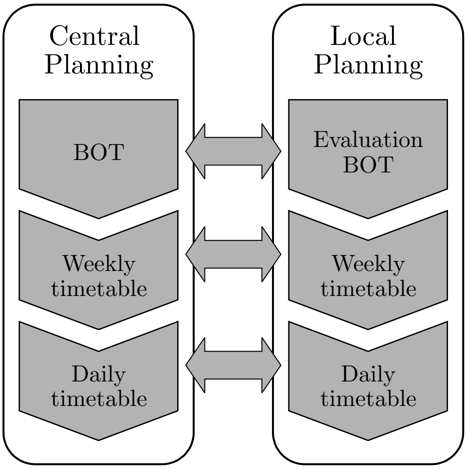


Figure 2.4: The timetable planning process at NSR

rolling stock planning.

2.2 The Timetable Planning Process at NSR

This section describes the timetable planning process at NSR. The description is mostly based on Prins (1998), who analyzes the planning processes at NSR from an information management point of view. Basically, the same processes take place at other railway operators as well, though some differences may be found.

In the Netherlands, timetables are constructed by the railway operators. Such an operator timetable is then proposed to Railned, which inspects whether the timetable respects the timetabling safety norms. Moreover, Railned checks whether sufficient infrastructure capacity is available for operating the timetable in combination with the proposed timetables of the other operators. If this is not the case, a feedback process to the railway operators is started.

The timetable planning process at NSR takes place at both a central level and a local level, and in several planning phases, as is shown in Figure 2.4. A rough initial timetable is first constructed centrally. This initial timetable is then sent to the local planning departments, which fill in the details so as to check its local feasibility, for example, by composing a routing of the trains through the stations. Moreover, the local planners can propose alterations to the central timetable to improve its feasibility and quality.

At both the central and local planning level, three types of timetables are constructed: the Basic One-hour Timetable (BOT), the Weekly Timetable, and the Daily Timetable. The BOT is a cyclic timetable for one hour, and contains the lowest level of detail. The Weekly Timetable is based on the BOT, and specifies the timetable for each day in a standard week. The Daily Timetable incorporates any deviations on a specific day from the prescribed schedule in the Weekly Timetable.

Basic One-hour Timetable. The timetable planning process starts with the construction of the BOT, which is a cyclic timetable for one 'standard hour' for the complete Dutch railway network. Usually, several BOTs are constructed, for example, for peak hours and off-peak hours. The BOT consists of a time-space diagram for all tracks, and of platform occupation charts for the stations. Since the BOT is the heart of the final railway timetable, the timetable is basically created in this phase.

Weekly Timetable. The Weekly Timetable extends the BOT from a standard hour timetable to a timetable for a standard week. The BOTs for peak and off-peak hours form the basis for constructing a timetable for every hour of all days of the week. The Weekly Timetable differs from the BOTs in that deviations from the standard hour are visible. These deviations are necessary because of differences between peak and off-peak hours, between early morning and late evening schedules, and between weekday and weekend schedules. The Weekly Timetable typically specifies a special timetable for Monday, because of post-weekend traffic, for Friday, because of pre-weekend traffic, and for Saturday and Sunday, because of weekend traffic. The Weekly Timetables for Tuesday, Wednesday, and Thursday are usually more or less identical. Each year, the Weekly Timetable is adjusted a number of times to incorporate changes in travel demand, and in the availability of infrastructure, crews, and rolling stock.

Daily Timetable. The Daily Timetable specifies the adjustments to the Weekly Timetable for a specific day of the year. These adjustments are necessary because of infrastructure maintenance, or because of short-term changes in the transportation demand due to holidays or special events. Depending on the impact of the adjustments, either some details are changed, or an adjusted BOT is created, which is then used as a basis for a new timetable.

Next, we describe the role of the central and local planning departments for each of these three timetables.

Central Planning. The initial versions of the BOT, Weekly Timetable, and Daily Timetable are constructed centrally. So, the core of the described timetables is made centrally for the whole of the Netherlands. These central timetables are represented by means of time-space diagrams and platform occupation charts. After an initial central BOT, Weekly Timetable, or Daily Timetable has been constructed, it is sent to the local planning departments.

Local Planning. The local planning departments plan the railway operations in the regions formed by the railway infrastructure around the major stations. The timetable planning task of the local planning departments is two-fold. First, the local planning departments evaluate the feasibility and quality of the BOT. Based on the feedback of the local planners, the BOT may be adjusted at the central level. For instance, local planners may have suggestions for adjustments that remove local infeasibilities, or improve the quality of the BOT in their region. The second task applies to the Weekly Timetable and the Daily Timetable only, and consists of further filling in the details of the central timetable. The central Weekly Timetable and Daily Timetable consist of time-space diagrams and platform occupation charts. So, it is known at what time each train departs from and arrives at a station, and what platforms the trains dwell at and for how long. Locally, each train has to be assigned a route through the station, from its entry point to the platform, and from the platform to its exit point. The in-station shunting has to be scheduled and routed, and crews have to be available for shunting operations and combining/splitting procedures. Also, rolling stock units may need inside or outside cleaning, and short term maintenance.

The planning process is an iterative process. First, the BOTs are constructed. Next, a Weekly Timetable is constructed based on the BOTs. Last, with the Weekly Timetable as input, the Daily Timetable is constructed. Besides that, the timetables are first constructed centrally, after which the local planning departments evaluate them, and/or fill in the details. The timetable planning process also contains feedback loops between the central and local planning, as is illustrated by the arrows in Figure 2.4. If the local evaluation of the BOT results in improvement suggestions, or in a negative advice because of operational infeasibility at a local level, then the central planning department may adjust the BOT. The same applies to the Weekly Timetable and the Daily Timetable.

For smaller railway operators, there may not be such a clear distinction between central and local plans. Indeed, this split follows the classical railway organizational structure, with central headquarters making the global decisions, and regional departments filling in the details of the central decisions (Wyckoff, 1976). A small railway operator may not have such an organizational structure, and therefore no

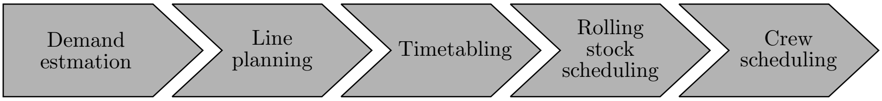


Figure 2.5: A railway operator's planning processes

distinction between central and local planning. Still, the separation between globally planning first, and working out the details of the global plan later, is likely to be present to a large extent.

Finally, we want to point out the dependencies between the timetabling process and other railway planning processes. For example, the timetable determines the environment in which a rolling stock plan and a crew plan are constructed. This means that, in constructing a timetable, one has to take into account the implications for the depending plans. These dependencies are further discussed in the next section.

2.3 Railway Planning Processes

Clearly, the timetable is not the only plan that needs to be composed in order to operate a railway system. There are several other issues that require planning, and those influence the timetable, or are influenced by the timetable. This section briefly describes the major railway planning problems, so as to clarify their relation with timetabling. Figure 2.5 shows the planning processes for a railway operator, and the time dependencies between them. The figure contains a mix of strategic planning problems, such as demand estimation and line planning, and more operational planning problems, such as rolling stock scheduling and crew scheduling. For the latter two, it is essential to also construct a strategic capacity plan, since acquiring new rolling stock and hiring new crews usually takes quite some time.

Demand Estimation. The estimation of the demand for railway services lies at the basis of a railway system. Travel demand is estimated as the number of people that wish to travel from an origin to a destination. By estimating the travel demand for each possible Origin-Destination combination, a so-called *OD-matrix* of total travel demand is obtained. Passenger counts, passenger interviews, and ticket sales form the basic information for constructing an OD-matrix. In the Netherlands, information on travel demand is also available through the LMS[1], an econometric model developed by the AVV (see the AVV website, 2002). The LMS estimates the travel demand between geographical zones in

[1] Landelijk Model Systeem (National Model System).

the Netherlands, and splits that demand, based on factors such as day of the week, time of the day, mode of transportation, and travel motivation.

Line Planning. After the OD-matrix has been estimated, one proceeds with deciding which train connections will be part of the railway system. The set of train connections which is operated is called the *train line system.* A train line is a direct train connection between an origin station and a destination station, via a certain route through the railway network. For each train line, a frequency is specified, and a type, which determines the stations that the line calls at. In the Netherlands, three line types are currently operated: the intercity type, which only calls at the major stations, the interregional type, which also calls at the smaller stations, and the local type, which stops at each station it passes. The line planning problem considers how to cover the railway network with lines, such that all traffic demand can be met, while meeting certain objectives. Common criteria are maximizing the number of direct travelers, and minimizing the costs of the railway system.

Timetabling. Once the line plan is complete, a timetable for its train lines can be constructed. For a detailed description of the timetable planning process we refer to the previous section.

Rolling Stock Planning. The next planning problem to be solved is the assignment of train units to the train lines in the timetable. A railway company typically owns a variety of rolling stock types, for example, single deck and double deck units, wagons that need a locomotive, units that have their own engine, etc. When each train line has been assigned one or more types of rolling stock, a plan is constructed that specifies how many units each train consists of. Moreover, a unit does not have to be assigned to the same train line for the entire day. A unit may be used for one train line first, then for another, etc. During off-peak hours, not all rolling stock is in use, and the idle train units need to be shunted from the platforms to shunting yards. A shunting plan is constructed in a later planning phase. Also, each train unit needs to be taken out of circulation after having traveled a certain distance, in order to be maintained. A final planning phase considers how to adjust the flow of train units such that each unit is routed to a workshop when it requires maintenance.

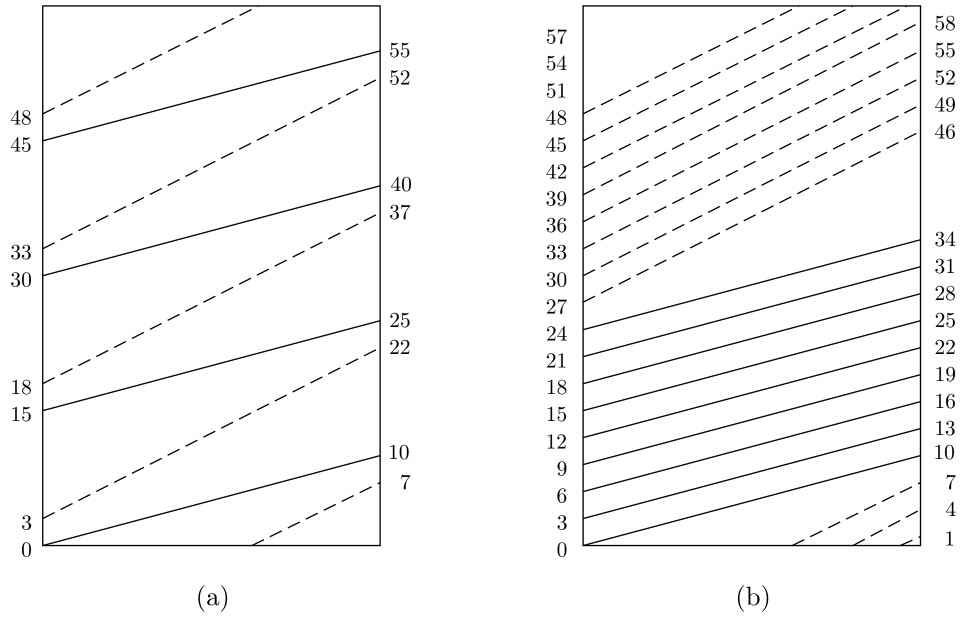
Crew Planning. Each train has to be manned by a driver and one or more conductors. This poses complex planning problems, since the crew plan needs to respect several complicated labor rules. The drivers and conductors have to return to their home bases by the end of the working day. Working shifts have to contain a meal break of at least half an hour, and may not contain a continuous period of work of more than five hours. More complicated rules also exist. A shift must contain some variation, so a driver or conductor may not be assigned to the same train line all day, going back-and-forth on that line. Drivers and conductors are allowed to transfer from one train to another only when sufficient buffer time for the transfer is available. Otherwise, a delayed arrival of the crew may result in a delayed departure of their next train. Furthermore, the crew schedule should incorporate various crew member characteristics, such as rostering qualifications, individual requests of crew members, and the past rosters of crew members. These past rosters are of importance for the labor rules. Finally, taking all these rules and characteristics into account, a railway operator aims at constructing a crew plan that meets certain objectives, such as minimizing the number of duties that cover all the work, or maximizing crew satisfaction.

The flow of the overall planning process as shown in Figure 2.5 arises because certain processes provide the input for others. However, the planning flow is not as linear as it may seem, since feedback loops between the several planning stages exist. For example, the timetable may be changed to improve the rolling stock circulation or the crew schedules.

These dependencies between the railway planning processes show the importance of timetabling for the planning chain as a whole. If fast methods exist to construct good timetables, then feedback can be given quicker. Also, any feedback or requested changes from a depending plan can be processed fast, which shortens the time span of the depending plan. It can be concluded that fast construction methods for high-quality timetables not only shorten the timetabling process lead time, but, because of the mutual dependencies with the other logistic railway planning processes, they also shorten the lead time of the whole railway planning process.

2.4 Railway Infrastructure Planning at Railned

When NS became an independent organization in 1995, the governmental organization Railned was established to manage the Dutch railway infrastructure. On one hand, Railned assigns infrastructure capacity to the railway operators, thereby acting as a referee when the operators have conflicting interests. On the other hand, Railned has the role of advisor for the Dutch government for future infrastructure investments. These infrastructure investment advices are mostly based on scenario studies, in which several options for infrastructure extensions are compared on such



(a)

(b)

Figure 2.6: Track capacity depending on train speed and train order

criteria as costs, spatial viability, capacity extension, effectiveness, etc. Indeed, extension of the railway capacity is one of the important factors in judging a scenario. Here, capacity extension should be interpreted as the number of extra train services which can be operated on the extended railway network.

Typically, the lead time of railway infrastructure projects is in the range of fifteen years. Therefore, one ideally wants to estimate the capacity extension of a scenario in some rough manner, without going through the details of the railway planning processes presented in the previous section. After all, it is uncertain what trains, train types, or timetables will be operated in fifteen years time. However, this desire for a rough capacity estimate conflicts with the high dependency of railway capacity on the details of the operated timetable.

More specifically, track capacity depends on the velocity of the trains, and on the order in which the trains enter the track. As an example, consider the time-space diagrams in Figure 2.6(a) and (b). Solid lines represent fast intercity trains, and dashed lines represent slower local trains. It takes an intercity train 10 minutes to travel along the track section, and a local train needs 19 minutes in total, because of some stops (that are not shown in the figure). We assume that upon entering and leaving the displayed track, there should be a safety distance of at least three minutes between subsequent trains. In Figure 2.6(a), the timetable alternates between intercity trains and local trains, as is customary in practice. This timetable offers a service of 8 trains per hour, 4 of each type, and it is clear from the figure that this is the maximum alternating service possible. In Figure 2.6(b), a service of 17 trains per hour can be offered. This 17 train service is the result of homogenizing the traffic order and velocity, by first operating 9 intercity trains, and after that 8 local trains. This example shows the strong dependency of track capacity on the timetable, and thus the dependency of network capacity on the timetable. Moreover, there is less freedom in using the available capacity than may appear at first sight. Because of commercial reasons, the alternating pattern in Figure 2.6(a) may be desired, or trains may have to be present at a station at the same time in order to provide a connection.

Because of the high dependency of railway capacity on the structure of the timetable, it is complicated to judge the effectiveness of infrastructure extensions without actually constructing a future timetable. So, in practice, Railned tries to construct several timetables for each future infrastructure scenario. Each of these timetables is in principle a BOT, though usually less detailed than the BOTs described in Section 2.2. If it is hard or impossible to create a BOT for a certain infrastructure scenario, then the scenario is not recommended to the Dutch government. If, on the other hand, various timetables can be constructed easily for a scenario, then one can argue that this scenario offers considerable flexibility, and it will be recommended for implementation. Such a recommendation for implementation also depends on other factors, such as the costs associated with the extension.

The above discussion shows the importance of timetable construction for Railned's infrastructure advisory process. The time span of the advisory process is negatively impacted if it takes a lot of time to construct only a single timetable. However, if decision support for the fast construction of timetables is available, then the time span of the advisory process is not only reduced, but the advice will also be better founded, since for each scenario a family of timetables can be generated.

2.5 The DONS Decision Support System

In 1992, NS started the development of a Decision Support System for supporting the planning process of constructing BOTs. Both the resulting DSS, and the project within which it was developed, are called DONS[2]. When NS became an independent organization in 1995, DONS and its further development continued as a joint project of passenger railway operator NSR and infrastructure capacity manager Railned. This section briefly describes the DONS system. For a more detailed description we

[2] Designer Of Network Schedules. Initially, DONS abbreviated the Dutch 'Dienstregeling Ontwikkeling bij de Nederlandse Spoorwegen'. This was later changed into the current meaning.

refer to Hooghiemstra (1996) and Hooghiemstra et al. (1999).

The goal of DONS is to provide long-term planners with a tool to generate BOTs in less time than it would take to construct them by hand, thereby enabling them to conduct more extensive scenario studies. The importance of being able to perform scenario studies was described in the previous sections, and in Section 1.2. The DONS system consists of three parts:

Graphical User Interface. The Graphical User Interface (GUI) allows the user to specify a timetable instance through mouse point-and-clicks, and by entering additional information through the keyboard. The exact ingredients of an instance are explained below. Moreover, once a timetable has been constructed, the GUI can display it in the form of time-space diagrams, platform assignment charts, and tabular material.

Database. The DONS database contains all the information of the various problem specifications that planners have entered and saved. It also contains detailed information on relevant Dutch railway infrastructure scenarios, such as the current infrastructure, and likely-to-be-realized future infrastructure layouts. Moreover, for each instance, it stores the timetables that have been computed.

Intelligent Modules. When an instance has been specified, by retrieving information from the database and/or using the GUI, the intelligent computing modules can be used to generate a BOT. The CADANS module (Schrijver and Steenbeek, 1994) calculates the departure and arrival times for all specified trains. These departure and arrival times form the information that is needed for a time-space diagram, so CADANS does not consider the events that take place inside the stations. The STATIONS module searches, for a single station, for a platform assignment and a train routing that match the timetable calculated by CADANS. For more information on STATIONS, we refer to Zwaneveld et al. (1996), Zwaneveld (1997), and Kroon et al. (1997). When STATIONS has solved this problem for every station in the problem setting, a complete BOT has been constructed.

The railway network infrastructure lies at the basis of a DONS instance. Using the GUI, one can specify stations as nodes in the network, and link two stations to indicate that there is a rail track between them. One must further specify several characteristics of the track, such as its length, the number of parallel tracks, and the voltage of the catenary. If only a time-space diagram is needed, these infrastructure data suffice. If one also wants to construct platform occupation charts, then detailed infrastructure information needs to be entered for each station. In practice, however, one mostly loads pre-defined infrastructure layouts from the database.

Next, the train lines are entered into the system. For each train, one specifies characteristics such as origin, destination, route, and train type. Dwell stations, and dwell times or dwell time windows, can also be entered, or the default values for the corresponding train type can be used.

Finally, logistic and commercial requirements for the timetable are specified. Logistic requirements include such matters as limiting the turn around times at termini, or preventing two trains from being at a station at the same time, when they have crossing routes within the station (recall that for the time-space diagram planning part the internal station layout is unknown). Commercial requirements involve connections between trains, bounds on dwell times to limit total trip times, and synchronizing the departure times of trains that share part of their route, so as to offer a higher frequency train service on the common section. The next chapter describes these issues in detail.

Chapter 3

Modeling Cyclic Railway Timetables

This chapter describes a model for cyclic railway timetables. Section 3.1 describes the basic model assumptions. We distinguish between assumptions with respect to the level of detail of the railway infrastructure, the level of detail of trains, and the considered types of timetable requirements. Next, Section 3.2 describes the objectives that are taken into account. Section 3.3 introduces periodic constraints as a basic tool for modeling cyclic timetable requirements. We demonstrate the modeling power of periodic constraints in Section 3.4 by applying them to model various representative examples of practical timetable requirements. We formalize these example constraints in Section 3.5, and present the full model. Section 3.6 shows how to rewrite the model in terms of a constraint graph, which results in a clearer and more compact model formulation. In Section 3.7, we present various specific instances of the general objective function of the model. Finally, Section 3.8 discusses the structure of cyclic railway timetabling constraint graphs.

3.1 Model Assumptions

In studying and modeling cyclic railway timetables, we assume the following to be given a priori:

- the infrastructure layout of the railway network,
- the trains to be scheduled, and
- a set of timetable requirements, representing safety regulations, service levels, and logistic limitations.

The following subsections elaborate on each of these three assumptions.

3.1.1 Infrastructure Assumptions

The railway infrastructure is considered as a network of nodes and tracks. The nodes represent locations in the railway network where train movements may interact. Nodes therefore require a coordination of train movements. A clear example of a node is a station, where trains from several directions arrive and dwell for some time, where passengers transfer between trains, where trains may turn before starting their return trip, etc. Other examples of network nodes are crossings, junctions, bridges, and shunting yards.

Tracks, then, connect the nodes and are used by trains to travel from one node to the next. When multiple parallel tracks exist between a pair of nodes, each train is assumed to be assigned a priori to one of the available tracks. Although this approach might overlook some technically possible timetables, it is easily validated by inspecting real-world situations. When two tracks are available, each track is typically used for one direction of travel. In the case of four parallel tracks, two tracks are used for each direction of travel. For one specific direction, one of the two tracks is assigned to the faster intercity and interregional trains, and the other track to the slower local trains and cargo trains. The reason for dedicating the track usage to equal velocity trains is that it results in a higher train frequency, and thus in a higher track utilization (see Section 2.4).

Since nodes are the locations where train movements are coordinated, train movements are by definition not allowed to interfere on a track. This means that it is not allowed for trains to meet or overtake one another on a track. In practice, meets and overtakes only take place at stations, junction yards, or on side tracks. In the latter case, the side track is a separate track, which is connected to the main track by two junction nodes at both ends. A faster train can overtake a slower one when the slower train makes way by temporarily using the side track. So, at the moment of meeting or overtaking, the two trains are not actually using the *same* track. The above described a priori assignment of trains to tracks also applies to side tracks. So, it is decided beforehand which trains should move to a side track to be overtaken.

By viewing stations as nodes in the network, most of the in-station infrastructure details are lost. But taking into account the complete in-station infrastructure layout results in large and complicated models. So, stations are assumed to be black boxes, and a timetable resulting from the model may be infeasible with respect to the detailed in-station infrastructure layout. For example, a station may have too few platforms to host the trains as specified by the timetable. Or, only conflicting routes exist to and from the platforms through the in-station infrastructure. The latter situation occurs when the timetable prescribes two trains to use crossing inbound routes at the same time. The problem of routing trains through stations is assumed to be solved in a later phase of the timetable planning process. Sections 2.2 and 2.5, and Zwaneveld et al. (1996), Zwaneveld (1997), and Kroon et al. (1997) describe the in-station routing of the trains in more detail. We relax this black box station assumption to a certain extent in Chapter 6. There, we present an extension that allows for incorporating station capacity into the model, which results in a grey box model of the stations.

3.1.2 Train assumptions

We consider trains in the form of so-called *train lines*. A train line is a direct train connection between an origin and a destination station along a given route. Each train line has a frequency, which specifies how many trains of that line are operated each cycle time. For example, with a cycle time of one hour, a frequency of two means that every half an hour a train of the line departs from each served station along the line. Associated with each train line is a type, which determines at which stations the trains of the line call. The most common current train types in the Netherlands are intercity trains, interregional trains, local trains, and cargo trains. For each train, the trip time between subsequent nodes is assumed to be fixed and known a priori. In Chapter 6 we extend the model by incorporating trip time windows.

3.1.3 Timetable Requirements

A timetable should meet various requirements, such as safety regulations, service levels to be achieved, and restrictions considering the operational feasibility of the timetable. Each requirement belongs to one of the categories described below. These categories have proven to cover most practical needs and are therefore the only requirements that are taken into account.

Dwelling at stations. In practice, limits are specified for the time that a train dwells at a station. On one hand, a minimum dwell time specifies the minimum time needed for passengers to alight and board. On the other hand, one might also want to limit the dwell time at a station because stations only have a limited capacity, and because each dwell minute adds to the train's total travel time.

Connecting trains. Two trains are said to be connected if there is a planned relation between their arrival and departure times at a certain node. A connection is desirable in two situations. The first situation occurs when a direct train connection between two nodes does not exist. In that case, passengers should still be offered a good travel scheme between these two nodes. This can be achieved by constructing a timetable in which a train for the destination node leaves from a transfer node shortly after the train from the origin node has arrived.

Another situation requiring the connection of two trains arises when two trains need to be combined. This situation occurs when the two trains share a considerable part of their routes. Combining the two trains into one for their common route saves both human resources and infrastructure resources. Only one train crew is required for the combined train, and infrastructure capacity is saved because otherwise a headway time between the two trains is to be respected. Such a combining of trains obviously requires both trains to be concurrently present at the station. The combining of trains also has disadvantages. It places an extra restriction on the timetable, and can be a source of delays since the combination procedure involves connecting the trains, a break test, and other checks, during which failures may occur.

Synchronizing trains. When trains from different train lines share part of their routes, their departure times are often synchronized to offer a higher frequency service on that common part. As an example, if two train lines have frequency equal to one, then synchronizing their departures provides a service with frequency two on the common part of their route. When a single train line has frequency higher than one, synchronization can also be applied to spread the multiple trains of that line evenly across the cycle time. Since this synchronization of train lines, or of trains within a train line, leads to a train service with a higher frequency, the involved constraints are also known as frequency constraints.

Turning around at termini. A *rolling stock composition* is the set of carriages, either self-powered, or including a locomotive, which operate a certain train line. After a rolling stock composition has arrived at its terminus, it generally turns around after some time, and operates a different connection. Typically, the rolling stock composition operates the return journey of the train line it operated before, but one can also encounter other situations. Before leaving for the opposite trip, the rolling stock is scheduled to spend some time at the terminus in order to be cleaned, for shunting, as buffer time to absorb delays, etc.

Fixed arrival and departure times. For some trains, the freedom of selecting arrival and departure times is limited. This especially applies to international trains. The time at which an international train appears at the border of a country is usually the result of negotiations with the neighboring railway companies, rather than a variable to be decided upon in the timetable planning process.

Once the train has entered the country, the timetable can usually be chosen as desired for the further time that the train spends inside the country, as long as the fixed trip times of the train are taken into account.

Safety regulations. A final, but clearly very important requirement is due to railway safety regulations. These require any two trains using the same track to be separated by a certain minimum headway time (see Section 2.1.2). For any pair of trains using the same track, the minimum headway time must be respected both at the origin node and at the destination node of the track. Moreover, the safety regulations forbid the meeting and overtaking of trains on the same track. The latter follows directly from the definition of nodes and tracks at the start of this section.

3.2 Objectives in Constructing Timetables

Several objectives come to mind when constructing a timetable. These can basically be divided into three groups:

- satisfying customers,
- creating a stable and robust railway system, and
- controlling costs.

These objectives may be conflicting. As an example, passengers would be immensely satisfied if each of them was offered a direct train connection, without any intermediate stops, and at exactly the preferred time of travel. But such a system would clearly result in colossal costs, if it were operationally feasible at all.

3.2.1 Travel Time

An important factor for customer satisfaction is the total journey time. However, there is only limited freedom in providing passengers fast travel opportunities with few transfers, since the train lines and their trip times are assumed to be given and fixed. Still, through dwell times and connection times, the timetable has some influence on the total journey time.

In fact, the objectives that aim at offering customers fast travel times correspond to the dwell and connection requirements mentioned in the previous section. Rather than just satisfying these requirements, the objective is to satisfy them as well as possible. So, rather than making sure that there is sufficient dwell time for passengers to alight and board, and providing just some connections between trains, these requirements should be incorporated into the timetable as well as possible.

3.2.2 Robustness

A second important factor for customer satisfaction is the robustness of a timetable. In a timetable that just meets the safety requirements, trains may follow one another at exactly the minimum headway time. A small delay of one train may then be easily knocked on to other trains, and so propagated through the entire network. Therefore, a second timetabling objective is to construct a robust timetable that contains some buffer time above the minimum headway time to absorb small disturbances. While planning a timetable, a certain buffer time is generally already added to the minimum headway time. Still, when track capacity is available, one can argue that timetable robustness is increased by 'pulling apart' the trains as far as possible, since a delayed train is then less likely to interfere with the other trains on the track.

3.2.3 Costs

An obvious third objective is to minimize the costs associated with the timetable. The major cost components of a railway system are formed by the infrastructure, the rolling stock, and the train crews. Given the assumptions in the previous section, there are only few choices left that influence these operating costs of a railway system. Infrastructure and train lines are assumed to be fixed and given, and the rolling stock scheduling and crew scheduling occur in a later planning phase. However, within this limited freedom, one can still pursue the objective of constructing a timetable that requires a minimum number of rolling stock compositions.

3.2.4 Infeasible Problem Instances

An initial set of requirements may be too tight, in which case it is impossible to construct a timetable that meets all requirements. Then, a clear objective is to find an acceptable timetable which minimally violates the initial requirements. Certain requirements are not allowed to be violated in such a procedure, such as safety requirements. Others could however be violated, such as commercial requirements.

	arrival	departure
The Hague CS		07:21
The Hague HS	07:24	07:25
Rotterdam	07:43	07:45
Dordrecht	08:00	08:01
Breda	08:18	08:19
Tilburg	08:32	08:33
Eindhoven	08:58	09:01
Helmond	09:10	09:11
Venlo	09:32	

Table 3.1: Detailed timetable for the 1900 intercity train of 07:21.

3.3 Periodic Constraints

This section describes how the cyclic behavior of the train arrivals and departures in a cyclic timetable can be modeled. Since a timetable consists of arrival and departure times for all trains at each node they pass, it is natural to use the following decision variables to model these arrival and departure times:

$a_n^t \in \{0, \ldots, T-1\}$ for the arrival time of train t at node n, and
$d_n^t \in \{0, \ldots, T-1\}$ for the departure time of train t from node n.

The parameter T is the cycle time of the timetable, in time units. For the Dutch case, with a cycle time of one hour, T equals 60 minutes. Because the cyclic timetable is exactly the same for each cycle period, it suffices to consider the decision variables on the domain $\{0, \ldots, T-1\}$. The decision variables a_n^t and d_n^t are restricted to take integer values because, in practice, timetables are calculated and published in integer minutes.

The following example illustrates the use of the decision variables a_n^t and d_n^t. The example is based on Table 3.1, which contains a detailed timetable for the 1900 train that departs from The Hague Central at 07:21. The table shows both the departure and the arrival times of the train. Our example considers the modeling of dwell times. The 1900 intercity train dwells at Tilburg for one minute. This is modeled as

$$
\begin{aligned}
& d^{1900}_{\text{Tilburg}} - a^{1900}_{\text{Tilburg}} = 1, \\
& \text{with in the current timetable } \begin{cases} a^{1900}_{\text{Tilburg}} & = 32, \\ d^{1900}_{\text{Tilburg}} & = 33. \end{cases}
\end{aligned}
$$

At Eindhoven, the train dwells for three minutes. The train arrives at Eindhoven at :58, and departs at :01, so it 'passes' the origin of the clock while dwelling at the station. This is modeled as follows:

$$
\begin{aligned}
& d^{1900}_{\text{Eindhoven}} - a^{1900}_{\text{Eindhoven}} = -57 = 3 - 60, \\
& \text{with in the current timetable } \begin{cases} a^{1900}_{\text{Eindhoven}} & = 58, \\ d^{1900}_{\text{Eindhoven}} & = 01. \end{cases}
\end{aligned}
$$

In general, we need to calculate modulo T to correctly express relations between the decision variables a^t_n and d^t_n. For the constraints above, this gives

$$
\begin{aligned}
d^{1900}_{\text{Tilburg}} - a^{1900}_{\text{Tilburg}} &= 1 \text{ modulo } 60, \\
d^{1900}_{\text{Eindhoven}} - a^{1900}_{\text{Eindhoven}} &= 3 \text{ modulo } 60.
\end{aligned}
$$

However, a timetabling model based on modulo T calculations is hard to solve with existing operations research techniques. The modulo T constraints are therefore reformulated using integer variables. More precisely, the modulo T operation is replaced by an integer decision variable, which is used to add or subtract an integer multiple of the cycle time T whenever needed. For the two example constraints, let these integer variables be p and p'. Replacing the modulo 60 by $60p$ and $60p'$ gives

$$
\begin{aligned}
& a^{1900}_{\text{Tilburg}} - a^{1900}_{\text{Tilburg}} + 60p = 1, \\
& d^{1900}_{\text{Eindhoven}} - a^{1900}_{\text{Eindhoven}} + 60p' = 3. \\
& p, p' \in \mathbb{Z}.
\end{aligned}
$$

The construction with the integer variables p and p' exactly models the modulo T operation.

The cyclic railway timetabling model used in this thesis is mainly based on this type of so-called *periodic constraints*, in which the modulo T operation is dealt with through integer variables. To generalize this idea, consider a^t_n, the arrival time of some train t at node n, and $d^{t'}_m$, the departure time of some other train t' from node

m. A general periodic constraint in the model relates these two variables a_n^t and $d_m^{t'}$ through a time window $[l, u]$, rather than just a fixed value, and has the form

$$a_n^t - d_m^{t'} + Tp \in [l, u], \qquad p \in \mathbb{Z}. \tag{3.1}$$

As a convenient shortcut for the general periodic constraint (3.1), the notation $[l, u]_T$ is used to indicate that the time window is actually a periodic time window with cycle time T. So the constraint

$$a_n^t - d_m^s \in [l, u]_T. \tag{3.2}$$

is a shortcut for (3.1).

The dwell times of a train represent the time spans of the dwelling processes, and can therefore be seen as *process times*. One can also model cyclic railway timetables using decision variables that model these process times, rather than arrival and departure times of trains. Chapter 5 presents such a model based on processing times.

3.4 Modeling Timetable Requirements

Recall from Section 3.1.3 that the following timetable requirements are taken into account:

- dwelling at stations,
- connecting trains,
- synchronizing trains,
- turning around at termini,
- fixed arrival and departure times, and
- safety regulations.

By presenting extensive examples, this section shows that the above requirements can be modeled by means of periodic constraints (3.1), (3.2). The modeling of the synchronization of trains and of the safety requirements is quite complicated. For these two requirements, we therefore present some examples, as well as a mathematical generalization of the examples.

Our examples are again based on the 1900 intercity train running from The Hague to Venlo, and on the trains it interacts with on its journey from The Hague to Venlo. These trains are:

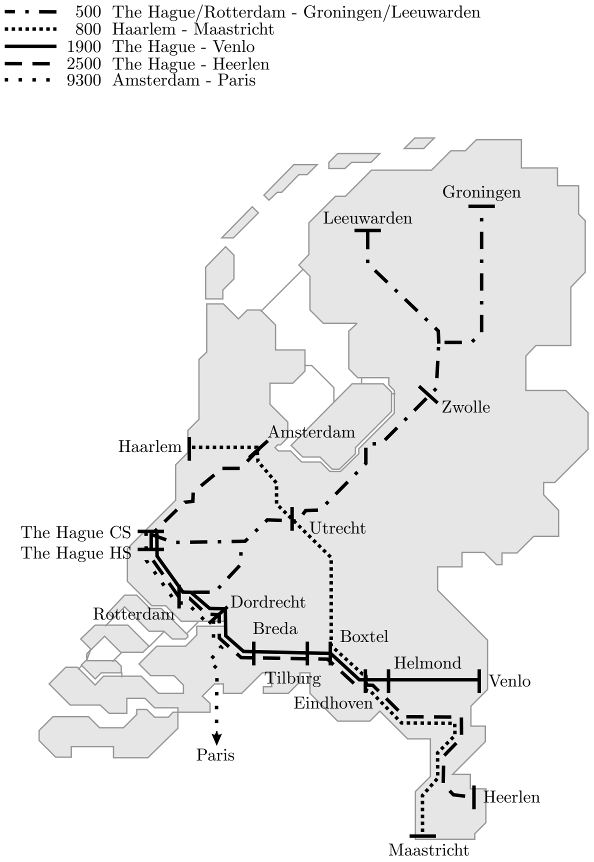


Figure 3.1: The routes of the 500, 800, 1900, 2500, and 9300 trains.

500 This so-called ‘North-East’ intercity train travels from Rotterdam and The Hague, via Utrecht and Zwolle, to Groningen and Leeuwarden.

800 The intercity train from Haarlem to Maastricht.

2500 The intercity train from The Hague to Heerlen.

9300 The high-speed Thalys train from Amsterdam to Paris.

The routes of these trains are sketched in Figure 3.1. Further, the examples also consider the opposite trains of the 1900 en 2500 trains, which are denoted by 1901 and 2501. For all examples, we set the timetable cycle time T equal to 60 minutes.

3.4.1 Trip Time and Dwell Time Constraints

We start with a set of constraints that were not mentioned above as a requirement, namely trip time constraints. Trip time constraints are closely related to dwell time constraints, since together they model the behavior of a train on its journey from origin to destination station. The trip times for the 1900 train can be derived from Table 3.1. For planning purposes, the dwell time window for intercity trains is usually set to $[1,5]_{60}$. The set of constraints in Figure 3.2 then models the complete journey of the 1900 train.

$$
\begin{array}{lclcl}
a^{1900}_{\text{The Hague HS}} & - & d^{1900}_{\text{The Hague Central}} & = & [3]_{60}, \\
d^{1900}_{\text{The Hague HS}} & - & a^{1900}_{\text{The Hague HS}} & \in & [1,5]_{60}, \\
a^{1900}_{\text{Rotterdam}} & - & d^{1900}_{\text{The Hague HS}} & = & [18]_{60}, \\
d^{1900}_{\text{Rotterdam}} & - & a^{1900}_{\text{Rotterdam}} & \in & [1,5]_{60}, \\
a^{1900}_{\text{Dordrecht}} & - & d^{1900}_{\text{Rotterdam}} & = & [15]_{60}, \\
d^{1900}_{\text{Dordrecht}} & - & a^{1900}_{\text{Dordrecht}} & \in & [1,5]_{60}, \\
a^{1900}_{\text{Breda}} & - & d^{1900}_{\text{Dordrecht}} & = & [17]_{60}, \\
d^{1900}_{\text{Breda}} & - & a^{1900}_{\text{Breda}} & \in & [1,5]_{60}, \\
a^{1900}_{\text{Tilburg}} & - & d^{1900}_{\text{Breda}} & = & [13]_{60}, \\
d^{1900}_{\text{Tilburg}} & - & a^{1900}_{\text{Tilburg}} & \in & [1,5]_{60}, \\
a^{1900}_{\text{Eindhoven}} & - & d^{1900}_{\text{Tilburg}} & = & [25]_{60}, \\
d^{1900}_{\text{Eindhoven}} & - & a^{1900}_{\text{Eindhoven}} & \in & [1,5]_{60}, \\
a^{1900}_{\text{Helmond}} & - & d^{1900}_{\text{Eindhoven}} & = & [9]_{60}, \\
d^{1900}_{\text{Helmond}} & - & a^{1900}_{\text{Helmond}} & \in & [1,5]_{60}, \\
a^{1900}_{\text{Venlo}} & - & d^{1900}_{\text{Helmond}} & = & [21]_{60}.
\end{array}
$$

Figure 3.2: Trip time and dwell time constraints for the 1900 train.

3.4.2 Turn Around Constraints

After the 1900 train has arrived in Venlo, its rolling stock composition stays there for some time, and is then used to operate the opposite 1901 service. The 1900 train is said to turn on the 1901 train in Venlo. After arriving in The Hague Central with the 1901, and having stayed there for some time, the rolling stock composition is next used to operate the 2500 train to Heerlen. So, in The Hague Central, the 1901 train turns on the 2500 train. After the arrival of the 2500 train in Heerlen, the rolling stock composition travels back to The Hague with the 2501 train, and is then used for the 1900 train again.

Suppose that the turn around time window for Venlo and Heerlen is set to $[20, 50]_{60}$, and that the turn around time window for The Hague CS is set to $[15, 40]_{60}$. So, a minimum time of 20 minutes is required before the rolling stock can leave again. The 50 minutes upper bound indicates that rolling stock compositions should not stand still too long at a terminus. This turn around time window yields the four turn around constraints in Figure 3.3, two for The Hague, one for Venlo, and one for Heerlen.

$$\boxed{\begin{array}{lclcl}
d^{1901}_{\text{Venlo}} & - & a^{1900}_{\text{Venlo}} & \in & [20, 50]_{60}, \\
d^{2500}_{\text{The Hague CS}} & - & a^{1901}_{\text{The Hague CS}} & \in & [15, 40]_{60}, \\
d^{2501}_{\text{Heerlen}} & - & a^{2500}_{\text{Heerlen}} & \in & [20, 50]_{60}, \\
d^{1900}_{\text{The Hague CS}} & - & a^{2501}_{\text{The Hague CS}} & \in & [15, 40]_{60}.
\end{array}}$$

Figure 3.3: Turn around constraints for the 1900, 2500, and 2501 trains at The Hague CS, Venlo, and Heerlen.

3.4.3 Connection Constraints

We consider both transfer connections and combination connections. Let us first consider passenger transfer connections. On its journey from The Hague to Venlo, the 1900 train should connect to the 800 train in Eindhoven. A pair of connections should be created between the 1900 train and the 800 train. The first connection allows passengers from the 1900 train, from the direction of The Hague, to transfer to the 800 train in the direction of Maastricht. The other connection enables the passengers arriving with the 800 train from the direction of Haarlem to transfer to the 1900 train in the direction of Venlo.

In practice, a connection time window of $[2, 5]_{60}$ is used for connections between intercity trains. Here the assumption is that, in the later station scheduling phase, the trains are assigned to neighboring platforms. Passengers can then transfer between the trains quite quickly, in two minutes time. The five minute upper bound is specified to limit the extra travel time of the transferring passengers. Since, in principle, the transferring passengers can transfer in two minutes, any connection time that exceeds these two minutes adds to their minimum possible travel time. Note that the infrastructure capacity must allow the two involved trains to be both present in the station in the $[2, 5]_{60}$ time window. Figure 3.4 contains the constraints that model the connections between the 1900 and 800 trains at Eindhoven.

$$\boxed{\begin{array}{rcll} d^{1900}_{\text{Eindhoven}} & - & a^{800}_{\text{Eindhoven}} \in & [2, 5]_{60}, \\ d^{800}_{\text{Eindhoven}} & - & a^{1900}_{\text{Eindhoven}} \in & [2, 5]_{60}. \end{array}}$$

Figure 3.4: Passenger connection constraints between the 800 and 1900 trains in Eindhoven.

The 1900 train is not combined with any other train, so it does not provide an example of a combination connection constraint. The clearest example of such a connection in the Dutch railway system is the 500 train. The route of the 500 train is shown in Figure 3.1, and is described as follows. Two trains depart for Utrecht, one from Rotterdam, and one from The Hague. Upon arrival in Utrecht, the trains are combined into one train, which travels further to Zwolle. Upon arrival in Zwolle, this combined train is split again into two trains. One of these trains travels to Groningen, the other to Leeuwarden. The 500 train can thus be seen as consisting of the following three trains between:

500 The Hague and Groningen,

501 Rotterdam and Utrecht,

502 Zwolle and Leeuwarden.

The combination in Utrecht then requires a connection relation between the arrival of the 501 train and the departure of the 500 train. Analogously, the splitting of the train into two parts in Zwolle can be established by a connection relating the arrival time of the 500 train to the departure time of the 502 train.

A common practical value for the connection time window for combining or splitting trains is $[5, 10]_{60}$. At least five minutes are needed in order to connect the trains, or to release them. A maximum of ten minutes is specified, to bound the time that is added to the total time of the travelers who do not alight in Utrecht. The system of constraints in Figure 3.5 models the connecting and splitting of the 500 train.

Finally, note that the turn around constraints from the previous section can also be seen as a connection relation between the arriving and the departing train. Chapter 6 presents an extension that allows for more flexibility in modeling connections.

$$\boxed{\begin{array}{rcll} d^{500}_{\text{Utrecht}} & - & a^{501}_{\text{Utrecht}} \in & [5,10]_{60}, \\ d^{502}_{\text{Zwolle}} & - & a^{500}_{\text{Zwolle}} \in & [5,10]_{60}. \end{array}}$$

Figure 3.5: Combining and splitting connection constraints for the 500 train in Utrecht and Zwolle.

3.4.4 Fixed Arrival and Departure Constraints

As an example of fixed departure and arrival times, consider the 9300 Thalys high-speed train between Amsterdam and Paris. In the 2003 timetable, this train enters the Netherlands at :25, and leaves the country on its return trip at :34. Entering the Netherlands corresponds to the departure of the 9300 Thalys from the border-node in the Dutch railway network, and leaving the Netherlands corresponds to arriving at the border-node. Therefore, the first two constraints in Figure 3.6 model these fixed arrival and departure times. If the border crossing times are the result of agreements stating that the 9300 Thalys is to enter the Netherlands between :23 and :27, and leave between :32 and :36, then the last two constraints in Figure 3.6 can be used. Note that the constraints in Figure 3.6 are not periodic. So the variables a^{9300}_{border} and d^{9300}_{border} should simply equal the values 25 and 34, respectively, or lie between the values 23 and 27, respectively 32 and 36.

$$\boxed{\begin{array}{l} d^{9300}_{\text{border}} \quad = 25, \\ a^{9300}_{\text{border}} \quad = 34, \\ d^{9300}_{\text{border}} \in [23,27], \\ a^{9300}_{\text{border}} \in [32,36]. \end{array}}$$

Figure 3.6: Fixed arrival and departure constraints for the 9300 Thalys at the Dutch border.

It is possible to write the constraints in Figure 3.6 in the general form (3.1), (3.2). To that end, we need an auxiliary decision variable $\beta \in \{0,\dots,T-1\}$. Using β, the constraints in Figure 3.6 are rewritten to

$$d^{9300}_{\text{border}} - \beta = [31]_{60}, \tag{3.3}$$

$$a^{9300}_{\text{border}} - \beta = [32]_{60}, \tag{3.4}$$

$$d^{9300}_{\text{border}} - \beta \in [29, 33]_{60}, \tag{3.5}$$
$$a^{9300}_{\text{border}} - \beta \in [30, 34]_{60}. \tag{3.6}$$

If the auxiliary variable β takes the value zero, then these constraints are exactly the same as the constraints in Figure 3.6. We claim that one can always set one of the variables in the model to some preferred value. Intuitively, this can be seen as follows. All requirements are stated as periodic time window constraints involving two variables. Therefore, if some timetable satisfies the constraints, then adding a certain number of minutes to each arrival and departure time results in a new timetable which also satisfies all constraints. The two timetables are basically the same, except for the shift in minutes. If this shift is chosen such that the auxiliary variable β takes the value zero, then all fixed arrival and departure time requirements are satisfied. Section 4.3.3 proves this property of cyclic timetables mathematically.

3.4.5 Synchronization Constraints

The 1900 train from our examples shares the major part of its route, from The Hague to Eindhoven, with the 2500 train, which runs from The Hague, via Eindhoven, to Heerlen. Since both trains have frequency one, synchronizing them yields a service with frequency two along the shared connection The Hague–Eindhoven. However, we do not want to fix the timetable too much beforehand. Therefore the model requires the departure times to be 30 minutes apart, with a bandwidth of two minutes. Note that the 30 minutes that the trains should lie apart is obtained by dividing the cycle time T of 60 minutes by the frequency two. The synchronization takes place at each station the trains call at. Figure 3.7 shows the synchronization constraints between the two trains. Because synchronizing trains aims at offering a higher frequency service, the constraints in Figure 3.7 are called frequency constraints. Next, we generalize the frequency constraints, and allow for the synchronization of the departures of more than two trains.

3.4.6 General Synchronization Constraints

Suppose that the frequency of k trains $t_1, \ldots, t_k$ is to be synchronized, so that the departures of the trains are spread evenly across the cycle time T. In an ideal situation, a train departs every $\frac{T}{k}$ minutes. To offer some flexibility in the departure times, a bandwidth δ is defined, by which the departure time of a train may deviate from its perfect departure time.

First, let us consider the frequency relation between the departures of train t_1,

$$\begin{array}{lcll}
d^{1900}_{\text{The Hague CS}} & - & d^{2500}_{\text{The Hague CS}} \in & [28,32]_{60}, \\
d^{1900}_{\text{The Hague HS}} & - & d^{2500}_{\text{The Hague HS}} \in & [28,32]_{60}, \\
d^{1900}_{\text{Rotterdam}} & - & d^{2500}_{\text{Rotterdam}} \in & [28,32]_{60}, \\
d^{1900}_{\text{Dordrecht}} & - & d^{2500}_{\text{Dordrecht}} \in & [28,32]_{60}, \\
d^{1900}_{\text{Breda}} & - & d^{2500}_{\text{Breda}} \in & [28,32]_{60}, \\
d^{1900}_{\text{Tilburg}} & - & d^{2500}_{\text{Tilburg}} \in & [28,32]_{60}.
\end{array}$$

Figure 3.7: Synchronization constraints for the 1900 and 2500 trains between The Hague and Tilburg.

and the other trains $t_2, \ldots, t_k$. This relation is defined by the following constraints:

$$d_j - d_1 \in [\underline{s}_{1j}, \overline{s}_{1j}]_T \text{ for all } j = 2, \ldots, k, \tag{3.7}$$

where the synchronization time windows $[\underline{s}_{1j}, \overline{s}_{1j}]_T$ are defined as

$$[\underline{s}_{1j}, \overline{s}_{1j}]_T = \left[\frac{j-1}{k}T - \delta, \frac{j-1}{k}T + \delta\right]_T .$$

The constraints (3.7) require the departure of train t_j to take place $(j-1)\frac{T}{k}$ minutes after the departure of train t_1, give or take δ minutes. This indeed results in one train leaving every $\frac{T}{k}$ minutes after t_1. As an example, for four trains t, and a bandwidth of one minute, (3.7) yields the following synchronization constraints:

$$\begin{aligned}
d_2 - d_1 &\in [14, 16]_{60}, \\
d_3 - d_1 &\in [29, 31]_{60}, \\
d_4 - d_1 &\in [44, 46]_{60}.
\end{aligned}$$

However, in a feasible solution to the constraints (3.7), the departures of two trains may lie apart more than an integer multiple of $\frac{T}{k} \pm \delta$ minutes. Consider two trains t_j and t_{j+1}, and suppose that $d_1 = 0$. Then, in a feasible solution to (3.7), we may have $d_j = \frac{j-1}{k}T - \delta, d_{j+1} = \frac{j}{k}T + \delta$. Thus, $d_{j+1} - d_j = \frac{T}{k} + 2\delta$. For the example above, a feasible solution would be $d_1 = 0, d_2 = 14, d_3 = 31, d_4 = 46$, which gives $d_3 - d_2 = 17$.

Therefore, the departure times of all other pairs of trains also need to be synchronized. The set of constraints below synchronizes the departures of all trains

$t_1, \ldots, t_k$:

$$d_j - d_i \in [\underline{s}_{ij}, \overline{s}_{ij}]_T \text{ for all } i < j, \quad i, j = 1, \ldots, k, \tag{3.8}$$

where the synchronization time windows $[\underline{s}_{ij}, \overline{s}_{ij}]_T$ are defined as

$$[\underline{s}_{ij}, \overline{s}_{ij}]_T = \left[\frac{j-i}{k}T - \delta, \frac{j-i}{k}T + \delta\right]_T. \tag{3.9}$$

We define the constraints for $i < j$ in order to prevent stating double constraints. Because of the symmetry of the time windows $[\underline{s}_{ij}, \overline{s}_{ij}]_T$, we have that

$$\underline{s}_{ij} + \overline{s}_{ij} = T. \tag{3.10}$$

This symmetry, together with equation (3.10), implies that we may also write

$$d_j - d_i \in [\underline{s}_{ij}, \overline{s}_{ij}]_T \text{ for any } i > j, \quad i, j = 1, \ldots, k, \tag{3.11}$$

where $\underline{s}_{ij} = \underline{s}_{ji}$, and $\overline{s}_{ij} = \overline{s}_{ji}$. Finally, for the four trains in the example, constraints (3.8) are the following:

$$\begin{aligned}
d_2 - d_1 &\in [14, 16]_{60}, \\
d_3 - d_1 &\in [29, 31]_{60}, \\
d_4 - d_1 &\in [44, 46]_{60}, \\
d_3 - d_2 &\in [14, 16]_{60}, \\
d_4 - d_2 &\in [29, 31]_{60}, \\
d_4 - d_3 &\in [14, 16]_{60}.
\end{aligned}$$

3.4.7 Safety Constraints

This section describes the constraints that ensure that the minimum headway time between trains is respected, and that trains do not meet and overtake one another on a track. First, consider two trains traveling in the same direction, at equal velocity. Meets and overtakes are no issue then, so we can concentrate on the headway time. The 1900 train from The Hague to Venlo and the 800 train from Haarlem to Maastricht form an example of this situation (see Section 3.4.3). From Boxtel until Eindhoven, the 1900 and 800 trains use the same infrastructure. There are some small nodes in between Boxtel and Eindhoven, but these are ignored in this example. Suppose that both trains take 10 minutes to travel from Boxtel to Eindhoven. Typically, a headway time of three minutes must be respected at any time between two consecutive trains. In particular, this holds for both the departure from a node, and for the arrival at the next node. This means that if some train leaves a node, the next train can not leave earlier than three minutes later. And if a train arrives at a node, the next train can not arrive earlier than three minutes later. Focusing on the departure from Boxtel, if the 1900 train leaves first, we have the constraint

$$d^{800}_{\text{Boxtel}} \quad - \quad d^{1900}_{\text{Boxtel}} \in \quad [3, 59]_{60}.$$

This constraint prevents the 800 train from leaving Boxtel 0, 1, or 2 minutes after the 1900 train does. Similarly, when the 800 train leaves first, the constraint is

$$d^{1900}_{\text{Boxtel}} \quad - \quad d^{800}_{\text{Boxtel}} \in \quad [3, 59]_{60}.$$

Taking the negative of the latter constraint gives

$$d^{800}_{\text{Boxtel}} \quad - \quad d^{1900}_{\text{Boxtel}} \in \quad [-59, -3]_{60}.$$

Because the time window in this constraint is modulo 60, we are free to add 60 to both the lower and upper bound, which results in

$$d^{800}_{\text{Boxtel}} \quad - \quad d^{1900}_{\text{Boxtel}} \in \quad [1, 57]_{60}.$$

Finally, intersecting the time windows of this constraint and the first constraint gives $[1, 57]_{60} \cap [3, 59]_{60} = [3, 57]_{60}$, which leads to the following safety constraint:

$$d^{800}_{\text{Boxtel}} \quad - \quad d^{1900}_{\text{Boxtel}} \in \quad [3, 57]_{60}.$$

The latter constraint can also be interpreted as: whenever the 1900 train leaves Boxtel, the 800 train should *not* leave Boxtel in the three minutes *after or before* the departure of the 1900 train.

A similar constraint can be constructed for the arrival of both trains in Eindhoven, and together these constraints in Figure 3.8 model the safety requirements between the 800 and 1900 trains for the track Boxtel-Eindhoven. Since both trains take 10

$$\boxed{\begin{array}{lcll} d^{800}_{\text{Boxtel}} & - & d^{1900}_{\text{Boxtel}} & \in \quad [3, 57]_{60}, \\ a^{800}_{\text{Eindhoven}} & - & a^{1900}_{\text{Eindhoven}} & \in \quad [3, 57]_{60}. \end{array}}$$

Figure 3.8: Safety constraints between Boxtel and Eindhoven for the 800 and 1900 trains, for equal velocities.

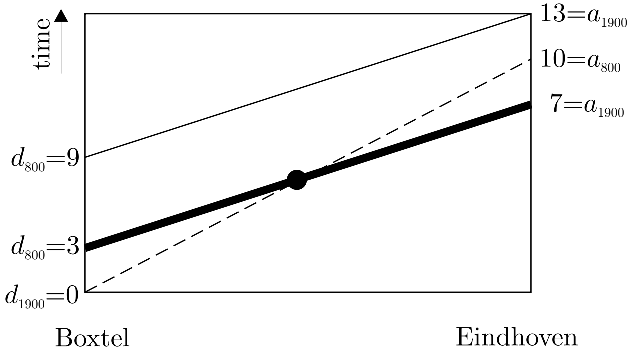


Figure 3.9: A feasible solution to the constraints in Figure 3.8.

minutes to travel from Boxtel to Eindhoven, we have the trip time relations

$$a^{800}_{\text{Eindhoven}} - d^{800}_{\text{Boxtel}} = [10]_{60},$$
$$a^{1900}_{\text{Eindhoven}} - d^{1900}_{\text{Boxtel}} = [10]_{60}.$$

Therefore, the second of the constraints in Figure 3.8 is already implied by the first one. This is also intuitively clear, since both trains leave Boxtel with a time difference of at least three minutes, and travel at the same velocity. Therefore, they will arrive in Eindhoven with the same time difference of at least three minutes.

Next, suppose that the 800 train is operated with a hypothetical new type of rolling stock that allows for an extreme increase in travel time. Therefore, it takes the 800 train only 4 minutes to travel from Boxtel to Eindhoven, whereas it still takes the 1900 train 10 minutes. Then, the situation depicted by the bold and the dashed lines in the time-space diagram in Figure 3.9 is a feasible solution to the constraints in Figure 3.8. It is clear that this solution violates the assumption that trains only overtake one another in nodes. Therefore, we need to add the difference in trip times between the two trains to the safety time window for the departures from Boxtel. That is, train 800 can leave Boxtel no earlier than $3 + (10 - 4) = 9$ minutes after train 1900 has left. Since the 1900 train is the slower train, the upper bound of the safety time window remains the same. This gives the safety constraints in Figure 3.10. Again, because of the fixed trip times, if the first constraint is satisfied, the second one can not be violated, and is therefore superfluous.

Next, we generalize the above example safety constraints.

$$\boxed{\begin{array}{llll} d^{800}_{\text{Boxtel}} & - & d^{1900}_{\text{Boxtel}} & \in & [9,57]_{60}, \\ a^{800}_{\text{Eindhoven}} & - & a^{1900}_{\text{Eindhoven}} & \in & [3,57]_{60}. \end{array}}$$

Figure 3.10: Headway times between the 800 and 1900 trains when the 800 train is faster.

3.4.8 General Safety Constraints

Consider a track $a = (n,m)$, and two trains t and t' that travel from n to m. Let r^t_a and $r^{t'}_a$ be the trip times of the trains t and t' between n and m. Suppose that train t' is the faster train of the two, so $r^t_a > r^{t'}_a$. From the above, it follows that, in order to prevent train t' from overtaking train t, we need to enforce

$$d^{t'}_n - d^t_n \notin (0, r^t_a - r^{t'}_a)_T,$$

Since, if train t' were to leave node n within $(r^t_a - r^{t'}_a)$ minutes after the departure of train t, then it would arrive at m before train t does. Since train t' is faster than train t, such a constraint is not needed for the reverse train order. The above constraint can be strengthened, since we know that the minimum headway times upon leaving n and entering m need to be respected.

Consider the time-space diagram shown in Figure 3.11. The infeasible dotted line in the figure shows that there should be a time difference of at least $X = r^t_a - r^{t'}_a + h$ between the departures of trains t and t' from n. This observation gives the following, stronger constraint

$$d^{t'}_n - d^t_n \notin (-h, r^t_a - r^{t'}_a + h)_T. \tag{3.12}$$

Using the cyclic nature of the timetable, this can be rewritten as

$$d^{t'}_n - d^t_n \in [r^t_a - r^{t'}_a + h, T - h]_T. \tag{3.13}$$

As was explained in Section 3.4.7, a safety constraint for entering node m is not needed because of the fixed trip times of the trains.

Next, consider a *single* track $a = (n,m)$, with a train t traveling from n to m, and a train t' traveling in the opposite direction. The trip times for the trains are r^t_a and $r^{t'}_a$. In order to avoid the trains from meeting one another on the track between n and m, we must ensure that

$$a^{t'}_n - d^t_n \notin (0, r^t_a + r^{t'}_a)_T. \tag{3.14}$$

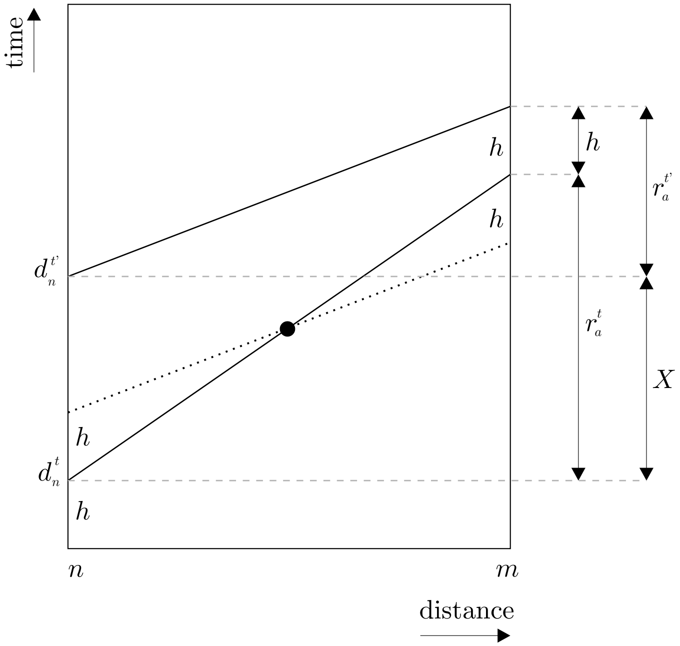


Figure 3.11: Prevent overtaking for fixed trip times.

If train t' were to arrive in n within $(r_a^t + r_a^{t'})$ minutes after train t has left, then train t' would meet train t somewhere between n and m. With a similar argument as above, the latter constraint can be strengthened using the headway constraints. After also rewriting this constraint, this leads to

$$a_n^{t'} - d_n^t \in [r_a^t + r_a^{t'} + h, T - h]_T. \tag{3.15}$$

Again, because of the fixed trip times, a constraint for node m is not needed. So, after train t has left node n, it takes quite some time before train t' can enter node n, namely the sum of the trip times of the two trains, plus the headway time h. On the other hand, after the arrival of train t' in node n, train t can leave from node n already after the headway time h. An obvious necessary condition for the existence of a feasible solution is $r_a^t + r_a^{t'} + h < T - h$. It follows that two trains traveling in opposite directions on a single track consume quite a bit of the available infrastructure capacity.

3.5 The Cyclic Railway Timetabling Model

This section generalizes the example constraints from the previous section. As such, we arrive at our mathematical model for cyclic railway timetabling. We first define the sets and parameters for the model. Then, we present the complete model. Appendix A contains an overview of the used notation.

3.5.1 Sets

The sets below contain the basic information for our mathematical model.

$\mathcal{G} = (\mathcal{N}, \mathcal{A} \cup \mathcal{A}^s)$ The railway network graph, consisting of nodes $\mathcal{N}$, regular tracks $\mathcal{A}$, and single tracks $\mathcal{A}^s$.

$\mathcal{N}$ The set of nodes in the railway network.

$\mathcal{A}$ The set of regular tracks $a = (n, m)$ in the railway network, with $n, m \in \mathcal{N}$.

$\mathcal{A}^s$ The set of single tracks $a = (n, m)$ in the railway network, with $n, m \in \mathcal{N}$.

$\mathcal{T}$ The set of trains.

A regular track is a track that is only used for one direction of travel. So, the regular tracks $a \in \mathcal{A}$ are directed. The single tracks $a \in \mathcal{A}^s$ are not directed, and for these tracks we somewhat sloppily write $(n, m) = (m, n)$. As was explained in Section 3.1.1, we may have multiple parallel tracks between two nodes. In that case, we need a third index in order to be able to distinguish between the parallel tracks. For clarity, this third index is omitted from our set definition, and the alternative model formulation in the next section does not need indices for tracks. The set $\mathcal{T}$ represents *all* trains in the system. Since a train line with frequency f yields f trains in each direction, $\mathcal{T}$ contains $2f$ trains for that line.

We use the indices n and m for the node set $\mathcal{N}$, the indices a and (n, m) for the track set $\mathcal{A}$, and the index t for the train set $\mathcal{T}$.

We further use the following derived sets to model the timetable requirements.

$\mathcal{N}^t \subseteq \mathcal{N}$ The set of nodes that train t visits.

$\mathcal{A}^t \subseteq \mathcal{A} \cup \mathcal{A}^s$ The set of tracks that train t travels along.

$\mathcal{T}_a$ For a track a, this set contains all pairs of trains (t, t') that travel along the track in the same direction, for which either t' is the faster train, or, in case of equal velocities, for which $t < t'$.

$\mathcal{T}_a^s$ For a single track $a = (n, m)$, this set contains all pairs of trains (t, t') that travel along the track in opposite directions, with train t departing from n, and train t' departing from m.

$\mathcal{F}_n^d, \mathcal{F}_n^a$ For a node n, these sets contain all trains t for which the departure respectively arrival time is fixed.

$\mathcal{S}_n$ For a node n, this set contains all train pairs $(t, t'), t < t'$, for which the departure times are to be synchronized at that node.

$\mathcal{C}_n$ For a node n, this set contains all train pairs $(t, t'), t < t'$, for which a connection or turn around constraint is required from train t to train t' at that node.

The reason for the somewhat awkward definition of the sets $\mathcal{T}_a$ and $\mathcal{T}_a^s$ will become clear below when we define the model. For the moment, it is important to see that each train pair appears at most once in these sets. That is, a specific set $\mathcal{T}_a$ or $\mathcal{T}_a^s$ never contains both (t, t') and (t', t).

The set $\mathcal{C}_n$ requires some extra explanation. If a connection relation is required at node n between the arrival of train t and the departure of train t', then $\mathcal{C}_n$ contains the element (t, t'). Similarly, if train t turns on train t' in node n, then $\mathcal{C}_n$ contains the element (t, t'). In the latter case, node n must be the destination node of train t and the origin node of train t'. So, for an element $(t, t') \in \mathcal{C}_n$, t is the arriving train, and t' is the departing train.

3.5.2 Parameters

The model uses the following parameters, which are all assumed to be integer valued.

T The cycle time of the timetable.

h The general headway time upon departure and arrival for any node.

r_a^t The trip time of train t for track a.

$[\underline{d}_n^t, \overline{d}_n^t]$ The dwell time window for train t at node n.

$[\underline{f}_n^t, \overline{f}_n^t]$ The fixed departure or arrival time window for train t at node n. In case of a completely fixed departure or arrival, we have $\underline{f}_n^t = \overline{f}_n^t$.

$[\underline{s}_n^{tt'}, \overline{s}_n^{tt'}]$ The time window for the synchronization of trains t and t' at node n.

$[\underline{c}_n^{tt'}, \overline{c}_n^{tt'}]$ The time window for the connection or turn around constraint between trains t and t' at node n.

We use a general headway time h here. If desired, one can also specify a headway time $h_a^{tt'}$ that depends on the type of the involved trains t and t', and on the track a. For example, the headway time between two fast trains may be larger than the headway time between two slow trains.

3.5.3 The Model

Using the above defined sets and parameters, we formally define the Cyclic Railway Timetabling Problem (CRTP) below. Recall that the model uses the following decision variables:

$a_n^t \in \{0, \dots, T-1\}$ The arrival time of train t at node n.

$d_n^t \in \{0, \dots, T-1\}$ The departure time of train t from node n.

Let a be the vector containing all a_n^t variables, and let d be the vector containing all d_n^t variables. For now, the objective will be to minimize the general function $F(a, d)$. The CRTP is formulated as the following integer program.

CRTP:

$$
\begin{array}{llll}
\text{Minimize} & F(a,d) & & (3.16\text{a}) \\
\text{subject to} & & & \\
a_m^t - d_n^t & = [r_a^t]_T & \text{for all } t \in \mathcal{T}, a = (n,m) \in \mathcal{A}^t & (3.16\text{b}) \\
d_n^t - a_n^t & \in [\underline{d}_n^t, \overline{d}_n^t]_T & \text{for all } t \in \mathcal{T}, n \in \mathcal{N}^t & (3.16\text{c}) \\
d_n^{t'} - d_n^t & \in [\underline{s}_n^{tt'}, \overline{s}_n^{tt'}]_T & \text{for all } n \in \mathcal{N}, (t,t') \in \mathcal{S}_n & (3.16\text{d}) \\
d_n^{t'} - a_n^t & \in [\underline{c}_n^{tt'}, \overline{c}_n^{tt'}]_T & \text{for all } n \in \mathcal{N}, (t,t') \in \mathcal{C}_n & (3.16\text{e}) \\
d_n^{t'} - d_n^t & \in [r_a^t - r_a^{t'} + h, T - h]_T & \text{for all } a = (n,m) \in \mathcal{A}, (t,t') \in \mathcal{T}_a & (3.16\text{f}) \\
a_n^{t'} - d_n^t & \in [r_a^t + r_a^{t'} + h, T - h]_T & \text{for all } a = (n,m) \in \mathcal{A}^s, (t,t') \in \mathcal{T}_a & (3.16\text{g}) \\
d_n^t & \in [\underline{f}_n^t, \overline{f}_n^t] & \text{for all } n \in \mathcal{N}, t \in \mathcal{F}_n^d & (3.16\text{h}) \\
a_n^t & \in [\underline{f}_n^t, \overline{f}_n^t] & \text{for all } n \in \mathcal{N}, t \in \mathcal{F}_n^a & (3.16\text{i}) \\
a_n^t, d_n^t & \in \{0, \dots, T-1\} & \text{for all } t \in \mathcal{T}, n \in \mathcal{N}_t & (3.16\text{j})
\end{array}
$$

Constraints (3.16b) and (3.16c) ensure the correct trip times and dwell times, respectively. Next, constraints (3.16d) model the departure synchronization requirements. The connection and turn around requirements are modeled by constraints (3.16e). Constraints (3.16f) and (3.16g) guarantee that trains do not meet or overtake on a track, and that the minimum headway time h is respected at all times. These constraints are required for each pair of trains that use a track. Moreover, because of the definition of the sets $\mathcal{T}_a$ and $\mathcal{T}_a^s$, it is always the lower bound of the time windows in (3.16f) and (3.16g) which should include the trip times of the trains. Constraints (3.16h) and (3.16i) represent the fixed departure and arrival times, respectively. Note that these constraints are not periodic. The domain of the decision variables d_n^t and a_n^t is specified in (3.16j). Finally, recall from Section 3.3 that each of the constraints (3.16b)–(3.16g) contains an integer variable p that models the modulo T operation.

Remark 3.1. The next chapter shows that the structure of the integer program (3.16) is such, that the integrality constraints (3.16j) can be relaxed, with the guarantee that an optimal solution still consists of integer arrival variables a_n^t and departure variables d_n^t. So, formulation (3.16) can be written as a mixed integer linear program (MIP) with continuous variables a_n^t and d_n^t, and integer variables p modeling the modulo T operations in the constraints.

3.6 A Constraint Graph Formulation

Each of the constraints in formulation (3.16) defines a relation between two events, either a relation (departure, departure), (arrival, arrival), (arrival, departure), or (departure, arrival). Moreover, each constraint states that one event should take place some time later than the other, and the required time difference between the two events is either stated as a fixed value, in the case of trip times, or as a time window. An instance of the CRTP can therefore be described by a so-called constraint graph. Each departure variable d_n^t and each arrival variable a_n^t induces a node in the constraint graph, and each constraint induces an arc.

Before we describe the CRTP in terms of a constraint graph, it is convenient to introduce the concept of an *event*. In the CRTP, an event is defined by a combination of

- a train t,
- a node n in the railway network, and
- either an arrival a or a departure d.

So an event i is defined by either a triple (t, n, a) or a triple (t, n, d), and corresponds either to an arrival variable a_n^t, or to a departure variable d_n^t. A set N of all events that need to be scheduled can be deduced from the infrastructure and train line information. Rather than using separate decision variables d_n^t and a_n^t for departure and arrival events, we introduce the following general event time variables:

$v_i \in \{0, \ldots, T-1\}$ The time instant at which event $i \in N$ takes place.

Each constraint in the CRTP is defined on a pair (i, j) of events, and a constraint is characterized by its time window. Instead of using specific parameters for each of the different types of time windows, we use the following general time window notation:

$[l_{ij}, u_{ij}]_T$	The periodic time window for the constraint involving events i and j.
$p_{ij} \in \mathbb{Z}$	The integer variable modeling the cyclic nature of the periodic constraint involving events i and j.

The general time window $[l_{ij}, u_{ij}]_T$ should be interpreted as: event j should take place between l_{ij} and u_{ij} minutes after event i takes place. For trip time constraints, we set both l_{ij} and u_{ij} equal to the trip time. Regarding the time windows, we assume $0 \leq l_{ij} \leq T-1$, and $0 \leq u_{ij} - l_{ij} \leq T-2$. The next chapter discusses these assumptions in detail.

Using these general time windows, each of the constraints (3.16b)–(3.16g) can be written as

$$v_j - v_i \in [l_{ij}, u_{ij}]_T, \tag{3.17}$$

or, including the integer variable p_{ij}, as

$$l_{ij} \leq v_j - v_i + T p_{ij} \leq u_{ij}, \text{ with } p_{ij} \in \mathbb{Z}. \tag{3.18}$$

The fixed departure and arrival time constraints (3.16h), (3.16i) are modeled by

$$\underline{f}_i \leq v_i \leq \overline{f}_i, \tag{3.19}$$

where the new parameters $\underline{f}_i$ and $\overline{f}_i$ correspond to the fixed departure and arrival time window bounds $\underline{f}_n^t$ and $\overline{f}_n^t$.

The constraint graph G is now defined as follows. The node set of the graph is equal to the event set N. The arc set A of the constraint graph corresponds to the constraints, so for each event pair (i, j) for which a constraint is defined, the graph contains an arc. We associate the general time window $[l_{ij}, u_{ij}]$ with an arc (i, j), where the values for l_{ij} and u_{ij} are obtained from the time windows in formulation (3.16). Let l and u be the vectors containing all l_{ij} and u_{ij} parameters. Then, a CRTP instance is completely described by the constraint graph $G = (N, A, l, u)$ and the cycle time T. Let v be the vector consisting of all variables $v_i, i \in N$, and let $F(v)$ be the general objective function in terms of the variables v_i. Having defined the constraint graph G, the CRTP can be reformulated as

$$
\begin{array}{lllll}
\textbf{CRTP:} & \text{Minimize} & F(v) & & (3.20\text{a}) \\
& \text{subject to} & l_{ij} \leq v_j - v_i + Tp_{ij} \leq u_{ij} & \text{for all } (i,j) \in A & (3.20\text{b}) \\
& & \underline{f}_i \leq v_i \leq \overline{f}_i & \text{for all } i \in N & (3.20\text{c}) \\
& & v_i \in \{0, \ldots, T-1\} & \text{for all } i \in N & (3.20\text{d}) \\
& & p_{ij} \in \mathbb{Z} & \text{for all } (i,j) \in A & (3.20\text{e})
\end{array}
$$

Here, we explicitly wrote down the integer variables that model the modulo T operations in the constraints (3.20b). In the translation from (3.16) to (3.20), it may occur that multiple constraints are defined between a single pair of events i and j, which would cause parallel arcs in the constraint graph G. So, formally we need to distinguish between these parallel arcs, for example by a third index k on the time window constraints and arcs in (3.20b). For clarity reasons we omit this third index.

In our opinion, formulation (3.20) is clearer and more compact than formulation (3.16), which is cluttered by the various types of time window constraints. Therefore, we use the constraint graph formulation for the CRTP throughout this thesis, and only return to the original formulation (3.16) when discussing specific types of constraints.

3.7 Objective Functions

As was set out in Section 3.2, we consider the following objectives:

- minimizing passenger travel time,
- maximizing timetable robustness,
- minimizing the required number of rolling stock units, and
- minimizing the violation of the initial constraints in case of an infeasible instance.

For each of the four objectives, we define a function that expresses the objective in terms of the decision variables:

F_t for passenger travel time,

F_r for timetable robustness,

F_s for the number of rolling stock units used in the timetable,

F_v for the amount of violation of the initial constraints.

Each of these functions can be substituted for the general objective function $F(v)$. The robustness function F_r is being minimized too, even though the objective is to maximize the timetable robustness. One can also assign weights to the four functions above, and use a weighted multi-objective function.

The following subsections describe the three functions $F_t, F_r,$ and F_v in more detail. For the rolling stock function F_s, we require some ideas that are introduced in Chapter 5. Therefore, the rolling stock function is not defined until Chapter 6. Finally, we describe how a quadratic variant for each of the three functions $F_t, F_r,$ and F_v can be incorporated into the CRTP.

As will become clear below, the objective functions depend on process times, rather than on event times. However, process times are not discussed until Chapter 5. For the moment, it suffices to define the process time decision variables $x_{ij}, (i,j) \in A$, as

$$x_{ij} = v_j - v_i + Tp_{ij}. \tag{3.21}$$

That is, the process time x_{ij} represents the value in the time window $[l_{ij}, u_{ij}]$ that corresponds to the partial solution v_j, v_i, p_{ij}. As an example, for a dwell time constraint, the process time represents the number of minutes that a train dwells at a station. In principle, any of the functions below can thus be written in terms of $v_j, v_i,$ and p_{ij}. But we find it clearer to use the process times x_{ij} for describing the objective functions.

3.7.1 Minimizing Passenger Travel Time

Given the assumption that train trip times are fixed, the only two decision variables in the model that influence the total journey time for a passenger are connection times and dwell times. For a connection between trains t and t' at node n, each minute of connection time above the minimum connection time $\underline{c}_n^{tt'}$ adds to the minimum possible travel time of the transferring passengers. Similarly, each minute of dwell time above the minimum dwell time $\underline{d}_n^t$ adds to the minimum possible travel time of the passengers that remain seated in train t at station n. So, by minimizing these excess connection times and excess dwell times, we minimize the passenger travel time.

Let A^c be the set of arcs corresponding to the connection constraints in (3.16e), and let A^d be the set of arcs corresponding to the dwell time window constraints (3.16c). The excess connection and dwell times are given by $(x_a - l_a)$ for $a \in A^c \cup A^d$. Further, define:

w_a The objective function weight for the excess connection time or the excess dwell time on arc $a \in A^c \cup A^d$.

The weight w_a specifies the importance of certain connection times and dwell times. Factors that influence the weight include the number of passengers involved in a connection, the number of passengers that remain in the dwelling train, or the types of the involved trains. The following expression represents the weighted excess connection and dwell time in a timetable:

$$\sum_{a \in A^c \cup A^d} w_a(x_a - l_a). \tag{3.22}$$

All terms $(-w_a l_a)$ together yield a constant. Since our goal is to minimize the above expression, the passenger travel time objective function can therefore be written as

$$F_t = \sum_{a \in A^c \cup A^d} w_a x_a. \tag{3.23}$$

Note that F_t does *not* measure the total travel time in a timetable. Rather, it reflects the factors that influence the travel time in a weighted manner.

In Chapter 6 we present a variable trip time extension of the CRTP. There, it becomes clear that the difficult part of introducing variable trip times into the model lies in the safety constraints (3.16f) and (3.16g), since these include the trip time in their time windows. If we only consider the trip time constraints (3.16b), then variable trip times can be modeled rather straightforwardly. Define a trip time window $[\underline{r}_a^t, \overline{r}_a^t]$ by

$\underline{r}_a^t$ The fastest trip time for train t along track a.

$\overline{r}_a^t$ The slowest trip time for train t along track a.

Then, variable trip times are incorporated in the CRTP model by replacing the constraints (3.16b) by

$$a_m^t - d_n^t \in [\underline{r}_a^t, \overline{r}_a^t]_T \text{ for all } t \in \mathcal{T}, a = (n,m) \in \mathcal{A}^t.$$

So, when variable trip times also influence the total travel time, it is quite easy to incorporate them into F_t. For passengers to travel as fast as possible, trains should run at the highest possible velocity, that is, at the smallest possible trip times. Therefore, in order to incorporate variable trip times in the travel time objective, we define a set A^{var} containing the variable trip arcs, specify a weight for the excess trip times, and include the set A^{var} in the summation in (3.23).

3.7.2 Maximizing Timetable Robustness

We argued in Section 3.2 that the timetable robustness can be improved by pulling apart trains that share a track. If there is a lot of time between consecutive trains, then this time can be used as a buffer in case of delays. A trade-off has to be made, however, between increasing the inter-train time on one hand, and meeting the timetable service requirements on the other hand. For example, because of defined train connections, certain trains have to be at a station at somewhat the same time. So, if the involved trains share the track for entering or leaving the station, then the connection requirement implies that these trains can not be pulled apart too far.

Let A^r be the set of arcs corresponding to the safety constraints (3.16f) and (3.16g). Then, the departure times of the trains are pulled apart when the process time x_a for $a \in A^r$ is close to the middle of the time window $[l_a, u_a]$. So, merely minimizing or maximizing the process times for the set A^r does not suit our goal.

Therefore, define the parameter μ_a as

$$\mu_a = \lfloor \tfrac{1}{2}(u_a - l_a) \rfloor \text{ for all } a \in A^r.$$

That is, μ_a denotes the middle of the time window $[l_a, u_a]$. For technical reasons, μ_a is rounded down to ensure that it is integer valued. Further, introduce an auxiliary variable δ_a for all $a \in A^r$. The auxiliary variable δ_a is constrained as follows:

$$\begin{aligned} \delta_a &\geq \mu_a - x_a, \\ \delta_a &\geq x_a - \mu_a. \end{aligned}$$

So $\delta_a \geq |x_a - \mu_a|$. Thus, minimizing δ_a means pushing x_a towards μ_a, and thus away from the time window bounds l_a and u_a.

Finally, as before, we define a weight w_a for all $a \in A^r$. For a pair of trains corresponding to a safety arc a, the weight indicates how likely the trains are to interfere with one another during the execution of the timetable. The more likely the trains are to interfere, the higher the value of w_a will be.

Using the above, the robustness objective function is defined as

$$F_r = \sum_{a \in A^r} w_a \delta_a. \tag{3.24}$$

Recall that the function F_r is minimized. This ensures that the auxiliary variables δ_a push the process times x_a towards μ_a. In other words, minimizing F_r means maximizing the timetable robustness.

3.7.3 Minimizing Initial Constraint Violation

When the specifications for a CRTP instance are too tightly defined, a timetable respecting all requirements may not exist. In such a case, we propose to choose a relaxation subset of the arcs. The time windows of the arcs in the relaxation subset are relaxed by decreasing the lower bounds, and increasing the upper bounds. A solution to the thus relaxed problem instance will violate the initial constraints. Our goal in this case is to find a solution to the relaxed instance that minimally violates the initial constraints.

Let the set $A^v \subseteq A$ be the relaxation subset. In general, A_v is not allowed to contain any arcs corresponding to safety constraints, since that may result in a timetable that does not comply with safety regulations. For each $a \in A^v$, define the following adjusted time window bounds:

l'_a	The relaxed time window lower bound.
u'_a	The relaxed time window upper bound.

Clearly, we require $0 \leq l'_a \leq l_a$, and $u_a \leq u'_a$. Moreover, we also require $u'_a - l'_a < T - 2$, as before for the original time window $[l_{ij}, u_{ij}]$.

Further, we define the following auxiliary decision variable for each $a \in A^v$:

σ_a	The used space of the relaxed time window $[l'_a, u'_a]$.

We use these variables to penalize the use of the extra space in the relaxed time windows $[l'_a, u'_a]$. To that end, the variables σ_a are constrained as follows:

$$\begin{aligned}\sigma_a &\geq l_a - x_a,\\ \sigma_a &\geq x_a - u_a,\\ \sigma_a &\geq 0.\end{aligned}$$

As in the case of maximizing timetable robustness, minimizing σ_a means driving x_a away from the relaxed time window bounds. If $l_a \leq x_a \leq u_a$, then σ_a takes the value zero. Thus, σ_a only takes a positive value when the extra space in the relaxed time window is used. With the relaxed time windows $[l'_a, u'_a]$, we minimize the initial constraint violation by minimizing the function

$$F_v = \sum_{a \in A^v} \sigma_a. \tag{3.25}$$

Variants of the function F_v are of course possible. One can think of not relaxing the time window lower bounds that stem from planning norms, such as the minimum dwell time or the minimum connection time. As before, one may associate weights w_a with the arcs $a \in A_v$. These weights reflect that it is less problematic to violate small weight constraints than it is to violate large weight constraints. As an example, safety constraints can typically not be violated, unless it is absolutely necessary for obtaining a solution, thereby indicating a serious infrastructure capacity problem. In the latter case the solution only indicates the infrastructure capacity problem, and does *not* provide a timetable. So safety arcs have a large weight, whereas connection arcs or dwell arcs typically have a smaller weight. This leads to the weighted constraint violation objective function $F_v = \sum_{a \in A^v} w_a \sigma_a$.

3.7.4 Quadratic Objective Functions

The objective functions F_t, F_r, and F_v are linear in the process times x_a. These linear functions can be used for situations in which each extra minute is equally important. Consider, however, the situation in which a passenger perceives each additional minute of extra travel time as worse than the previous one. A quadratic objective function is useful for expressing this perception. In the case of maximizing robustness, a quadratic objective function can be used to model that the first minute that trains are pulled apart is more valuable than the second minute, etc. In the same vein, for minimizing the initial constraint violation, a quadratic objective function can indicate that the first used minute of the relaxed time window weighs less than the second, etc.

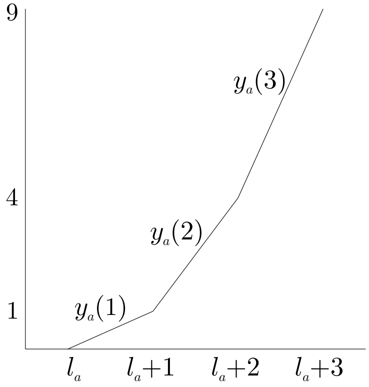


Figure 3.12: The lines $y_a(1), y_a(2)$, and $y_a(3)$ for linearizing the function $g(x_a)$.

Consider the quadratic objective function

$$F_q = \sum_{a \in A_q} (x_a - l_a)^2.$$

For some subset $A_q \subseteq A$, the function F_q measures, quadratically, how far the variables $x_a \in A_q$ deviate from their time window lower bounds l_a. This section describes how F_q can be linearized in such a way, that the linearization can be incorporated in the CRTP model.

To that end, consider a single arc $a \in A_q$, and the function $g(x_a) = (x_a - l_a)^2$. For all integer values $x_a \in [l_a, u_a]$, the function $g(x_a)$ passes through the points $(x_a, g(x_a))$ defined by

$$(l_a + \lambda, \lambda^2) \qquad \lambda = 0, \ldots, u_a - l_a.$$

For $\lambda = 0, \ldots, u_a - l_a - 1$, the line that passes through the consecutive points $(l_a + \lambda, \lambda^2)$ and $(l_a + \lambda + 1, (\lambda + 1)^2)$ is described by

$$y_a(\lambda) = \alpha(\lambda)x_a + \beta(\lambda) \qquad \lambda = 0, \ldots, u_a - l_a - 1,$$

where $\alpha(\lambda)$ and $\beta(\lambda)$ are defined by

$$\begin{aligned} \alpha(\lambda) &= 2\lambda + 1, \\ \beta(\lambda) &= -\lambda(\lambda + 1) - (2\lambda + 1)l_a. \end{aligned}$$

Figure 3.12 illustrates the lines $y_a(1), y_a(2)$, and $y_a(3)$.

In order to linearize the quadratic function $g(x_a)$, we introduce an auxiliary variable $f_a \geq 0$. This variable is bounded from below as follows

$$f_a \geq y_a(\lambda) \text{ for all } \lambda = 0, \ldots, u_a - l_a - 1. \tag{3.26}$$

The basic idea is to minimize the auxiliary variable f_a. When doing so, f_a takes a value that lies on one of the lines $y_a(\lambda)$. Moreover, if the process time variable x_a is integer, then the minimum value for f_a lies at the intersection of two lines $y_a(\lambda)$ and $y_a(\lambda+1)$. Therefore, when x_a is integer, f_a takes exactly the value $g(x_a)$.

For the general quadratic objective function $F_q = \sum_{a \in A_q} g(x_a)$, we apply the above procedure to each arc $a \in A_q$, and minimize the sum of all auxiliary variables f_a. This gives the following:

$$\begin{aligned}
&\text{Minimize} && \sum_{a \in A_q} f_a && && \text{(3.27a)}\\
&\text{subject to} && f_a \geq y_a(\lambda) && \text{for all } a \in A_q, \lambda = 0, \ldots, u_a - l_a - 1 && \text{(3.27b)}\\
& && f_a \geq 0 && \text{for all } a \in A_q && \text{(3.27c)}
\end{aligned}$$

By using the function $\sum_{a \in A_q} f_a$ and the constraints (3.27b), a quadratic objective function can be incorporated in the CRTP. So, this procedure can be applied to each of the three objective functions $F_t, F_r,$ and F_v. We call the corresponding linearized quadratic objective functions $F_{t,q}, F_{r,q},$ and $F_{v,q}$, respectively.

Remark 3.2. The quadratic functions $g(x_a)$ are approximated such that the approximation is exact for every integer value in the time window $[l_a, u_a]$. For wide time windows, this results in many linear inequalities that bound the variables f_a from below. In such a case, a compromise is to approximate the functions $g(x_a)$ such that the approximation is exact only for $\lambda = 0, \ldots, \lambda_{\max}$, and to approximate to remainder of $g(x_a)$ by the last linear inequality $f_a \geq y_a(\lambda_{\max})$.

3.8 The Structure of CRTP Constraint Graphs

A CRTP constraint graph $G = (N, A, l, u)$ has a very specific structure. This section describes that structure. First, we focus on trip time arcs, dwell time arcs, and turn around arcs, which describe the behavior of trains during their trips. Then, we describe safety arcs and frequency arcs, which ensure the spacing of trains on stations and tracks. Finally, the synchronization of trains through connection arcs is described.

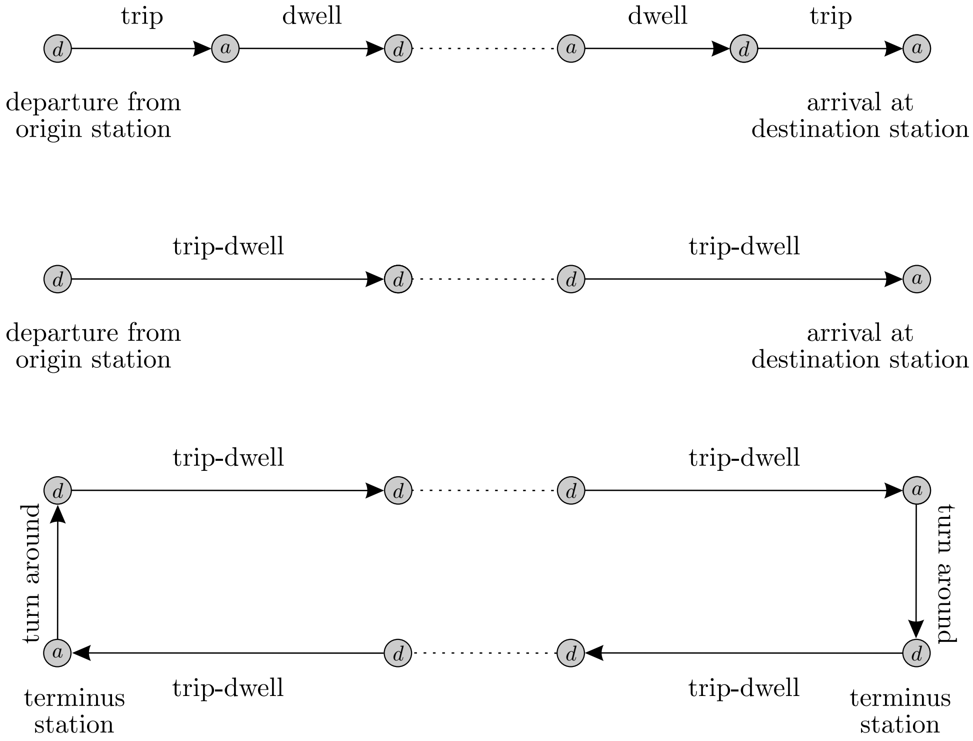


Figure 3.13: Train path P_t, merged train path P_t, and train cycle $C_{tt'}$.

3.8.1 Train Paths and Train Cycles

Consider a train t on the journey from its origin station to its destination station. Figure 3.2 showed the involved constraints for the journey of the 1900 train from The Hague Central to Venlo. From this figure, it is clear that the total journey of a train t defines a path P_t in G, starting and ending with a trip arc, and in between consisting alternatingly of dwell arcs and trip arcs, see Figure 3.13. The path P_t is called the *train path* of train t. Since all trip arcs represent equality constraints, a trip arc (i, j) and its successor dwell arc (j, k) can be merged into one joint *trip-dwell* arc (i, k). Such a merging procedure deletes all arrival nodes, and yields a *merged train path*, consisting of merged trip-dwell arcs, and of nodes representing train departures only, see Figure 3.13. For a more detailed description of the substitution of trip arcs, we refer to Section 7.2.

After train t arrives at its destination, it is common practice for its rolling stock composition to be assigned to the reverse journey, say that of train t'. That means that the train paths P_t and $P_{t'}$, together with the turn around arcs for both termini, form a cycle $C_{tt'}$, see Figure 3.13. Such a cycle $C_{tt'}$ is called a *train cycle*.

3.8.2 Safety Cliques

Consider the merged train paths P_t for four trains $t_1, \dots, t_4$. The four trains follow the routes depicted in the upper part of Figure 3.14, and their merged train paths have been drawn, again as solid arcs, in the lower part of Figure 3.14. Suppose that the trains use the same track for the joint part of their routes. That means that safety arcs are defined on each pair of departure nodes on the joint route. These safety arcs are shown in Figure 3.14 as dashed arcs.

In general, for a track $a \in \mathcal{A} \cup \mathcal{A}^s$, and a set of trains $t_1, \dots, t_k \in \mathcal{T}^a$ using track a, a *safety clique* K_a is the sub-graph formed by the nodes representing the departures of the trains, and the safety arcs between them. The term safety clique stems from the fact that the structure forms a clique on the k departure nodes. Figure 3.14 shows the constraint graph for the departures from four subsequent tracks, which means that the constraint graph contains four safety cliques. The left-most and right-most safety cliques consist of three nodes and three arcs, the two safety cliques in between consist of four nodes and six arcs.

Suppose that two trains use the same track, which defines a safety constraint between them, and that the same two trains are also synchronized through a frequency constraint. In this case, one can view the frequency constraint as a tight safety constraint. In fact, a frequency constraint usually has a narrower time window than the corresponding safety constraint, and therefore the former constraint dominates the latter. Such frequency arcs are also considered to be part of a safety clique.

3.8.3 Station Graphs

This section describes the *station graph*, the part of the constraint graph that corresponds to a station n. A station graph consists of connection arcs, dwell arcs, safety arcs, and frequency arcs. In describing station graphs, we consider original dwell arcs, instead of merged trip-dwell arcs.

Connection arcs and dwell arcs are directed from an arrival node to a departure node (see Section 3.4.3). This means that the connection and dwell arcs for a station n induce a *bipartite dwell-connection graph* (N_n^a, N_n^d, A_n). The node sets N_n^a and N_n^d are formed by the arrival and departure nodes, respectively. In Figure 3.15, the arrival node set N_n^a and the departure node set N_n^d have been drawn on the left and right side, respectively. The arc set A_n of a bipartite connection graph contains dwell and connection arcs, which are all directed from N_n^a to N_n^d. Consider four

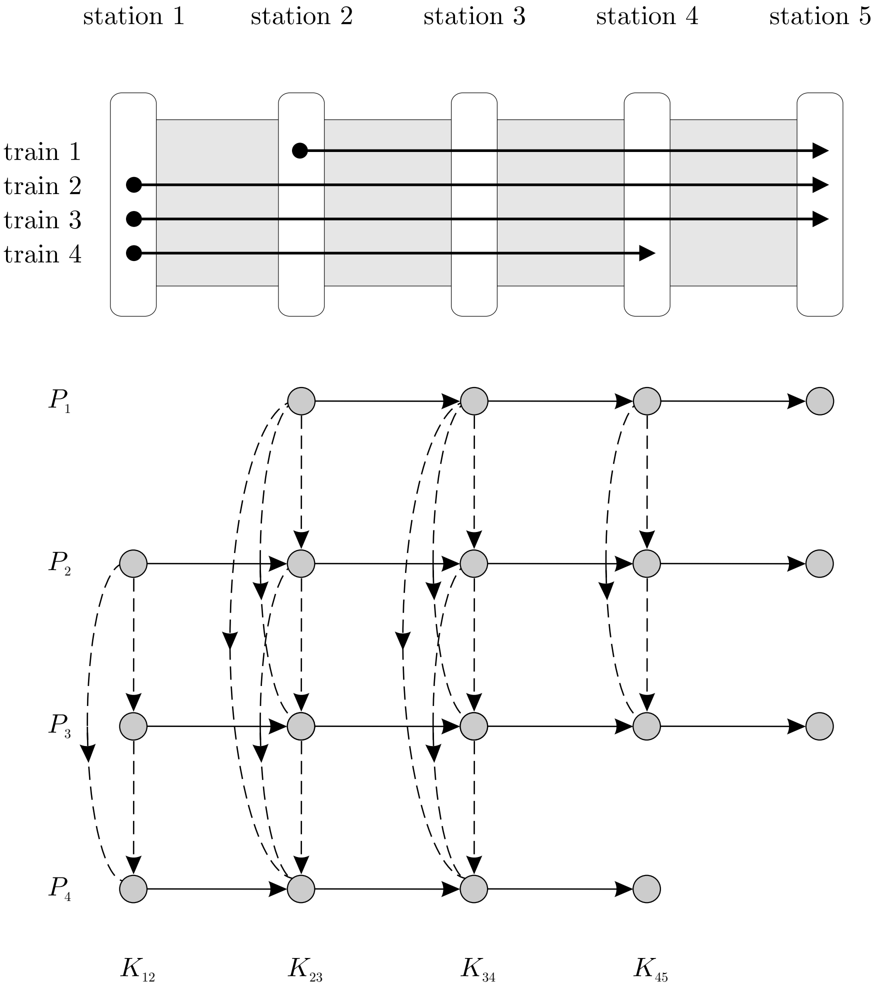


Figure 3.14: Trajectories for trains $t_1, \ldots, t_4$, and the corresponding train paths $P_1, \ldots, P_4$, and safety cliques $K_{12}, \ldots, K_{45}$.

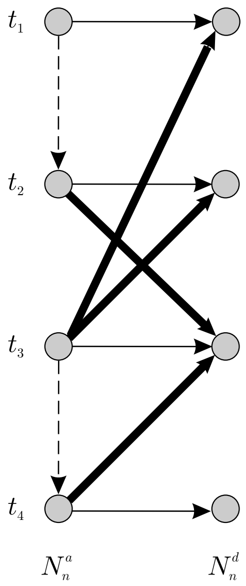


Figure 3.15: Station graph G_n^S with arrival and departure node sets N_n^a and N_n^d.

trains $t_1, \ldots, t_4$ dwelling at station n, and suppose that the following connections should be realized at n:

- $t_2 \longrightarrow t_3$,
- $t_3 \longrightarrow t_2$,
- $t_3 \longrightarrow t_1$,
- $t_4 \longrightarrow t_3$.

These connections give the bold connection arcs in Figure 3.15. Further, the solid arcs going from arrival to departure nodes represent the four dwell arcs.

Finally, suppose that trains t_1 and t_2 arrive at and depart from station n using the same track, and that trains t_3 and t_4 enter and leave station n using a second track. That gives the dashed safety arcs in Figure 3.15.

For a station n, the sub-graph formed by (N_n^a, N_n^d, A_n) and all adjacent safety arcs is called the *station graph* G_n^S, see Figure 3.15. A station graph may also contain turn around arcs, namely when the station is a terminus for one of the trains. A final remark regards small stations where no connections have been defined. For such small stations, we do not define station graphs. The same holds for nodes that do not represent stations, but crossings, junctions, etc. (see Section 3.1.1).

3.8.4 The CRTP Graph Structure

Train paths and train cycles, safety cliques, and station graphs are the three building blocks of a CRTP graph. Consider a railway network $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, and a corresponding instance of the CRTP described by $G = (N, A, l, u)$. One can think of G as being placed on top of $\mathcal{G}$. Each track $a \in \mathcal{A}$ gives rise to a safety clique K_a, and each station $n \in \mathcal{N}$ gives a station graph G_n^S. Connecting all train paths, safety cliques, and station graphs results exactly in the CRTP graph G. Note that the train paths and cycles, safety cliques, and station graphs have a certain overlap with one another.

As an example, consider Figure 3.16. The upper part sketches a small railway network with 7 stations and 3 trains. The stations are represented by the white blocks labeled A through G. The tracks between the stations are represented by the gray bars. The trains calling at a station are written above the blocks. The three trains have the following routes:

Train 1: $A \rightarrow C \rightarrow D \rightarrow E$.

Train 2: $A \rightarrow C \rightarrow D \rightarrow F$.

Train 3: $B \rightarrow C \rightarrow D \rightarrow G$.

The lower part of Figure 3.16 shows the corresponding constraint graph G. For each station, a rectangle is drawn around the nodes corresponding to that station. The solid arcs in the constraint graph represent the train paths P_t. Note that these are regular train paths, *not* merged train paths.

All safety arcs are dashed. As explained in Section 3.4, it suffices to state the safety constraints for the departures of trains only. Trains 1 and 2 leave station A using the same track, so we have a safety arc between the departure nodes of trains 1 and 2. Trains 1, 2, and 3 use the same track to travel from station C to station D. Therefore, we have safety arcs between the three departure nodes at station C. Upon leaving station D, all trains use a different track, so there are no safety arcs defined for the departure from station D.

The connection arcs have been drawn in bold. Since the three trains head for three different destinations after leaving station D, connections are defined between all trains. So, passengers can travel from any of the stations A, B, C, and D, to any of the stations E, F, and G with a good connection. Figure C.1 in Appendix C shows the constraint graph for the Dutch intercity train network for 1997/1998.

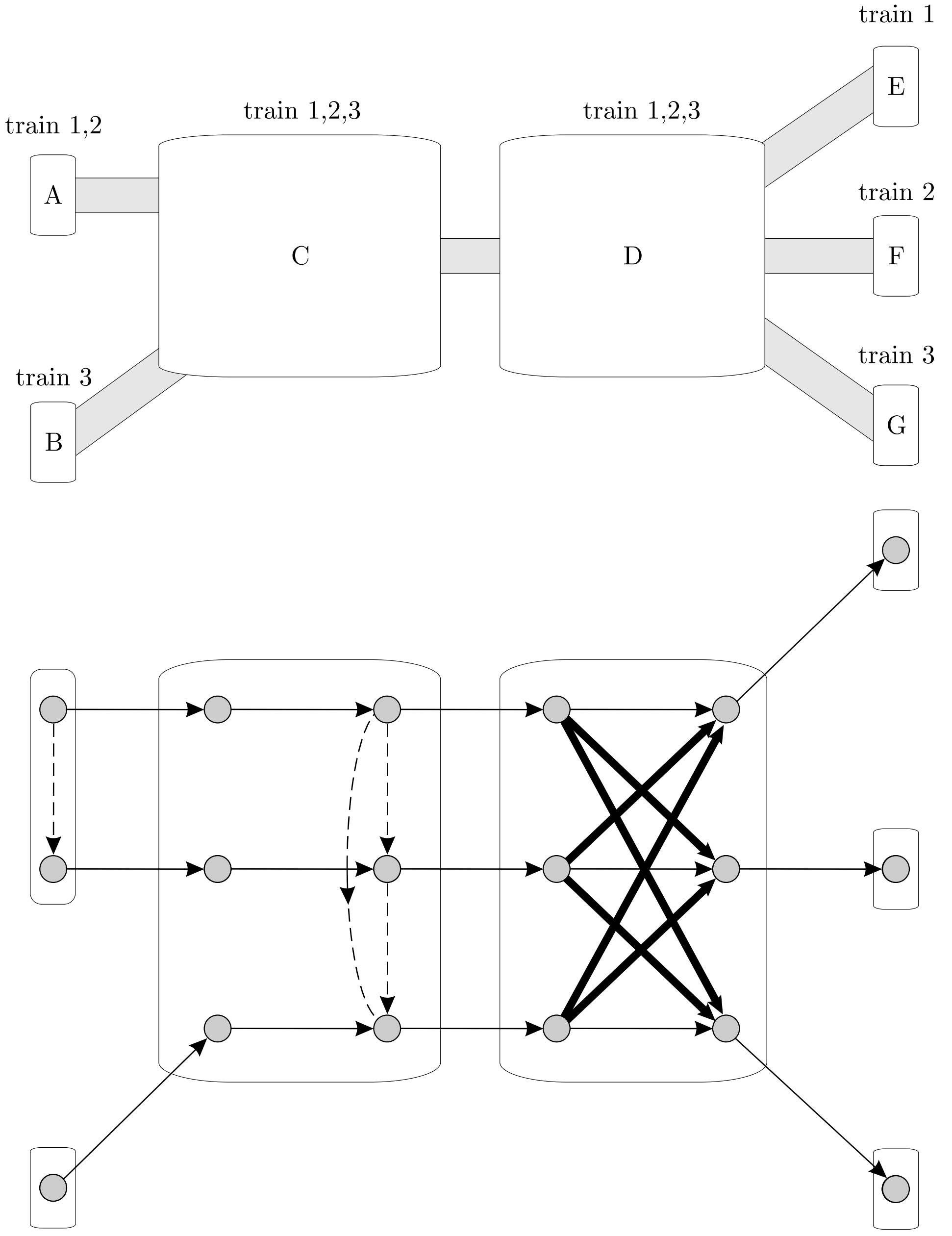


Figure 3.16: A small railway network with three trains, and the corresponding constraint graph.

Chapter 4

The Periodic Event Scheduling Problem

The constraint graph formulation for the CRTP described in the previous chapter uses constraints of the type

$$v_j - v_i + Tp_{ij} \in [l_{ij}, u_{ij}]. \tag{4.1}$$

This chapter describes the so-called Periodic Event Scheduling Problem, abbreviated to PESP, that was formulated by Serafini and Ukovich (1989a). The PESP provides a general framework for constraints of type (4.1).

We first define the PESP. Next, Section 4.2 discusses some theoretical results that follow from its constraint graph representation. Section 4.3 derives some useful properties of the PESP. Cyclic sequences are the topic of Section 4.4, and Section 4.5 describes an existing complexity result that proves the NP-completeness of the PESP. We conclude the chapter with a literature review on solution methods for the PESP.

4.1 Definition of the PESP

The PESP considers the scheduling of periodically recurring events under periodic time window constraints. For railway timetabling, one should think of an event as a combination (train, node, arrival) or (train, node, departure), as defined in Section 3.6. The PESP is formally defined as:

Definition 4.1 (PESP). Given a set N of events, a set $A \subseteq N \times N$, a cycle time T, and time windows $[l_{ij}, u_{ij}]$ for all $(i, j) \in A$, the Periodic Event Scheduling

Problem is to find a periodic schedule $v_i \in [0, T), i \in N$, satisfying

$$(v_j - v_i) \text{ modulo } T \in [l_{ij}, u_{ij}] \text{ for all } (i,j) \in A,$$

or to conclude that no such schedule exists.

So the PESP considers a feasibility problem. Because of the periodicity of the schedule, an event i that is scheduled to take place at time instant $v_i \in [0, T)$, in fact also takes place at all time instants $v_i + zT, z \in \mathbb{Z}$.

For the time windows $[l_{ij}, u_{ij}]$, we assume $0 \leq l_{ij} < T$ and $0 \leq u_{ij} - l_{ij} < T$. These assumptions are without loss of generality, the first one because of the periodic nature of the PESP constraints, and the second one because a time window which is wider than T imposes no periodic restriction. The PESP allows for multiple constraints for a single pair of events. That is, for an event pair (i,j) there may be M constraints, giving

$$v_j - v_i + \in [l^k_{ij}, u^k_{ij}]_T \text{ for all } k = 1, \ldots, M.$$

However, for clarity reasons we omit the third index k while discussing the PESP.

Let v be the vector consisting of all variables v_i, and let p be a vector containing an integer variable p_{ij} for each $(i,j) \in A$. A mathematical programming formulation for the PESP is then given by

$$\begin{array}{llll}
\textbf{PESP:} & \text{Find a solution} & (v, p) & \\
& \text{satisfying} & l_{ij} \leq v_j - v_i + Tp_{ij} \leq u_{ij} & \text{for all } (i,j) \in A \quad (4.2a) \\
& & 0 \leq v_i < T & \text{for all } i \in N \quad (4.2b) \\
& & p_{ij} \in \mathbb{Z} & \text{for all } (i,j) \in A \quad (4.2c)
\end{array}$$

The CRTP formulation (3.20) is quite similar to the above formulation of the PESP. The PESP formulation also uses periodic time windows, and an integer variable to model that periodicity. The only differences are the bounds on the CRTP event time variables (3.20c), and the domains of the event time variables (3.20d). Section 3.4.4 showed that the constraints (3.20c) can also be written as periodic constraints of the form (4.2a). Moreover, the next section shows that the structure of the PESP is such, that a feasible PESP instance has an all-integer solution (v, p) whenever all parameters l_{ij} and u_{ij} are integer. Therefore, the PESP formulation and the CRTP formulation are in fact equivalent.

4.2 On the Constraint Graph Representation for PESP

As for the CRTP, we can represent a PESP instance by a constraint graph G. Throughout this thesis we assume the constraint graph G to be connected. If a constraint graph is not connected, the PESP can be solved for each connected component of G separately, since no constraints are defined between decision variables belonging to different components.

The constraint graph G provides a convenient tool for graphically representing PESP instances. When discussing sets of periodic constraints, we usually depict them in the constraint graph form to graphically clarify the relations between the constraints. This is however not the only reason for using constraint graphs when studying the PESP. Trees and cycles in G play an important part in the analysis of the PESP, as will be shown later in the thesis.

One important insight into the PESP is obtained by defining, for a given vector p, a slightly different graph $G_p = (N, A_p)$ with *two* arcs for every constraint $(i, j) \in A$. One arc is directed from i to j, the other is directed from j to i. We define lengths for the arcs in G_p that depend on the value of the vector p:

$d_{ij}(p_{ij}) = u_{ij} - Tp_{ij}$ The variable length of arc $(i, j) \in A_p$, corresponding to arc $(i, j) \in A$.

$d_{ji}(p_{ij}) = -l_{ij} + Tp_{ij}$ The variable length of arc $(j, i) \in A_p$, corresponding to arc $(i, j) \in A$.

For convenience, define $p_{ji} = p_{ij}$ for all arcs $(i, j) \in A$, so we can always write $d_{ij}(p_{ij})$ for any arc $(i, j) \in A_p$.

Now, consider the problem of finding a shortest path in G_p for the arc lengths $d_{ij}(p_{ij})$ from some source node s to all other nodes in N. The following theorem is well known (see Ahuja et al., 1993, Chapter 5).

Theorem 4.2 (Shortest Path Optimality Conditions). For every node $i \in N$, let π_i denote the length of some directed path from s to i. Then the numbers π_i represent shortest path distances for the arc lengths $d_{ij}(p_{ij})$ if and only if they satisfy the following shortest path optimality conditions:

$$\pi_j \leq \pi_i + d_{ij}(p_{ij}) \qquad \text{for all } (i, j) \in A_p. \tag{4.3}$$

Let us index the shortest path optimality conditions (4.3) on the original constraint set A instead of on A_p:

$$\begin{aligned} \pi_j &\leq \pi_i + d_{ij}(p_{ij}) && \text{for all } (i,j) \in A, \\ \pi_i &\leq \pi_j + d_{ji}(p_{ji}) && \text{for all } (i,j) \in A. \end{aligned}$$

Then, substitute $d_{ij}(p_{ij})$, and use $p_{ij} = p_{ji}$ to obtain

$$\begin{aligned} \pi_j &\leq \pi_i + u_{ij} - Tp_{ij} && \text{for all } (i,j) \in A, \\ \pi_i &\leq \pi_j - l_{ij} + Tp_{ij} && \text{for all } (i,j) \in A. \end{aligned}$$

These two conditions can be taken together and written as

$$l_{ij} \leq \pi_j - \pi_i + Tp_{ij} \leq u_{ij} \text{ for all } (i,j) \in A.$$

So, a PESP instance $G = (N, A, l, u)$ with cycle time T is equivalent to an instance of the shortest path problem in $G_p = (N, A_p)$ with variable arc lengths $d_{ij}(p_{ij})$.

Feasible distance labels can be assigned to the nodes in a graph if and only if there are no negative directed cycles, that is, directed cycles for which the sum of the arc lengths is negative (again, see Ahuja et al., 1993, Chapter 5). The arc lengths in G_p depend on the integer variables p_{ij}. Therefore, feasibility of a PESP instance corresponds to the existence of a vector p such that there are no negative directed cycles in G_p with respect to the arc lengths $d_{ij}(p_{ij})$.

This observation on negative cycles in the graph G_p with respect to the arc lengths $d_{ij}(p_{ij})$ leads to the following important theorem for the PESP by Odijk (1996). We present a proof different from Odijk (1996), which is based on and further clarifies the above presented relation between the PESP and shortest path problems. The theorem and proof use the following notation for cycles in the constraint graph. A cycle C need not be directed, and therefore consists of forward and backward arcs. These are denoted by the sets C^+ for forward arcs, and C^- for backward arcs. For a more detailed description of this notation, and an example, we refer to appendix A.6.

Theorem 4.3 (Odijk, 1996). A PESP instance defined by $G = (N, A, l, u)$ and T is feasible if and only if there exists an integer vector p such that, for each cycle $C \in G$,

$$a_C \leq \sum_{(i,j) \in C^+} p_{ij} - \sum_{(i,j) \in C^-} p_{ij} \leq b_C,$$

where a_C and b_C are defined by

$$a_C = \left\lceil \frac{1}{T} \left(\sum_{(i,j)\in C^+} l_{ij} - \sum_{(i,j)\in C^-} u_{ij} \right) \right\rceil$$

$$b_C = \left\lfloor \frac{1}{T} \left(\sum_{(i,j)\in C^+} u_{ij} - \sum_{(i,j)\in C^-} l_{ij} \right) \right\rfloor$$

Proof. Consider the above described graph $G_p = (N, A_p)$ with arc lengths $d_{ij}(p_{ij})$. Let π_i denote the shortest-path distance label for node i. As was mentioned, feasible values for the variables π_i exist if and only if there are no negative cycles in G_p with respect to the arc lengths $d_{ij}(p_{ij})$. Thus, the PESP instance is feasible if and only if

$$\sum_{(i,j)\in C} d_{ij}(p_{ij}) \geq 0 \text{ for all cycles } C \in G_p. \tag{4.4}$$

First, for a directed cycle in G_p consisting of an arc (i,j) and its counter arc (j,i), we have by definition

$$d_{ij}(p_{ij}) + d_{ji}(p_{ji}) = u_{ij} - Tp_{ij} - l_{ij} + Tp_{ij} = u_{ij} - l_{ij} \geq 0.$$

So these cycles satisfy (4.4).

Next, we rewrite the necessary and sufficient condition (4.4) in terms of cycles in G, and arc values l_{ij} and u_{ij}. Each arc in G induces two arcs in G_p, and therefore each cycle in G induces two directed cycles in G_p. Indexing (4.4) on cycles $C \in G$ gives

$$\begin{cases} \displaystyle\sum_{(i,j)\in C^+} d_{ij}(p_{ij}) + \sum_{(i,j)\in C^-} d_{ji}(p_{ji}) \geq 0 \\ \displaystyle\sum_{(i,j)\in C^+} d_{ji}(p_{ji}) + \sum_{(i,j)\in C^-} d_{ij}(p_{ij}) \geq 0 \end{cases} \text{for all cycles } C \in G.$$

Substituting $d_{ij}(p_{ij})$ yields

$$\begin{cases} \displaystyle\sum_{(i,j)\in C^+} (u_{ij} - Tp_{ij}) + \sum_{(i,j)\in C^-} (-l_{ij} + Tp_{ij}) \geq 0 \\ \displaystyle\sum_{(i,j)\in C^+} (-l_{ij} + Tp_{ij}) + \sum_{(i,j)\in C^-} (u_{ij} - Tp_{ij}) \geq 0 \end{cases} \text{for all cycles } C \in G,$$

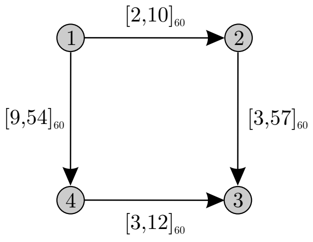


Figure 4.1: An example for the computing parameters a_C and b_C from Theorem 4.3.

which is equivalent to

$$\begin{cases} \sum_{(i,j)\in C^+} p_{ij} - \sum_{(i,j)\in C^-} p_{ij} \leq \frac{1}{T}\left(\sum_{(i,j)\in C^+} u_{ij} - \sum_{(i,j)\in C^-} l_{ij}\right) \\ \sum_{(i,j)\in C^+} p_{ij} - \sum_{(i,j)\in C^-} p_{ij} \geq \frac{1}{T}\left(\sum_{(i,j)\in C^+} l_{ij} - \sum_{(i,j)\in C^-} u_{ij}\right) \end{cases} \quad \text{for all cycles } C \in G.$$

Because of the integrality of the variables p_{ij}, the right hand sides of these expressions can be rounded. Rounding down the first expression gives b_C, and rounding up the second expression gives a_C.

In the above, each step after (4.4) is either an equivalence relation or a substitution. Therefore, both the necessity and the sufficiency claimed in Theorem 4.3 have been proved. ■

The example cycle in Figure 4.1 illustrates Theorem 4.3. Directing the cycle in the figure clockwise, we obtain the following values for the parameters a_C and b_C:

$$a_C = \left\lceil \frac{1}{60}(2+3-12-54) \right\rceil = \left\lceil \frac{-61}{60} \right\rceil = -1$$
$$b_C = \left\lfloor \frac{1}{60}(10+57-3-9) \right\rfloor = \left\lfloor \frac{55}{60} \right\rfloor = 0$$

So, this small example instance is feasible if and only there exists an integer vector $p = (p_{12}, p_{23}, p_{34}, p_{13})$ such that $-1 \leq p_{12} + p_{23} - p_{34} - p_{13} \leq 0$.

The constraint graph provides another valuable insight into the structure of the PESP. To that end, consider a matrix formulation for the PESP, and define

$l \in \mathbb{R}^{|N|}$ The vector of time window lower bounds l_{ij}.
$u \in \mathbb{R}^{|N|}$ The vector of time window upper bounds u_{ij}.

Recall that $v \in \mathbb{R}^{|N|}$ is the vector of variables v_i, and $p \in \mathbb{R}^{|A|}$ the vector of variables p_{ij}. Further, let M be the node-arc incidence matrix of G. That is, M is an $|N| \times |A|$ matrix defined as

$$M(i,a) = \begin{cases} 1 & \text{if } a = (i,j) \text{ for some } j \in N, \\ -1 & \text{if } a = (j,i) \text{ for some } j \in N, \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 4.4. Consider a PESP instance $G = (N, A, l, u)$ and T, with l, u and T integer-valued. If the instance is feasible, then it has an integer solution.

Proof. Consider the matrix formulation of the PESP: find a feasible solution (v, p) to the set of inequalities given by

$$l \leq M^t v + Tp \leq u. \tag{4.5}$$

Suppose that (v^*, p^*) is a solution, with p^* integer, and v^* possibly non-integer. Substitute p^* in (4.5), giving

$$l - Tp^* \leq M^t v \leq u - Tp^*. \tag{4.6}$$

Since M is the node-arc incidence matrix of G, it is totally unimodular, and so M^t is totally unimodular (see Schrijver, 1986, Nemhauser and Wolsey, 1988, for a definition and description of total unimodularity). Moreover, since l, u, p^*, and T are integer, both the leftmost and rightmost term in equation (4.6) are all-integer. Therefore, the polyhedron

$$\mathcal{P}(p^*) = \left\{ v \in \mathbb{R}^{|N|} \mid l - Tp^* \leq M^t v \leq u - Tp^* \right\} \tag{4.7}$$

has integer vertices only. From the existence of solution (v^*, p^*) it follows that $\mathcal{P}(p^*)$ is non-empty, and therefore that the PESP instance has an integer solution. ■

Remark 4.5. Theorem 4.4 and its proof explain Remark 3.1, and also why the v_i variables in the PESP are not required to be integer. The polyhedron $\mathcal{P}(p^*)$ has integer vertices only. Therefore, when using an objective function that is linear or convex, the PESP has an integer valued optimal solution. So, the variables a_n^t and d_n^t (3.16j), and v_i (3.20d), can be relaxed to take real values:

$$\begin{aligned}&a_n^t, d_n^t \in [0, T-1] \text{ for all } t \in \mathcal{T}, n \in \mathcal{N}(t), \text{ and}\\&v_i \in [0, T-1] \text{ for all } i \in N.\end{aligned}$$

The relaxation to real values only holds for these variables. The integer variables p and p_{ij} (3.20e) are still required to be integer.

For the remainder of the thesis, we therefore assume all l_{ij} and u_{ij} to be integer. Moreover, the upper bound for the v_i variables is set to $v_i \leq T-1$. We also refine our assumptions on the time window bounds. The assumption for the time window lower bounds is adjusted to $0 \leq l_{ij} \leq T-1$. The assumption on the time window widths is adjusted to $0 \leq u_{ij} - l_{ij} \leq T-2$, since a time window with width greater than or equal to $T-1$ imposes no periodic constraint.

4.3 Properties of the PESP

This section derives some useful properties and theorems for the PESP. First, it shows how each arc $(i,j) \in A$ can be reversed by adjusting its time window. Next, we explain how any PESP instance can be transformed such that it is written as an integer program containing binary variables p_{ij} only. Then, we discuss the procedure of shifting a cyclic schedule, which is closely related to fixed arrival and departure times of trains. Finally, the modeling of disjoint periodic time windows is explained.

4.3.1 Reversing Constraints

Consider an arc (i,j) in the constraint graph G, with constraint

$$l_{ij} \leq v_j - v_i + Tp_{ij} \leq u_{ij}.$$

Multiplying this constraint by -1 gives

$$-u_{ij} \leq v_i - v_j - Tp_{ij} \leq -l_{ij}.$$

Since the integer variable p_{ij} allows for adding or subtracting an arbitrary integer multiple of T, we may add an integer multiple of T to each of the members in this inequality. Let us add $T\bar{p}, \bar{p} \in \mathbb{Z}$, to each member, where $\bar{p}$ is such that $0 \leq T\bar{p} - u_{ij} \leq T-1$. This gives

$$T\bar{p} - u_{ij} \leq v_i - v_j + T(\bar{p} - p_{ij}) \leq T\bar{p} - l_{ij}.$$

This inequality is in fact the PESP constraint

$$l_{ji} \leq v_i - v_j + Tp_{ji} \leq u_{ji},$$

with

$$\begin{aligned} l_{ji} &= T\bar{p} - u_{ij}, \\ u_{ji} &= T\bar{p} - l_{ij}. \end{aligned}$$

Note that l_{ji} and u_{ji} satisfy the time window assumptions $0 \leq l_{ji} \leq T-1$ and $0 \leq u_{ji} - l_{ji} \leq T-2$. Moreover, we have the relation $p_{ji} = \bar{p} - p_{ij}$. We do not need to enforce this relation however, since a PESP constraint only requires some integer variable, so we could use a new integer variable p_{ji}. The above is summarized in the following lemma.

Lemma 4.6. Any arc (i, j) in the PESP constraint graph G with time window $[l_{ij}, u_{ij}]$ can be reversed to an arc (j, i) with time window $[l_{ji}, u_{ji}]$ defined by

$$\begin{aligned} l_{ji} &= T\bar{p} - u_{ij}, \\ u_{ji} &= T\bar{p} - l_{ij}, \end{aligned}$$

where $\bar{p} \in \mathbb{Z}$ is such that $0 \leq l_{ji} \leq T-1$. Moreover, $p_{ji} = \bar{p} - p_{ij}$.

4.3.2 Binary Variables

If desired, any PESP instance with $p \in \mathbb{Z}^{|A|}$ can be transformed into a formulation where the variables p_{ij} may take only *binary* values, that is, $p \in \{0,1\}^{|A|}$. This transformation follows from considering the bounds on the variables v_i. Since $0 \leq v_i \leq T-1$, we have $-T+1 \leq v_j - v_i \leq T-1$. Moreover, if the time window $[l_{ij}, u_{ij}]$ is strictly contained in $[0, T-1]$, then the only periodic shift that the corresponding variable p_{ij} can achieve, is the shift from the time window $[l_{ij}, u_{ij}]$ into the time window $[l_{ij} - T, u_{ij} - T]$, by setting $p_{ij} = 1$. This is illustrated in Figure 4.2. Any other non-zero value for p_{ij} is useless because of the bounds on v_i and v_j. We obtain the following lemma:

Lemma 4.7. If $0 \leq v_i \leq T-1$ for all $i \in N$, and $0 \leq u_{ij} \leq T-1$, then $p_{ij} \in \{0,1\}$, for all $(i,j) \in A$.

Note that, by assumption, we have $0 \leq l_{ij} \leq T-1$ and $0 \leq u_{ij} - l_{ij} \leq T-2$.

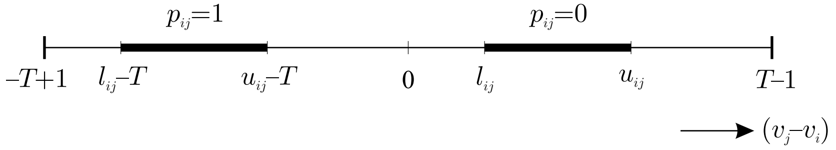


Figure 4.2: Possible values for p_{ij} when $u_{ij} \leq T - 1$.

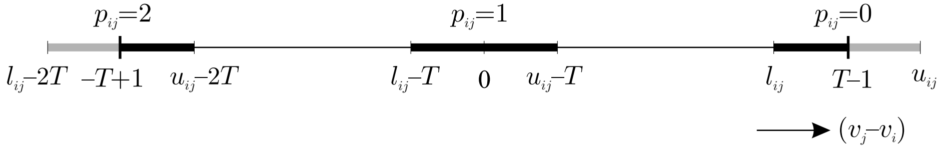


Figure 4.3: Possible values for p_{ij} when $u_{ij} > T - 1$.

The next section describes some aspects of sequencing in a cyclic environment. For now, we want to point at a sequencing interpretation of the integer variable p_{ij} when $u_{ij} \leq T-1$. Consider the linear time axis $[0, T-1]$, for the moment forgetting that the time axis is cyclic for the PESP. On the linear axis $[0, T-1]$, if $v_j \geq v_i$, then $p_{ij} = 0$. Else, if $v_i > v_j$, then $p_{ij} = 1$. So, in a sense, the variable p_{ij} can be interpreted as representing the sequence in which the events i and j take place on the linear axis $[0, T-1]$. Note that i and j may take place concurrently if $p_{ij} = 0$.

If there exist time windows $[l_{ij}, u_{ij}]$ with $u_{ij} \geq T$, then p_{ij} may take three values, namely $p_{ij} \in \{0, 1, 2\}$. These three values correspond to the following shift of the time window $[l_{ij}, u_{ij}]$, as is illustrated in Figure 4.3:

$$p_{ij} = \begin{cases} 0 & \text{for the non-shifted window } [l_{ij}, T-1], \\ 1 & \text{for the shifted window } [l_{ij} - T, u_{ij} - T], \\ 2 & \text{for the shifted window } [-T+1, u_{ij} - 2T]. \end{cases}$$

In this case, a sequence interpretation of p_{ij} does not exist, since the value $p_{ij} = 1$ can correspond to both $v_j \geq v_i$ and $v_j \leq v_i$.

The above shows that, for transforming a PESP instance into a binary program, all that is needed is to ensure that $0 \leq u_{ij} \leq T - 1$ for all $(i, j) \in A$. The following splitting procedure deals with arcs that violate this condition.

Step 1. For any arc (i, j) with $u_{ij} \geq T$, split the arc into two arcs (i, k) and (k, j) by introducing a new node k.

Step 2. For the arc (i, k), set $l_{ik} := l_{ij}, u_{ik} := l_{ij}$.

Step 3. For the arc (k, j), set $l_{kj} := 0, u_{kj} := u_{ij} - l_{ij}$.

After splitting an arc, the original constraint between i and j is still implied. This can be seen from the constraints associated with the two new arcs (i, k) and (k, j), when substituting the values for their time windows:

$$l_{ij} \leq v_j - v_k + Tp_{kj} \leq l_{ij}, \tag{4.8}$$

$$0 \leq v_k - v_i + Tp_{ik} \leq u_{ij} - l_{ij}. \tag{4.9}$$

Adding these two constraints gives $l_{ij} \leq v_j - v_i + T(p_{kj} + p_{ik}) \leq u_{ij}$, where the term $(p_{kj} + p_{ik})$ is an integer variable. And the two arcs (i, k) and (k, j) each satisfy $0 \leq u \leq T - 1$. The splitting procedure, together with Lemma 4.7, implies the following lemma.

Lemma 4.8. The PESP can be transformed such that $p_{ij} \in \{0, 1\}$ for all $(i, j) \in A$.

The price for writing the PESP with binary variables p_{ij} is an increase in the number of constraints by the splitting procedure, and therefore an increase in the number of integer variables.

4.3.3 Shifting a Cyclic Schedule

Suppose that we are given a feasible solution (v, p) to a PESP instance $G = (N, A, l, u)$ with cycle time T. Then, all event times v_i can be shifted by an arbitrary integer number $\beta \in \{0, \ldots, T - 1\}$, to obtain a new feasible schedule $(\bar{v}, \bar{p})$. With shifting a schedule by β, we mean the following:

$$\bar{v}_i := (v_i - \beta) \text{ modulo } T$$

$$\bar{p}_{ij} := \begin{cases} 0 & \text{if } \bar{v}_j \geq \bar{v}_i \\ 1 & \text{if } \bar{v}_j < \bar{v}_i, \end{cases}$$

assuming that $u_{ij} \leq T - 1$. It is easy to see that the new schedule $(\bar{v}, \bar{p})$ is feasible. For each constraint, both $\bar{v}_j$ and $\bar{v}_i$ have been shifted by the same amount β, so $\bar{v}_j - \bar{v}_i = v_j - v_i$. Further, the variables $\bar{p}_{ij}$ have been adjusted so as to incorporate a change in the sequence of i and j on the linear time axis $[0, T - 1]$.

If $u_{ij} \geq T$, the argument is similar, though more complicated. Finally, note that this shifting of a cyclic schedule allows for modeling fixed departure and arrival times of trains using PESP constraints only (see Section 3.4.4).

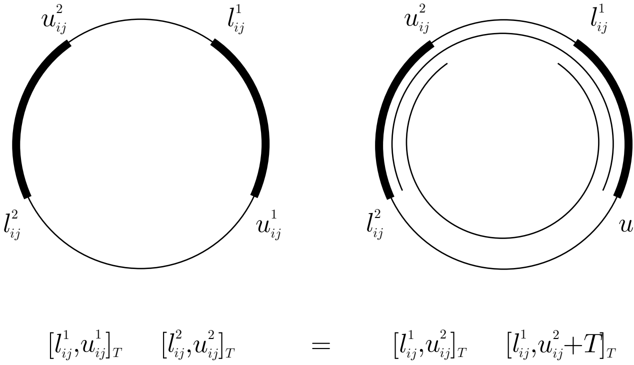

u^2_{ij}
l^1_{ij}
l^2_{ij}
u^1_{ij}
u^2_{ij}
l^1_{ij}
l^2_{ij}
u^1_{ij}
$[l^1_{ij},u^1_{ij}]_T$ $[l^2_{ij},u^2_{ij}]_T$ = $[l^1_{ij},u^2_{ij}]_T$ $[l^1_{ij},u^2_{ij}+T]_T$

Figure 4.4: Relation between the union and intersection of periodic time windows.

4.3.4 Disjoint Periodic Time Windows

Periodic constraints can model a choice between multiple disjoint time windows for a given event pair (i, j). Suppose that two events i and j are related by a constraint, and that their event time difference should lie *either* in the time window $[l^1_{ij}, u^1_{ij}]$, *or* in the time window $[l^2_{ij}, u^2_{ij}]$. So, for a certain event pair, we have a choice between two time windows. Figure 4.4 shows that these disjoint time windows are equivalent to the periodic constraints:

$$\begin{aligned} v_j - v_i &\in [l^1_{ij}, u^2_{ij}]_T, \\ v_j - v_i &\in [l^2_{ij}, u^1_{ij} + T]_T. \end{aligned}$$

This idea generalizes to the following lemma.

Lemma 4.9. Suppose that for some arc (i, j) we want to impose the constraint

$$v_j - v_i \in [l^1_{ij}, u^1_{ij}]_T \cup [l^2_{ij}, u^2_{ij}]_T \cup \ldots \cup [l^k_{ij}, u^k_{ij}]_T,$$

where the k time windows are disjoint and ordered:

$$0 \leq l^1_{ij} \leq u^1_{ij} < l^2_{ij} \leq u^2_{ij} < \cdots < l^k_{ij} \leq u^k_{ij} < l^1_{ij} + T.$$

Then this union of k time windows is equivalent to the intersection of k periodic time windows given by the periodic constraints

$$\begin{aligned} v_j - v_i &\in [l_{ij}^1, u_{ij}^k]_T, \\ v_j - v_i &\in [l_{ij}^2, u_{ij}^1 + T]_T, \\ &\vdots \\ v_j - v_i &\in [l_{ij}^k, u_{ij}^{k-1} + T]_T. \end{aligned}$$

4.4 Cyclic Sequencing

On a cyclic time axis, both event i takes place after event j, and event j takes place after event i, unless i and j take place at the same time. This means that there exists no sequence for two events i and j. Consequently, sequences in a cyclic schedule are defined on at least three events. These three events i, j, and k may occur in one of the two essentially different orders $i \rightarrow j \rightarrow k$ and $i \rightarrow k \rightarrow j$. Any other cyclic sequence of the three events is equivalent to one of these two. For example, the sequence $k \rightarrow j \rightarrow i$ is equivalent to $i \rightarrow k \rightarrow j$. The following definitions formalize the concept of a cyclic sequence.

Definition 4.10 (Cyclic Sequence). The events $1, \dots, k$ are said to be cyclically sequenced in the order $1 \rightarrow \cdots \rightarrow k$ if

$$0 \leq (v_2 - v_1) \text{ modulo } T \leq \cdots \leq (v_k - v_1) \text{ modulo } T.$$

For some of the results we derive below, it is necessary to prevent the case in which two or more events take place concurrently. Therefore, we define a proper cyclic sequence as follows.

Definition 4.11 (Proper Cyclic Sequence). The events $1, \dots, k$ are said to be proper cyclically sequenced in the order $1 \rightarrow \cdots \rightarrow k$ if

$$0 < (v_2 - v_1) \text{ modulo } T < \cdots < (v_k - v_1) \text{ modulo } T.$$

A cyclic sequence of the events $1, \dots, k$ is a proper cyclic sequence if the events are restricted by some periodic constraints, such that neither 1 and k, nor any of the subsequent events i and $i+1, i = 1, \dots, k-1$, can take place concurrently.

Below, we present four lemmata that characterize the relationship between the variables p_{ij} and the sequence in which events take place. For these lemmata, we assume that all involved constraints have $u_{ij} \leq T-1$, and thus $p_{ij} \in \{0,1\}$. If this is not the case, one can use Lemma 4.8 to transform any constraint with $u_{ij} \geq T$ into two constraints with $u_{ij} \leq T-1$.

Lemma 4.12. Consider a directed cycle $C = (1, 2, \ldots, k, 1)$ in the constraint graph G, and a schedule (v, p). If the events $1, \ldots, k$ take place in the proper cyclic sequence $1 \to \cdots \to k$, then $\sum_{(i,j)\in C} p_{ij} = 1$.

Proof. The inter-event time $v_j - v_i + Tp_{ij}$ expresses the number of minutes that j takes place after i. If the events $1, \ldots, k$ are proper cyclically sequenced in the order $1 \to \cdots \to k$, then none of the inter-event times equals zero. Thus, going from event 1 to 2 to $\cdots$ to k, and back to 1, the sum of the inter-event times must equal T. In other words, $\sum_{(i,j)\in C}(v_j - v_i + Tp_{ij}) = T$. In this summation, the variables v_i cancel out, giving $\sum_{(i,j)\in C} p_{ij} = 1$. ■

Lemma 4.13. Consider a directed cycle $C = (1, 2, \ldots, k, 1)$ in the constraint graph G, and a schedule (v, p). If $\sum_{(i,j)\in C} p_{ij} = 1$, then the events $1, \ldots, k$ take place in the cyclic sequence $1 \to \cdots \to k$.

Proof. $\sum_{(i,j)\in C} p_{ij} = 1$ means that all but one of the variables p_{ij} in C are zero, because all $p_{ij} \in \{0,1\}$. Let (i^*, j^*) be the single constraint with $p_{i^*j^*} = 1$. Since $p_{ij} = 0$ for all other constraints (i,j), we have

$$v_{j^*} \leq v_{j^*+1} \leq \cdots \leq v_k \leq v_1 \leq \cdots \leq v_{i^*-1} \leq v_{i^*}.$$

So, the events are cyclically sequenced as $j^* \to \cdots k \to 1 \to \cdots \to i^*$, which is equivalent to $1 \to \cdots \to k$. ■

Note that Lemma 4.12 only holds for a *proper* cyclic sequence, that is, the reverse of Lemma 4.13 does not hold. A regular cyclic sequence in which *all* events take place concurrently would yield $\sum_{(i,j)\in C} p_{ij} = 0$. If, however, at least one arc a in the cycle C has $l_a > 0$, then Lemma 4.12 also holds for a regular cyclic sequence, and in that case Lemma 4.13 can be stated as an if and only if relation.

Next, we consider cycles that may contain both forward and backwards arcs.

Lemma 4.14. Consider a not necessarily directed cycle $C = (1, 2, \ldots, k, 1)$ in the constraint graph G, and a schedule (v, p). If the events $1, \ldots, k$ take place in the proper cyclic sequence $1 \to \cdots \to k$, then

$$\sum_{(i,j)\in C^+} p_{ij} - \sum_{(i,j)\in C^-} p_{ij} = 1 - |C^-|.$$

Proof. Using Lemma 4.6, we reverse each backward arc $(i, j) \in C^-$. Since $p_{ij} \in \{0, 1\}$, this gives $\bar{p} = 1$, and thus $p_{ji} = 1 - p_{ij}$. Moreover, for the reversed arc (j, i) it still holds that $u_{ji} \leq T$, since we are considering a proper cyclic sequence. For the now directed cycle C, we apply Lemma 4.12. This gives that, if the proper cyclic sequence $1 \to \cdots \to k$ takes place, then

$$\sum_{(i,j)\in C^+} p_{ij} + \sum_{(i,j)\in C^-} p_{ji} = 1.$$

Substituting $p_{ji} = 1 - p_{ij}$ for the reversed arcs, we obtain

$$\sum_{(i,j)\in C^+} p_{ij} + \sum_{(i,j)\in C^-} (1 - p_{ji}) = 1,$$

which concludes the proof. ■

Lemma 4.15. Consider a not necessarily directed cycle $C = (1, 2, \dots, k, 1)$ in the constraint graph G, and a schedule (v, p). If

$$\sum_{(i,j)\in C^+} p_{ij} - \sum_{(i,j)\in C^-} p_{ij} = 1 - |C^-|,$$

then the events $1, \dots, k$ take place in the cyclic sequence $1 \to \cdots \to k$.

Proof. As in the proof of the previous lemma, we reverse each backward arc $(i, j) \in C^-$. Then, apply Lemma 4.13 to the directed cycle C. ■

Again, we need a proper cyclic sequence in Lemma 4.14, since the reverse of Lemma 4.15 does not hold. Consider, for example, a cycle C consisting of alternatingly forward and backward arcs. Suppose that for each forward arc (i, j), and for the subsequent backward arc (k, j), we have $v_i \neq v_j, v_j = v_k$. Then, for each backward arc (i, j) in this cycle C, the inter-event time difference $v_j - v_i = 0$, so $p_{kj} = 0$. In that case, the cyclic sequence $1 \to \cdots \to k$ may yield $\sum_{(i,j)\in C^+} p_{ij} - \sum_{(i,j)\in C^-} p_{ij} > 1 - |C^-|$.

However, if the constraints defined on the events $1, \dots, k$ are such that none of the subsequent events in C can take place concurrently, then any cyclic sequence is a proper cyclic sequence, and therefore the reverse of the lemmata does hold. The constraints that prevent the events from taking place concurrently need not be the constraints in the cycle C. The following corollary summarizes the above:

Corollary 4.16. Consider a not necessarily directed cycle $C = (1, 2, \dots, k, 1)$ in the constraint graph G, and a schedule (v, p). Suppose that some set of constraints

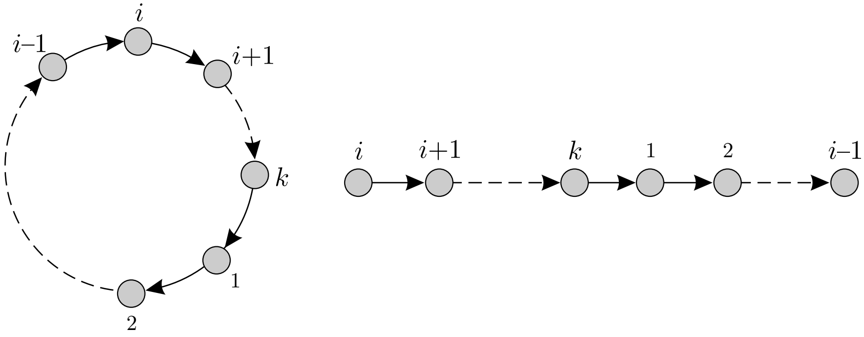


Figure 4.5: Cyclic versus linear sequencing.

on the events $1, \dots, k$ prevents any subsequent pair $(i, i+1) \in C$ from taking place concurrently. Then the events $1, \dots, k$ take place in the cyclic sequence $1 \to \cdots \to k$ if and only if

$$\sum_{(i,j)\in C^+} p_{ij} - \sum_{(i,j)\in C^-} p_{ij} = 1 - |C^-|.$$

This corollary is the result that we mainly use in the remainder of the thesis.

As was noticed by Nachtigall (1999), some authors loosely say that fixing the variables p_{ij} is equivalent to fixing the order in which events take place. Lemma 4.15 shows what relations exactly exist between the variables p_{ij} and sequences of events. Fixing all variables p_{ij} clearly fixes the sequence of events. Fixing the variables p_{ij} for some subset of the arcs only fixes the sequences of events that appear in cycles induced by those arcs. Conversely, Lemma 4.14 shows that fixing a sequence of some events only fixes sums of variables p_{ij} along the cycles through these events.

Cyclic sequencing is not as different from linear sequencing as it may appear at first sight. This was already hinted at in Section 4.3.2, where we explained that a variable p_{ij} can be seen as indicating which of the events i, j takes place first on the linear time axis $[0, T-1]$. Consider a set of events $1, \dots, k$, and suppose they are cyclically sequenced as $S = 1 \to \cdots \to k$, see the left part of Figure 4.5. Because of its cyclic nature, S is equivalent to $2 \to \cdots \to k \to 1$, or to $3 \to \cdots \to k \to 1 \to 2$, etc. Next, choose a reference point on the cyclic time axis, cut the axis there, and then consider the sequence as a linear one, starting at the reference point. This is illustrated in the right part of Figure 4.5, where the reference point chosen is i, and, on the linear time axis, we see the linear sequence $i \to i{+}1 \to \cdots \to 1 \to k \to \cdots \to i{-}1$. This linear sequence is equivalent to the cyclic sequence S. This analogy between cyclic and linear sequencing gives the following lemma.

Lemma 4.17. The cyclic sequencing of k events is equivalent to the linear sequencing of $k-1$ events. In particular, one can linearly sequence the events $2, \ldots, k$ as the sequence S, which gives the cyclic sequence $1 \rightarrow S$.

4.5 Complexity of the PESP

Several authors have proven that the PESP is NP-complete, by reductions from the Hamiltonian Circuit Problem (Serafini and Ukovich, 1989a, Nachtigall, 1994), Graph K-Colorability (Odijk, 1997), and the Linear Ordering Problem (Liebchen and Peeters, 2002b). For completeness, we include the proof by Odijk (1997). This proof reduces the question of PESP feasibility to Graph K-Colorability, which is known to be NP-complete for fixed $K \geq 3$ (see Garey and Johnson, 1979).

Definition 4.18 (Graph K-Colorability). Given an undirected graph $U = (N, E)$ and an integer $K \leq |V|$, is it possible to assign to each node $i \in N$ a color $c(i) \in \{1, \ldots, K\}$, such that $c(i) \neq c(j)$ if i and j are adjacent?

Theorem 4.19. PESP is NP-complete for fixed $T \geq 3$.

***Proof*.** Clearly, PESP is in NP. Given an instance $U = (N, E)$ and K of Graph K-Colorability, construct an instance of the PESP as follows. Set the cycle time T equal to K. Take N as the node set for the PESP instance, and construct the arc set A by assigning an arbitrary direction to each edge in E. For each arc $(i, j) \in A$, define a time window $[1, K-1]_K$. This gives the PESP instance $G = (V, A, 1, K-1)$ with cycle time K. If this instance of the PESP is feasible, then by Theorem 4.4 it has an integer feasible solution. Moreover, because of the time windows $[1, K-1]_K$, such an integer feasible solution to the PESP assigns different values $v_i \neq v_j$ to any adjacent pair of events i, j. Conversely, any feasible coloring of U assigns colors $c(i) \neq c(j)$ to the nodes of an arc $(i, j) \in A$, for which it thus holds that $c(j) - c(i) \in [1, K-1]_K$. Finally, the transformation from the Graph K-Colorability instance to the PESP instance can be done in polynomial time. ■

In the case $T = 2$, we have the following situation. Because of the assumptions on the time window bounds, the only time windows that are allowed for $T = 2$ are $[0, 0]_2$ and $[1, 1]_2$. Let A_0 be the set of arcs with time window $[0, 0]_2$, and let A_1 be the set of arcs with time window $[1, 1]_2$. Further, each variable v_i has domain $v_i \in \{0, 1\}$. This means that arcs $(i, j) \in A_0$ imply the restriction $v_i = v_j$, and that arcs $(i,j) \in A_1$ imply the restriction $v_i \neq v_j$. Section 4.3.3 showed that a cyclic schedule can always be shifted, so we can fix the time instant v_{i^*} of some event i^* to an arbitrary value in its domain. So, for $T = 2$, we can fix $v_{i^*} = 0$ for some event i^*. Next, this solution can be propagated through the constraint graph G, setting $v_i = v_j$ for all $(i,j) \in A_0$, and $v_i \neq v_j$ for all $(i,j) \in A_1$. This procedure clearly either yields a feasible solution, or proves that no feasible solution exists. Moreover, it takes $O(m)$ time.

4.6 Literature Review

All work on the PESP that we know of, other than the original papers (Serafini and Ukovich, 1989a,b, Gertsbakh and Serafini, 1991), is done in the area of cyclic railway timetabling. Therefore, most references below were already mentioned in Section 1.4. Here, we discuss them in the light of this chapter. For an in-depth overview of PESP algorithms, we refer to Lindner (2000, Chapter 3).

Serafini and Ukovich (1989a) introduced the PESP. Their main interest is in finding feasible periodic schedules, so an objective function is not taken into account. They prove that the general PESP is NP-complete, and propose a Branch&Bound procedure for finding feasible solutions. They also present some applications of the PESP in job shop scheduling, transportation scheduling, and traffic light scheduling.

Voorhoeve (1993) first considered the PESP for cyclic railway timetabling problems. He developed an algorithm based on constraint propagation and backtracking, for finding a feasible cyclic railway timetable.

Schrijver and Steenbeek (1993, 1994) developed a constraint programming based algorithm, called CADANS, for the problem of finding a feasible timetable. The algorithm first uses a preprocessing phase to eliminate redundant variables and constraints, and to tighten constraints. Next, every variable v_i is assigned a domain that initially consists of the set $[0, \ldots, T-1]$. The algorithm then enters an iterative procedure. In each iteration, the value of some v_i is fixed to one of the values in its domain, and the consequences of this choice are propagated through the constraints. This possibly decreases the size of the domains of some other variables. Whenever an empty domain is found, a sophisticated backtracking procedure is started that resets some of the previously made choices. This backtracking procedure is the key to the generally very good performance of the algorithm. Most real-life railway timetabling instances from NSR and Railned can be solved in several minutes up to two to three hours of computation time. Still, some hard instances require very large computation times. Furthermore, Schrijver and Steenbeek (1994) implemented a procedure for post-optimizing an obtained feasible timetable. This procedure considers the neighborhood consisting of all timetables with vector p equal to that of the obtained timetable, and uses Linear Programming to find an optimal timetable in this neighborhood. In a later paper, Schrijver (1998) briefly discusses some other properties of cyclic railway timetables.

Odijk (1996, 1997) developed a cutting plane algorithm to solve the feasibility problem. The algorithm iteratively cuts off solutions violating the inequalities in Theorem 4.3. His objective is to quickly generate a set of timetables in order to evaluate candidate infrastructure projects within and around railway stations. He obtains good results, but only for rather small instances consisting of a railway station and the small railway network surrounding the station.

Based on the same cuts, Hurkens (1996) uses a Branch&Cut approach to solve the PESP formulation for larger railway timetabling problems. He concludes that it is hard to make the method work fast. Furthermore, very large scale LP problems have to be solved to obtain good bounds for large problem instances, and this also causes problems.

Hassin (1996) describes the Network Synchronization Problem (NSP), which provides an optimization formulation for the PESP. The NSP does not take into account constraints as hard rules. Instead, forbidden time window values are highly penalized. Hassin reports that penalizing very wide time windows causes problems for his algorithm, and the CRTP typically has many wide safety constraints (see Section 3.4.8).

Lindner (2000) studied the PESP as a sub-problem for constructing cost-optimal railway timetables. He developed a modification of the Serafini and Ukovich (1989a) algorithm that works much faster. Further, he investigated the polyhedral structure of the PESP, and proposes a Branch&Cut algorithm. For some real-life instances from Netherlands and German railways, his methods work quite fast, with computation times of up to 300 seconds. However, for other real-life instances, a feasible solution can not be found after ten hours of computation time.

Chapter 5

A Cycle Periodicity Formulation for the CRTP

The previous two chapters considered the CRTP for cyclic railway timetabling, and the PESP which provides a general framework for the CRTP. This chapter describes a transformed formulation for the CRTP and the PESP, which was introduced by Nachtigall (1999). The transformed formulation views the PESP in terms of process times, and cycles in the constraint graph $G = (N, A, l, u)$. This formulation ensures the cyclic nature of the timetable through so-called cycle periodicity constraints, and therefore we name it the Cycle Periodicity Formulation (CPF).

In describing the CPF, we first give a brief overview of the graph theoretic concepts of potentials and tensions, and extend these concepts to the periodic case. Next, using the idea of periodic potentials and tensions, we describe the transformation from the PESP to the CPF. Cycles in the constraint graph, and cycle bases of the constraint graph are an important aspect of the CPF. Therefore, the remainder of the chapter is devoted to several properties of cycles and cycle bases associated with the constraint graph. As such, cycles and paths in the constraint graph G play an important part in this chapter. Our notation for cycles and paths is described in Appendix A.6.

5.1 Periodic Potentials and Tensions

This section first reviews the graph theoretic concepts of potentials and tensions. Next, we extend these concepts to the periodic case, and introduce periodic tensions, which form the building blocks for formulating the cycle periodicity formulation in the next section.

5.1.1 Classical Potentials and Tensions

Consider a directed graph $G = (N, A)$, and a set of values π_i defined by the function $\pi : N \to \mathbb{R}$. Such a set of node values is called a *potential* (the terminology arises from electrical networks). Let a set of arc values y_a defined by the function $y : A \to \mathbb{R}$ correspond to some potential π as follows

$$y_a = \pi_j - \pi_i \text{ for all } a = (i, j).$$

Such a set of arc values is called a *tension*. The following result characterizes a tension without relating it to a potential (see Rockafellar, 1984, Bollobás, 1998).

Theorem 5.1. Given a directed graph $G = (N, A)$, a function $y : A \to \mathbb{R}$ is a tension if and only if

$$\sum_{a \in C^+} y_a - \sum_{a \in C^-} y_a = 0 \text{ for all cycles } C \in G.$$

5.1.2 Periodic Potentials and Tensions

One can also consider a periodic variant of potentials and tensions. We define periodic potentials and tensions as follows.

Definition 5.2. A function $v : N \to \mathbb{R}$ is a *periodic potential with period T* if $0 \leq v_i \leq T - 1$ for all $i \in N$.

Definition 5.3. A function $x : A \to \mathbb{R}$ is a *periodic tension with period T* if $x \geq 0$, and, for a periodic potential v with period T and some integer vector $p \in \mathbb{Z}^{|A|}$, it holds that

$$x_a = v_j - v_i + Tp_a \text{ for all } a = (i, j) \in A. \tag{5.1}$$

So, in a solution (v, p) to the PESP, v is a periodic potential. Moreover, we can associate a periodic tension x with the solution (v, p).

As for the non-periodic case, the theorem below characterizes periodic tensions through cycles in the graph G, rather than through periodic potentials. First, we define the concept of cycle periodicity.

Definition 5.4. For a cycle C, a set of arc values $x_a, a \in C$, satisfies the *cycle periodicity property* for a cycle time T if, for some integer variable q_C,

$$\sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = Tq_C. \tag{5.2}$$

We refer to constraint (5.2) as the *cycle periodicity constraint* for cycle C, and to q_C as the *cycle periodicity integer variable* for cycle C.

Theorem 5.5. Given a directed graph $G = (N, A)$ and a period T, a set of non-negative values $x_a, a \in A$, is a periodic tension if and only if, for each cycle $C \in G$, it satisfies the cycle periodicity property for cycle time T:

$$\sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = Tq_C.$$

Proof. When taking the sum of periodic tension variables $x_a = v_j - v_i + Tp_a$ along a cycle C, the variables v_i cancel out:

$$\sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = \sum_{a \in C^+} Tp_a - \sum_{a \in C^-} Tp_a.$$

The right hand side is clearly an integer multiple of T. In fact, we have $q_C = \sum_{a \in C^+} p_a - \sum_{a \in C^-} p_a$.

Now suppose we have a function $x : A \to \mathbb{R}, x \geq 0$, satisfying the cycle periodicity property (5.2) for all cycles $C \in G$. To show that x is a periodic tension, we construct a corresponding periodic potential v for which there exists an integer vector p fulfilling (5.1). To that end, let H be a spanning tree of G. Choose an arbitrary node $s \in N$, and set $v_s = 0$. For all other nodes $i \in N$, take the undirected path P_{si} in H from s to i, and set

$$v_i = \Big(\sum_{a \in P_{si}^+} x_a - \sum_{a \in P_{si}^-} x_a \Big) \text{ modulo } T.$$

This clearly defines a periodic potential $v : N \to \mathbb{R}$.

For an arc $(i, j) \in H$, the difference of the periodic potentials of j and i is given by

$$v_j - v_i = \Big(\sum_{a \in P_{sj}^+} x_a - \sum_{a \in P_{sj}^-} x_a \Big) - \Big(\sum_{a \in P_{si}^+} x_a - \sum_{a \in P_{si}^-} x_a \Big) \text{ modulo } T = x_{ij} \text{ modulo } T.$$

This means that there exists an integer p_{ij} fulfilling condition (5.1) for each tree arc (i, j).

Next, consider a non-tree arc (i, j). Adding (i, j) to the tree H creates a cycle C. If C contains s, then C consists of $P_{si}, (i, j),$ and P_{sj}. Directing C as the arc (i, j), the cycle periodicity constraint (5.2) for C reads

$$\Big(\sum_{a \in P_{si}^+} x_a - \sum_{a \in P_{si}^-} x_a\Big) + x_{ij} - \Big(\sum_{a \in P_{sj}^+} x_a - \sum_{a \in P_{sj}^-} x_a\Big) = Tq_C.$$

Using the above defined values for v_j and v_i, this expression is rewritten to

$$v_j - v_i = x_{ij} - Tq_C \text{ modulo } T,$$

with q_C some integer variable. So in this case, for each non-tree arc (i, j), there exists an integer p_{ij} fulfilling condition (5.1). For the case where C does not contain s, the common part of the paths P_{si} and P_{sj} cancels out in the above cycle periodicity constraint for C, and we obtain the same result. It follows that x is a periodic tension. ■

5.2 The Cycle Periodicity Formulation

From the previous section, one sees a strong relation between the PESP and periodic potentials. In this section, we use Theorem 5.5 to transform the PESP into a model based on periodic tensions x_a, cycle periodicity variables q_C, and the cycle periodicity constraints (5.2). With the definition $x_{ij} = v_j - v_i + Tp_{ij}$, the transformed model can be interpreted as being stated in terms of process times, that is, time differences between pairs of events. In contrast, the CRTP and PESP are both stated in terms of event times, that is, the time instants at which events take place. At the end of this section, we formally state a mathematical program for the transformed model. Below, we first derive some lemmata that are needed to transform the model.

The following lemma shows the relation between the variables p_{ij} in the PESP, and the variables q_C.

Lemma 5.6. The cycle periodicity integer variables q_C and the PESP integer variables p_a are related by

$$q_C = \sum_{a \in C^+} p_a - \sum_{a \in C^-} p_a.$$

***Proof*.** See the proof of Theorem 5.5. ■

Theorem 4.3 introduced bounds a_C and b_C on the directed sum of p_{ij} variables along a cycle. In fact, these bounds are also valid for the cycle integer variable q_C, as the following lemma clarifies.

Lemma 5.7. The cycle periodicity integer variables q_C are bounded by
$a_C \leq q_C \leq b_C$.

Proof. Substitute the lower bounds and upper bounds $l_a \leq x_a \leq u_a$ in (5.2), and round as in the proof of Theorem 4.3. ■

Lemma 5.8. A cycle periodicity integer vector q is feasible if and only if $a_C \leq q_C \leq b_C$ for all cycles $C \in G$.

Proof. This follows from Lemma 5.6 in combination with Theorem 4.3. ■

The Cycle Periodicity Formulation (CPF) for the PESP is described by

$$\begin{array}{llll} \textbf{CPF:} & \text{Minimize} & F(x) & & \\ & \text{subject to} & \displaystyle\sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = T q_C & \text{for all } C \in G & (5.3a) \\ & & l_a \leq x_a \leq u_a & \text{for all } a \in A & (5.3b) \\ & & a_C \leq q_C \leq b_C & \text{for all } C \in G & (5.3c) \\ & & x_a \in \mathbb{R} & \text{for all } a \in A & (5.3d) \\ & & q_C \in \mathbb{Z} & \text{for all } C \in G & (5.3e) \end{array}$$

Here, we use the general objective function $F(x)$, where x is the vector containing all variables $x_a, a \in A$. We described the specific instances $F_t(x), F_r(x)$, and $F_v(x)$ of $F(x)$ in Section 3.7. The constraints in the model consist of constraints (5.3a), requiring periodicity for the cycles in the constraint graph, together with the time window constraints (5.3b). The time window constraints themselves are no longer periodic. Constraints (5.3c) represent the bounds from Lemma 5.7.

Suppose that (x, q) is a solution to (5.3). From the proof of Theorem 5.5, the following procedure can be deduced for constructing a periodic potential v from the periodic tension x.

Step 1. Construct a spanning tree H of G.

Step 2. Choose an arbitrary node $s \in N$, and set $v_s = 0$.

Step 3. For each other node $i \in N$, take the path P_{si} in H, and set

$$v_i = \sum_{(i,j)\in P_{si}^+} x_{ij} - \sum_{(i,j)\in P_{si}^-} x_{ij}$$

A corresponding integer vector p is given by $p_{ij} = (x_{ij} - v_j + v_i)/T$. Conversely, given a solution (v, p) to (4.2), a periodic tension x is straightforwardly computed by $x_{ij} = v_j - v_i + Tp_{ij}$. The corresponding cycle periodicity integer variables q_C are computed by applying Lemma 5.6.

A drawback of the CPF is that it contains constraints (5.3a) requiring the cycle periodicity property (5.2) for every cycle in G, since G may contain exponentially many cycles. Moreover, the cycle based model has an integer variable q_C for every cycle C, so the CPF may have an exponential number of integer variables. The PESP formulation (4.2), on the other hand, has $|A|$ periodic time window constraints and $|A|$ integer variables p_{ij}.

However, the next section presents a certain condition under which it suffices to state the cycle periodicity constraints (5.3a) only for the cycles in a basis B of the cycle space of G. The cycle space of G has dimension $c = |A| - |N| + 1$, which is much smaller than the possibly exponential number of all cycles in G. That means that the CPF has less constraints and integer variables than the PESP.

5.3 Cycle Bases for Formulating the CPF

This section first briefly describes cycle spaces and cycle bases of directed graphs. Next, we present the condition under which it suffices to require the cycle periodicity constraints (5.3a) only for the cycles in a cycle basis. Finally, we show that the CPF may return an infeasible timetable when using a cycle basis that does not satisfy this condition.

5.3.1 Cycle Bases for Directed Graphs

We briefly describe cycle bases of directed graphs. Appendix B contains a more detailed description of cycle bases of both undirected and directed graphs, and includes references to existing literature on the subject. Moreover, it illustrates these concepts through some examples.

We do not require cycles in directed graphs to be directed. So, a cycle in a directed graph $G = (N, A)$ may contain forward and backward arcs. Therefore, a cycle C in a directed graph is encoded by a $\{0, \pm 1\}$ cycle vector γ_C. Choosing an arbitrary direction for the cycle, γ_C is defined as

$$\gamma_{Ca} = \begin{cases} 1 & \text{if } a \text{ is a forward arc in } C, \\ -1 & \text{if } a \text{ is a backward arc in } C, \\ 0 & \text{if } a \notin C. \end{cases}$$

The *cycle space of a directed graph* $G = (N, A)$ is the linear vector space spanned by the $\{0, \pm 1\}$ cycle vectors γ_C of cycles $C \in G$. A *cycle basis* B of G is a basis of the cycle space of G. A cycle basis for a directed graph G can be constructed as follows. First, construct a spanning tree H of G. This spanning tree is not required to be directed. Then, iteratively add a non-tree arc a to H. Together with the arcs in H, the non-tree arc a forms a cycle. This cycle is said to be *generated* by a. The set of cycles generated by all non-tree arcs forms a basis of the cycle space of G. Hence, the dimension of the cycle space of a directed graph equals $c = |A| - |N| + 1$. For a cycle basis $B = \{C_1, \ldots, C_c\}$ with cycle basis vectors $\gamma_1, \ldots, \gamma_c$, the *cycle matrix* Γ_B is the $c \times |A|$ matrix with $\gamma_1, \ldots, \gamma_c$ as rows.

5.3.2 Expressing Cycle Periodicity with Cycle Bases

The following definition characterizes the class of cycle bases that can be used for stating the cycle periodicity constraints (5.3a).

Definition 5.9. A cycle basis B of G is an *integral cycle basis* if every non-basis cycle is an integer linear combination of the cycles in B.

Let us explain this definition in more detail. Consider a cycle basis $B = \{C_1, \ldots, C_c\}$ with basis cycle vectors $\gamma_1, \ldots, \gamma_c$. For a cycle $D \notin B$, let $(\lambda_D^1, \ldots, \lambda_D^c)$ be the unique linear combination of basis cycles that yields D, that is,

$$\gamma_{Da} = \sum_{i=1}^{c} \lambda_D^i \gamma_{ia} \text{ for all } a \in A. \tag{5.4}$$

Then B is an integral cycle basis if $(\lambda_D^1, \ldots, \lambda_D^c)$ is an integral vector for every non-basis cycle D.

The theorem below states that it suffices to enforce the cycle periodicity constraints (5.3a) only for the cycles in an integral cycle basis of G.

Theorem 5.10. If the cycle periodicity property $\sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = Tq_C$ holds for every cycle C in an integral cycle basis B of G, then it holds for every cycle in G.

Proof. Let $B = \{C_1, \dots, C_c\}$ be an integral cycle basis of G, with cycle vectors $\gamma_1, \dots, \gamma_c$. Suppose that the cycle periodicity property holds for every cycle $C_i \in B$:

$$\sum_{a \in C_i^+} x_a - \sum_{a \in C_i^-} x_a = \sum_{a \in A} \gamma_{ia} x_a = T q_i \text{ for all } i = 1, \dots, c, \tag{5.5}$$

where q_i is the integer variable for cycle C_i. Consider a non-basis cycle D, and let $(\lambda_D^1, \dots, \lambda_D^c)$ be the unique linear combination of basis cycles that span D, so

$$\gamma_{Da} = \sum_{i=1}^{c} \lambda_D^i \gamma_{ia} \text{ for all } a \in A. \tag{5.6}$$

For the directed sum of tensions along D we have

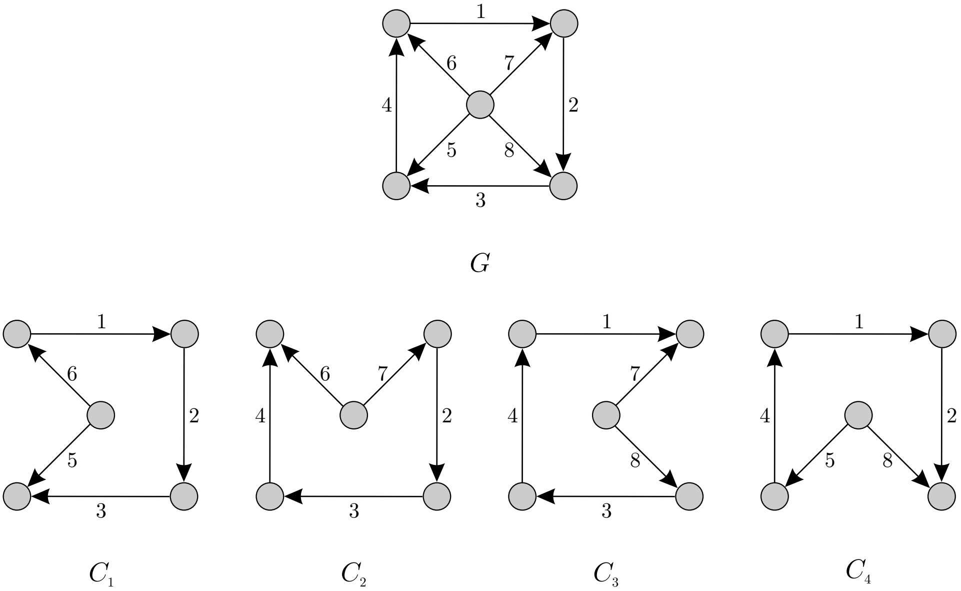
$$\begin{aligned}\sum_{a \in D^+} x_a - \sum_{a \in D^-} x_a = \sum_{a \in A} \gamma_{Da} x_a = \sum_{a \in A} x_a \sum_{i=1}^{c} \lambda_D^i \gamma_{ia} = \sum_{i=1}^{c} \lambda_D^i \sum_{a \in A} \gamma_{ia} x_a \\ = T \sum_{i=1}^{c} \lambda_D^i q_i.\end{aligned}$$

Since B is an integral cycle basis, we have $(\lambda_D^1, \dots, \lambda_D^c) \in \mathbb{Z}$. Together with assumption (5.5), this implies that $\sum_{i=1}^{c} \lambda_D^i q_i$ is integer. Therefore, the cycle periodicity property holds for any cycle $D \notin B$. ■

Theorem 5.10 proves our claim: as long as we use an integral cycle basis, it suffices to require the cycle periodicity property only for the cycles in the cycle basis. Next, we present an example which illustrates that using a non-integral cycle basis to formulate the CPF may yield an incorrect solution to the CRTP.

5.3.3 An Example of Violated Cycle Periodicity

First, we present an example of a non-integral cycle basis. To that end, consider the directed graph G in Figure 5.1, which was presented by Hartvigsen and Zemel (1989) for a related cycle basis problem for undirected graphs. We claim that the four cycles $C_1, \dots, C_4$ form a cycle basis B for G. First, note that the cycle space of



G

C_1 C_2 C_3 C_4

Figure 5.1: Directed graph G with cycle basis $B = \{C_1, \ldots, C_4\}$.

G has dimension $c = 4$. Next, consider the cycle matrix Γ_B:

$$\begin{array}{c c}
 & \begin{array}{cccccccc} \scriptstyle 1 & \scriptstyle 2 & \scriptstyle 3 & \scriptstyle 4 & \scriptstyle 5 & \scriptstyle 6 & \scriptstyle 7 & \scriptstyle 8 \end{array} \\
\begin{array}{c} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & -1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & -1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & -1 \end{bmatrix}
\end{array}$$

One can check that it is impossible to construct a linear combination of the cycle vectors $\gamma_1, \ldots, \gamma_4$ which yields the zero vector. This proves our claim that $B = \{C_1, \ldots, C_4\}$ is a basis of the cycle space of G.

Now consider the non-basis cycles C_5, C_6, C_7 in Figure 5.2, with the cycle vectors

$$\gamma_5 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \end{bmatrix}$$
$$\gamma_6 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix}$$
$$\gamma_7 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

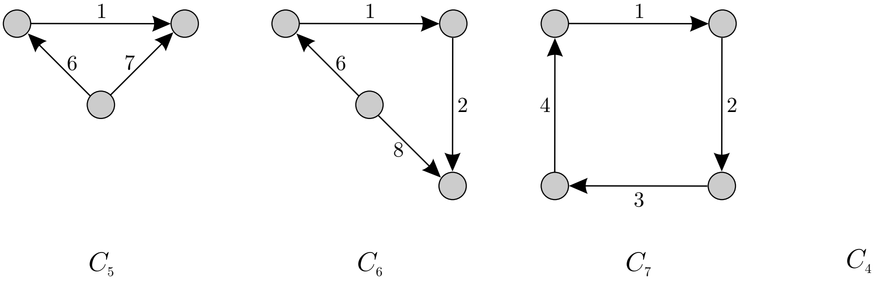


Figure 5.2: Non-basis cycles C_5, C_6, C_7.

The cycle vectors $\gamma_5, \gamma_6, \gamma_7$ can be expressed in the basis cycle vectors as follows

$$\begin{aligned}\gamma_5 &= \tfrac{1}{3}\gamma_1 - \tfrac{2}{3}\gamma_2 + \tfrac{1}{3}\gamma_3 + \tfrac{1}{3}\gamma_4,\\ \gamma_6 &= \tfrac{2}{3}\gamma_1 - \tfrac{1}{3}\gamma_2 - \tfrac{1}{3}\gamma_3 + \tfrac{2}{3}\gamma_4,\\ \gamma_7 &= \tfrac{1}{3}\gamma_1 + \tfrac{1}{3}\gamma_2 + \tfrac{1}{3}\gamma_3 + \tfrac{1}{3}\gamma_4.\end{aligned}$$

Since B is a basis, these expressions are unique. It follows that B is non-integral. Moreover, the three cycles C_5, C_6, C_7 represent the structure of *all* cycles in G that are not in the cycle basis, and so *every* non-basis cycle in G is a non-integer linear combination of the cycles in B. Changing the direction of a cycle or arc does not disturb the result, since the former means changing the sign of a row, and the latter means changing the sign of a column.

A non-integer linear combination of basis cycles poses the following problem for the CPF. Suppose that the CPF contains cycle periodicity constraints for the cycles in B. For example, for C_5, we have

$$\gamma_5 = \tfrac{1}{3}\gamma_1 - \tfrac{2}{3}\gamma_2 + \tfrac{1}{3}\gamma_3 + \tfrac{1}{3}\gamma_4,$$

giving

$$\begin{aligned}\gamma_5 x &= \left(\tfrac{1}{3}\gamma_1 - \tfrac{2}{3}\gamma_2 + \tfrac{1}{3}\gamma_3 + \tfrac{1}{3}\gamma_4\right) x\\ &= \tfrac{1}{3}Tq_1 - \tfrac{2}{3}Tq_2 + \tfrac{1}{3}Tq_3 + \tfrac{1}{3}Tq_4\\ &= \tfrac{1}{3}T\left(q_1 - 2q_2 + q_3 + q_4\right).\end{aligned}$$

This means that the directed sum of tensions along C_5 may be unequal to an integer multiple of T.

Next, as a concrete example of the problem that a non-integral cycle basis may cause for the CPF, consider the CRTP instance in Figure 5.3(a), with $T = 60$. Each

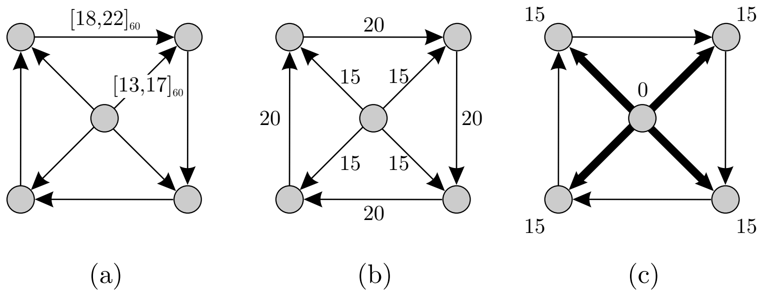


Figure 5.3: (a) Infeasible CRTP instance, (b) Solution $\bar{x}$ that is feasible for the CFP with cycle basis $B = \{C_1, \ldots, C_4\}$, (c) Infeasible implied periodic potential.

of the outer arcs has periodic time window $[18, 22]_{60}$, and each of the inner arcs has periodic time window $[13, 17]_{60}$. A closer inspection of the instance immediately shows that it is infeasible.

Suppose that we formulate the CPF (5.3) with the cycle basis $B = \{C_1, \ldots, C_4\}$. Then the solution

$$\bar{x} = \begin{bmatrix}\bar{x}_1, \ldots, \bar{x}_8\end{bmatrix} = \begin{bmatrix}20 & 20 & 20 & 20 & 15 & 15 & 15 & 15\end{bmatrix}$$

is feasible, since $l_a \leq \bar{x}_a \leq u_a$ for all $a \in A$, and $\sum_{a \in C^+} \bar{x}_a - \sum_{a \in C^-} \bar{x}_a = 60$ for all $C \in B$. The solution $\bar{x}$ is shown in Figure 5.3(b). However, for the non-basic cycles C_5, C_6, C_7, we have respectively

$$\bar{x}_1 + \bar{x}_6 - \bar{x}_7 = 20, \tag{5.7}$$

$$\bar{x}_1 + \bar{x}_2 + \bar{x}_6 - \bar{x}_8 = 40, \tag{5.8}$$

$$\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \bar{x}_4 = 80. \tag{5.9}$$

Since $T = 60$, these cycles clearly do not satisfy the cycle periodicity property. Finally, consider Figure 5.3(c), which tries to construct a periodic potential solution for the periodic tension solution $\bar{x}$. Choosing the value 0 for the center node, the corner nodes each have to take the value 15 because of the value $\bar{x}_a = 15$ for the bold arcs $a = 5, \ldots, 8$. Clearly, this solution is infeasible for $T = 60$.

5.4 Sequencing Trains on Tracks[*]

In this section, we show that there exists a relation between the integer variables q_C of certain cycles C, and the sequence in which trains travel along tracks, and leave and enter stations. First, we state a sequencing lemma in terms of the q_C variables. Next, the cycles that relate to train sequences are identified. Based on these cycles, we derive a class of valid inequalities for the CPF. Finally, we show that there exists a relation between the identified cycles and the Linear Ordering Problem (Grötschel et al., 1984, 1985).

5.4.1 Cyclic Sequencing in the CPF

The cyclic sequencing ideas for the PESP from Section 4.4 are easily extended to the CPF. Indeed, using Lemma 5.6, we can state Corollary 4.16 in terms of q_C variables.

Lemma 5.11. Consider a cycle $C = (1, \ldots, k, 1)$, a periodic tension x, and the corresponding vector q. Suppose that some set of constraints prevents any subsequent pair of events $(i, i+1) \in C$ to take place concurrently. Then the events $1, \ldots, k$ are cyclically sequenced as $1 \to \cdots \to k$ if and only if $q_C = 1 - |C^-|$.

Proof. Apply Lemma 5.6 to Corollary 4.16. ■

5.4.2 Safety Triangles

In Section 3.5 we formulated the CRTP as (3.16). Consider the safety constraints (3.16f), (3.16g) in that formulation:

$$d_n^{t'} - d_n^t \in [r_a^t - r_a^{t'} + h, T - h]_T \quad \text{for all } a = (n,m) \in \mathcal{A}, (t,t') \in \mathcal{T}_a, \qquad (3.16f)$$
$$a_n^{t'} - d_n^t \in [r_a^t + r_a^{t'} + h, T - h]_T \quad \text{for all } a = (n,m) \in \mathcal{A}^s, (t,t') \in \mathcal{T}_a. \qquad (3.16g)$$

In order to describe these constraints in terms of the constraint graph formulation (3.20), consider k trains $t_1, \ldots, t_k$ traveling along a track $a \in \mathcal{A}$. Let the departure nodes involved in (3.16f), (3.16g) be indexed $1, \ldots, k$ corresponding to their train index, so node i represents the departure of train t_i. As was described in Section 3.8, the safety constraints for a track a define a safety clique K_a in G.

[*] This section is partly based on Liebchen and Peeters (2002b).

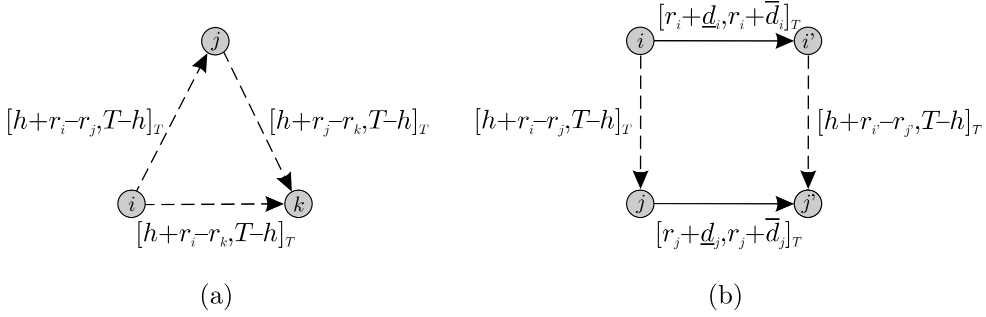


Figure 5.4: (a) Safety triangle (i, j, k), (b) dwell square (i, j, k, l)

Let us write the constraint graph formulation of the safety constraints (3.16f), (3.16g) as

$$h + r_i - r_j \le v_j - v_i + Tp_{ij} \le T - h \qquad \text{for all } i < j = 1, \dots, t. \tag{5.10}$$

Here, we use r_i as a shorthand notation for the trip time of the train corresponding to node i. Moreover, we assume that each safety constraint is directed from a lower indexed node i to a higher indexed node j. That is, we assume that a train pair (t, t') in $\mathcal{T}_a$ corresponds to a safety arc (i, j), where node i represents train t, and node j represents train t'. So, for each safety arc $(i, j), r_j \le r_i$.

Definition 5.12. For a track $a \in \mathcal{A}$, a *safety triangle* (i, j, k) is a cycle (i, j, k, i) on three nodes $i, j, k \in K_a$, which is directed as the arc (i, j). The safety arcs in (i, j, k) are directed such that the time windows are

$$\begin{aligned} &[h + r_i - r_j, T - h]_T \quad \text{for arc } (i, j),\\ &[h + r_j - r_k, T - h]_T \quad \text{for arc } (j, k),\\ &[h + r_i - r_k, T - h]_T \quad \text{for arc } (i, k). \end{aligned}$$

Note that a safety triangle (i, j, k) contains two forward arcs and one backward arc. Moreover, we have $r_k \le r_j \le r_i$. Figure 5.4(a) shows a safety triangle.

The cycle periodicity integer variable for a safety triangle (i, j, k) is denoted by q_{ijk}. Theorem 5.7 gives the following bounds on q_{ijk}

$$\begin{aligned} a_{ijk} &= \left\lceil -1 + \frac{3h + r_i - r_k}{T} \right\rceil,\\ b_{ijk} &= \left\lfloor 2 - \frac{3h + r_i - r_k}{T} \right\rfloor. \end{aligned}$$

So, under the condition $r_i - r_k \leq T - 3h$, we have $q_{ijk} \in \{0, 1\}$.

Because of the safety constraints in K_a, none of the events $1, \ldots, k$ can take place concurrently. Applying Lemma 5.11 to (i, j, k) gives that the events i, j, k are cyclically sequenced as $i \to j \to k$ if and only if $q_{ijk} = 0$. On the other hand, applying the lemma to the oppositely directed safety triangle (k, j, i) gives that the opposite cyclic sequence $k \to j \to i$ occurs if and only if $q_{kji} = -1$. Using

$$Tq_{ijk} = x_{ij} + x_{jk} - x_{ik} = -Tq_{kji},$$

we obtain the following result:

$$q_{ijk} = \begin{cases} 0 & \text{if and only if events } i, j, k \text{ take place in the cyclic sequence } i \to j \to k, \\ 1 & \text{if and only if events } i, j, k \text{ take place in the cyclic sequence } k \to j \to i. \end{cases}$$

The above analysis generalizes to the following lemma.

Lemma 5.13. Consider a track $a \in \mathcal{A}$, and its safety clique K_a. Let the nodes $1, \ldots, k \in K_a$ represent the departures of the trains $t_1, \ldots, t_k$ using the track. For each safety triangle $(i, j, k) \in K_a$ with $r_i - r_k \leq T - 3h$, the following holds:

$$q_{ijk} = \begin{cases} 0 & \text{if and only if } t_i, t_j, t_k \text{ depart in the cyclic sequence } t_i \to t_j \to t_k, \\ 1 & \text{if and only if } t_i, t_j, t_k \text{ depart in the cyclic sequence } t_i \to t_k \to t_j. \end{cases}$$

Remark 5.14. A common practical value for h is $3 \leq h \leq 5$. Trip time differences are never greater than 45 minutes in the Dutch network. With a cycle time $T = 60$, this means that the condition $r_i - r_k \leq T - 3h$ is always satisfied in practice.

Next, consider the case in which one of the arcs in the safety triangle (i, j, k) is a frequency arc with time window $[f, T - f]$. For this case, we can derive a similar analysis as above. The analysis is a bit more extensive, since we need to consider each of the three arcs as a frequency arc. The resulting condition is $\Delta r \leq T - 2h - f$, where Δr is the maximum trip time difference between the trains. With the common practical value $25 \leq f \leq 30$, this condition is also always satisfied in practice.

Similarly, for the case in which two of the arcs in (i, j, k) are frequency arcs, one can derive the condition $\Delta r \leq T - h - 2f$, where Δr is the trip time for the non-frequency arc. In this case, the condition may be violated in practice. If the condition is violated, then either $a_C = b_C = 0$, or $a_C = b_C = 1$. In both cases, the sequence for the three trains is fixed beforehand.

5.4.3 Dwell Squares

For a further insight into the sequence in which trains travel along a track, this section studies the order in which two trains enter and leave a network node $n \in \mathcal{N}$. To that end, consider a dwell square, which is defined as follows.

Definition 5.15. For a network node $n \in \mathcal{N}$, a *dwell square* (i, j, j', i') is a cycle (i, j, j', i', i) consisting of two merged trip-dwell arcs (i, i') and (j, j'), and of the two safety arcs (i, j) and (i', j'). Nodes i' and j' correspond to the departures from n, and nodes i and j to the departures from the previous node m. The two trip-dwell arcs have time windows

$$[r_i + \underline{d}_i, r_i + \overline{d}_i]_T,$$
$$[r_j + \underline{d}_j, r_j + \overline{d}_j]_T,$$

and the two safety arcs have time windows

$$[h + r_i - r_j, T - h]_T,$$
$$[h + r_{i'} - r_{j'}, T - h]_T.$$

Here, the trip times r_i are the same shorthand notation as before. The train corresponding to the nodes j and j' is the faster train, as follows from the direction of the safety arcs. For this train we use the shorthand dwell time window notation $[\underline{d}_j, \overline{d}_j]$, for the other train $[\underline{d}_i, \overline{d}_i]$. Figure 5.4(b) shows a dwell square.

The cycle periodicity integer variable for a dwell square (i, j, j', i') is denoted by $q_{ijj'i'}$. Theorem 5.7 gives the following bounds on $q_{ijj'i'}$:

$$a_{ijj'i'} = \left\lceil -1 + \frac{2h - (\overline{d}_i - \underline{d}_j)}{T} \right\rceil,$$
$$b_{ijj'i'} = \left\lfloor 1 - \frac{2h + (r_i - r_j) + (r_{i'} - r_{j'}) - (\overline{d}_j - \underline{d}_i)}{T} \right\rfloor.$$

Therefore, whenever $\overline{d}_i - \underline{d}_j < 2h$, we have $a_{ijj'i'} = 0$. Since the train corresponding to the nodes j and j' is the faster train, we have $r_i - r_j \geq 0, r_{i'} - r_{j'} \geq 0$. Therefore, as long as $\overline{d}_j - \underline{d}_i < 2h$, we obtain that $b_{ijj'i'} = 0$, and thus that $q_{ijj'i'} = 0$.

Because of the constraints in the dwell square, no pair of subsequent events in the dwell square can take place concurrently. Note that it is allowed for i and j', and for i' and j to take place concurrently. Applying Lemma 5.11 twice to the dwell square (i, j', i', j'), once directing the cycle clockwise, and once directing it counterclockwise, we obtain

$$q_{ijj'i'} = \begin{cases} -1 & \text{if and only if } i, j, i', j' \text{ occur in the cyclic sequence } i \to j \to j' \to i', \\ 1 & \text{if and only if } i, j, i', j' \text{ occur in the cyclic sequence } j \to i \to i' \to j'. \end{cases}$$

These two sequences are the only sequences in which one of the trains enters node n after the other train, and leaves the node before the other train. Therefore, whenever $\overline{d}_i - \underline{d}_j < 2h$, and thus $q_{ijj'i'} = 0$, the two trains depart from n in the same sequence as the sequence in which they arrive.

Suppose that a third train dwells at node n, with dwell arc (k, k'), and that the constraint graph contains the dwell squares (i, k, k', i') and (j, k, k', j'). Then the analysis above shows that, if $\overline{d}_i - \underline{d}_k < 2h$ and $\overline{d}_k - \underline{d}_i < 2h$, then the trains corresponding to the nodes i and k arrive at and depart from n in the same sequence. Similarly, the trains corresponding to the nodes j and k arrive at and depart from network node n in the same sequence if $\overline{d}_j - \underline{d}_k < 2h$ and $\overline{d}_k - \underline{d}_j < 2h$. And if the arrival and departure sequences for the three trains are the same, then they must also travel along the tracks (m, n) and (n, m') in the same sequence. Then, by Lemma 5.13, we have that

$$q_{ijk} = q_{i'j'k'},$$

with (i, j, k) and (i', j', k') safety triangles in the safety cliques A_{mn} and $A_{nm'}$, respectively.

The above analysis is summarized in the following lemma.

Lemma 5.16. Consider a network node $n \in \mathcal{N}$, and three dwell squares (i, j, j', i'), (i, k, k', i'), (j, k, k', j') for that network node. The trains involved in the dwell square enter network node n using the track (m, n), and depart using the track (n, m'). Suppose that $\overline{d}_i - \underline{d}_j < 2h$, $\overline{d}_j - \underline{d}_i < 2h$, $\overline{d}_i - \underline{d}_k < 2h$, $\overline{d}_k - \underline{d}_i < 2h$, $\overline{d}_j - \underline{d}_k < 2h$, and $\overline{d}_k - \underline{d}_j < 2h$. Then it holds that

$$q_{ijk} = q_{i'j'k'},$$

with (i, j, k) and (i', j', k') safety triangles in the safety cliques A_{mn} and $A_{nm'}$, respectively.

Remark 5.17. In practice, dwell windows are usually between 1 and 10 minutes wide. As was mentioned before, the headway time h takes practical values between 3 and 5 minutes. So, in many cases, the condition of Lemma 5.16 is satisfied.

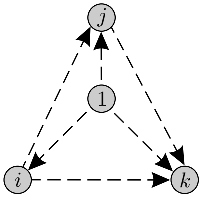


Figure 5.5: Triangles $(1,i,j),(1,j,k),(1,i,k)$ and (i,j,k)

5.4.4 Valid Inequalities for Sequencing Trains

Consider a track $a \in \mathcal{A}$, and the safety clique K_a with nodes $1,\ldots,k$, and $i < j$ for each arc $(i,j) \in K_a$, similar to the above. All arcs (i,j) are safety arcs with time window $[h + r_i - r_j, T - h]_T$. Let H be the star tree in K_a rooted at node 1, so H consists of the arcs $(1,i)$ for $i = 2,\ldots,k$. The tree H generates a cycle basis B, which consists of the safety triangles $(1,i,j)$ for all $(i,j) \notin H$. As before, each triangle $(1,i,j)$ contains two forward arcs and one backward arc, and therefore we have $q_{1ij} \in \{0,1\}$ for all cycles $(1,i,j) \in B$. The CPF for the constraint graph induced by K_a, with cycle basis B, reads

$$\begin{array}{lllr}
\text{Find a solution} & (x,q) & & \\
\text{satisfying} & \displaystyle\sum_{(i,j)\in C^+} x_{ij} - \sum_{(i,j)\in C^-} x_{ij} = Tq_C & \text{for all } C \in B & (5.11a) \\
& h + r_i - r_j \le x_{ij} \le T - h & \text{for all } (i,j) \in K_a & (5.11b) \\
& x_{ij} \in \mathbb{R} & \text{for all } (i,j) \in K_a & (5.11c) \\
& q_C \in \{0,1\} & \text{for all } C \in B & (5.11d)
\end{array}$$

The following theorem presents valid inequalities for such a constraint graph K_a.

Lemma 5.18. Consider a non-basic triangle $(i,j,k) \in K_a$, with $i < j < k$. Then the following inequality is valid

$$0 \le q_{1ij} + q_{1jk} - q_{1ik} \le 1. \qquad (5.12)$$

***Proof*.** Figure 5.5 shows that the cycle vector γ_{ijk} for triangle (i,j,k) is expressed in basis cycle vectors as follows

$$\gamma_{ijk} = \gamma_{1ij} + \gamma_{1jk} - \gamma_{1ik}.$$

Therefore, the cycle periodicity variable q_{ijk}, which is *not* in the model, is expressed as $q_{ijk} = q_{1ij} + q_{1jk} - q_{1ik}$. Applying Theorem 5.7 to triangle (i,j,k) yields $a_{ijk} = 0, b_{ijk} = 1$. ■

Remark 5.19. Note that (5.12) is not implied by the bounds $0 \leq q_{1ij}, q_{1jk}, q_{1ik} \leq 1$, since these only give $-1 \leq q_{1ij} + q_{1jk} - q_{1ik} \leq 2$. Further, the analysis in this section does not depend on the star tree H being rooted at node 1 of K_a. Such a star tree is only used because it yields triangle cycles $(1,i,j)$ with integer variables $q_{1ij} \in \{0,1\}$, and these keep the argument clear. Using any other star tree also gives a cycle basis consisting only of triangles. And because of the time windows $[h + r_i - r_j, T - h]$ for all arcs, these triangles correspond to integer variables that can still take only two values. But those values may be different from 0 and 1, for example $\{-1,0\}$ or $\{1,2\}$.

Using the analogy between cyclic and linear sequencing presented in Lemma 4.17, one can also interpret Lemma 5.18 as follows. Choose event i as reference point, and consider the linear sequence of the events j and k. The variable q_{ijk} can be seen as a binary decision variable, representing whether the events j and k are linearly sequenced as $j \to k$, corresponding to the value $q_{ijk} = 0$, or as $k \to j$, giving the value $q_{ijk} = 1$. Choosing node 1 as reference point, we know that the *linear* sequence of the events i, j, k may not contain a cyclic sub-sequence. So the cyclic subsequence given by $i \to j$, $j \to k$, $k \to i$ is forbidden. Interpreting the variables q as sequence indicators, one sees that this subsequence corresponds to the solution

$$(q_{1ij}, q_{1jk}, q_{1ik}) = (0,0,1).$$

Similarly, the cyclic subsequence given by $i \to k$, $k \to j$, $j \to i$ yields the forbidden solution

$$(q_{1ij}, q_{1jk}, q_{1ik}) = (1,1,0).$$

Inequality (5.12) exactly forbids these two solutions. Moreover, it allows all other solutions, as can be checked.

In the light of the above, the valid inequality (5.12) can be seen as a so-called *3-dicycle inequality* for the Linear Ordering Problem (see Grötschel et al., 1984, 1985). In fact, from the Linear Ordering Problem, it follows that by enforcing inequalities (5.12) for all non-basis triangles in K_a, *all* cyclic sub-sequences are forbidden. Moreover, any other class of valid inequalities for the Linear Ordering Problem can be used for this substructure of the CRTP graph.

5.5 Cutting Planes for the CPF

This section describes two classes of cutting planes for the CPF. The first class follows from Lemma 5.8, and generalizes the safety triangle inequalities from the previous section. The second class was introduced by Nachtigall (1999).

5.5.1 Cycle Cuts

Lemma 5.8 formulates Odijk's Theorem 4.3 in terms of the cycle periodicity variables q_C. The lemma states that an integer vector q is feasible for the CPF if and only if $a_C \leq q_C \leq b_C$ for all $C \in G$. The CPF however contains only c cycle periodicity variables q_C, one for each cycle $C \in B$. The lemma below shows how the cycle bounds a_D and b_D for non-basis cycles D can be added to the CPF as cutting planes.

Lemma 5.20. Let $B = \{C_1, \ldots, C_c\}$ be a cycle basis of G, with cycle vectors $\gamma_1, \ldots, \gamma_c$. For a non-basis cycle $D \notin B$, let $(\lambda_D^1, \ldots, \lambda_D^c)$ be the linear combination of basis cycles that span D. Then the following inequality is valid

$$a_D \leq \sum_{i=1}^{c} \lambda_D^i q_i \leq b_D. \tag{5.13}$$

Proof. The proof follows directly from Lemma 5.8. ■

Lemma 5.20 is a general form of the safety triangle inequalities from the previous section. We first derived those inequalities to show their relation with the sequence in which trains depart from a station, and with the Linear Ordering Problem.

5.5.2 Change Cycle Cuts

Consider a cycle periodicity constraint written in the following way

$$\sum_{a \in C^-} x_a = \sum_{a \in C^+} x_a - Tq_C.$$

Let $x^+ = \sum_{a \in C^+} x_a$, and $x^- = \sum_{a \in C^-} x_a$. The dotted parallel lines in Figure 5.6 depict the function $x^- = x^+ - Tq_C$ for fixed values of q_C. The dotted lines for the values a_C and b_C correspond to Odijk's cycle cuts, and are therefore partly drawn as bold arcs.

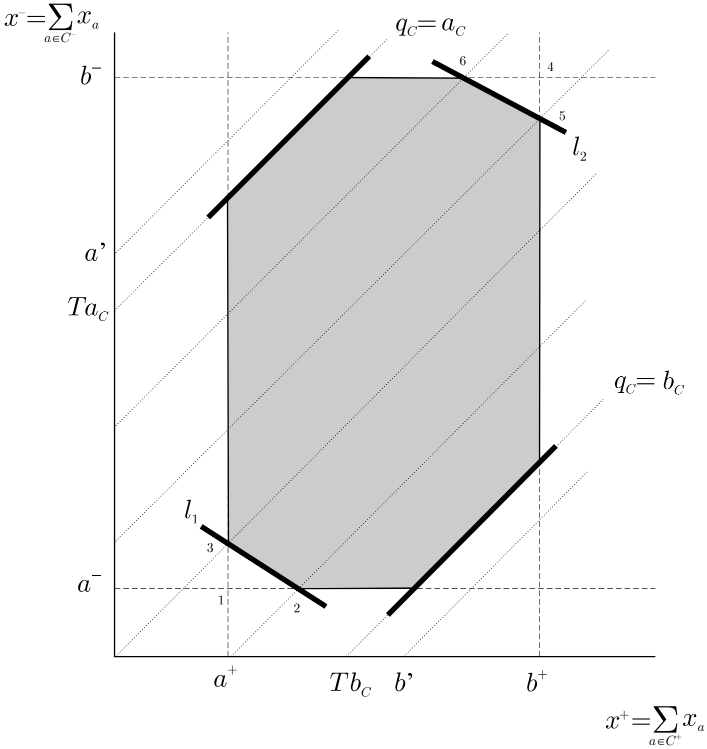


Figure 5.6: A change cycle cut.

Next, define

$$a^+ = \sum_{a\in C+} l_a,\ b^+ = \sum_{a\in C+} u_a,$$
$$a^- = \sum_{a\in C^-} l_a,\ b^- = \sum_{a\in C^-} u_a.$$

Note that we do not divide by T or round here. It is clear that $a^+ \le x^+ \le b^+$, and $a^- \le x^- \le b^-$. These bounds are drawn as dashed lines in Figure 5.6. Further, define $a' = a^+ - b^-$, and $b' = b^+ - a^-$. The uppermost and lowermost dotted lines in Figure 5.6 depict the function $x^- = x^+ - Tq_C$ for the values a' and b'. Note that we have $a_C = \lceil a'/T \rceil$ and $b_C = \lfloor b'/T \rfloor$. So, Odijk's cycle cuts cut off the region between the lines corresponding to a' and a_C, and the region between the lines corresponding to b' and b_C.

The figure further shows that the gray polygon is the feasible region for the cycle periodicity constraint. Therefore, the lines l_1 and l_2 cut off two triangles of the feasible region.

Let us first consider an expression for the line l_1. The line that passes through point 1 in the figure, with coordinates $(x^+, x^-) = (a^+, a^-)$, and that is parallel to the dotted lines, corresponds to the equation

$$x^- = x^+ - (a^+ - a^-).$$

To keep the figure clear, this line is not drawn. Point 2 is the intersection of the line $x^- = a^-$, and the first line below the line $x^- = x^+ - (a^+ + a^-)$ that corresponds to an integer value of q_C. So, point 2 is the intersection of the line $x^- = a^-$ and the line $x^- = x^+ - \overline{a}$, where $\overline{a}$ is defined as

$$\overline{a} = T \left\lceil \frac{a^+ - a^-}{T} \right\rceil .$$

It follows that point 2 has coordinates $(x^+, x^-) = (a^- + \overline{a}, a^-)$. Similarly, point 3 is the intersection of the line $x^+ = a^+$, and the first line above the line $x^- = x^+ - (a^+ + a^-)$ that corresponds to an integer value of q_C. That means that point 3 is the intersection of the line $x^+ = a^+$ and the line $x^- = x^+ - \overline{a} - 1$, under the condition $a^+ - a^- \neq 0$ modulo T. Therefore, point 3 has coordinates $(x^+, x^-) = (a^+, a^+ - \overline{a} - 1)$.

With these coordinates for the points 2 and 3, the line l_1 is represented by the inequality

$$x^- \geq \alpha_1 x^+ + \beta_1,$$

with

$$\alpha_1 = 1 + \frac{1}{a^- - a^+ + \overline{a}}, \text{ and } \beta_1 = (1 - \alpha_1)a^+ - \overline{a} - 1.$$

A similar analysis for the points 4, 5, and 6 yields that line l_2 passes through point 5 with coordinates $(x^+, x^-) = (b^+, b^+ - \overline{b})$, and through point 6 with coordinates $(x^+, x^-) = (b^- + \overline{b} - 1, b^-)$, where $\overline{b}$ is defined as

$$\overline{b} = T \left\lceil \frac{b^+ - b^-}{T} \right\rceil .$$

Therefore, line l_2 is represented by the inequality

$$x^- \leq \alpha_2 x^+ + \beta_2,$$

with

$$\alpha_2 = \frac{b^+ - b^- - \overline{b}}{b^+ - b^- - \overline{b} + 1}, \text{ and } \beta_2 = (1 - \alpha_2)b^+ - \overline{b}.$$

Nachtigall (1999) introduced these so-called *change cycle cuts* with the following theorem.

Theorem 5.21 (Nachtigall, 1999). Consider a cycle periodicity constraint (5.3a). Let $a^+, b^+, a^-, b^-, \alpha_1, \beta_1, \alpha_2$, and β_2 be defined as above, and a_C and b_C as in Theorem 4.3. If $\alpha_1 < 0$, then the following inequality is valid

$$\sum_{a \in C^-} x_a \geq \beta_1 + \alpha_1 \sum_{a \in C^+} x_a. \tag{5.14}$$

If $\alpha_2 < 0$, then the following inequality is valid

$$\sum_{a \in C^-} x_a \geq \beta_2 + \alpha_2 \sum_{a \in C^+} x_a. \tag{5.15}$$

5.6 Cycle Bases and Fundamentality[*]

This section shows that the class of fundamental cycle bases is integral. In the existing literature, there is some ambiguity regarding the terminology for fundamental cycle bases (see Horton, 1987, Deo, 1982, Bollobás, 1998, Mardon, 1990, Gleiss, 2001). Therefore, we clearly distinguish between fundamental cycle bases and strictly fundamental cycle bases.

Definition 5.22 (Strictly Fundamental Cycle Basis). A set B of $c = |A| - |N| + 1$ cycles in a graph G is a *strictly fundamental cycle basis* if there exists a spanning tree H of G, such that its non-tree arcs generate B.

Lemma 5.23. Any non-basis cycle D is a $\{0, \pm 1\}$ linear combination of the cycles in a strictly fundamental cycle basis B.

Proof. Each cycle in a strictly fundamental cycle basis B contains an arc that appears exclusively in it, namely its generating non-tree arc. Placing these c unique arcs in the first columns, the cycle matrix Γ_B can be written in the form $\Gamma_B = [I|X]$, where I is the identity matrix, and X represents the remaining part of the matrix.

Let the unique linear combination of basis cycles that yields the non-basis cycle D be denoted by $\lambda = (\lambda_D^1, \ldots, \lambda_D^c)$. That is, $\gamma_D = \sum_{i=1}^{c} \lambda_D^i \gamma_i$, or, equivalently

$$\gamma_D = \Gamma_B^t \lambda_D = \begin{bmatrix} I & X \end{bmatrix}^t \lambda_D = \begin{bmatrix} I \\ X^t \end{bmatrix} \lambda_D = \begin{bmatrix} \lambda_D \\ X^t \lambda_D \end{bmatrix}.$$

[*] This section is based on Liebchen and Peeters (2002a).

So, for the first c elements of γ_D, we have $[\gamma_{D1}, \ldots, \gamma_{Dc}] = [\lambda_D^1, \ldots, \lambda_D^c]$. This uniquely determines λ_D. Since $\gamma_D \in \{0, \pm 1\}^{|A|}$, it follows that $\lambda_D \in \{0, \pm 1\}^c$. ■

The proof indicates the following scheme for constructing the linear combination $\lambda_D = (\lambda_D^1, \ldots, \lambda_D^c)$ that gives the non-basis cycle D:

$$\lambda_D^i = \begin{cases} 1 & \text{if the generating non-tree arc of } \gamma_i \text{ is a forward arc in } D, \\ -1 & \text{if the generating non-tree arc of } \gamma_i \text{ is a backward arc in } D, \\ 0 & \text{otherwise.} \end{cases}$$

Definition 5.24 (Fundamental Cycle Basis). A set $B = \{C_1, \ldots, C_c\}$ of cycles in a graph G is a *fundamental cycle basis* if there exists an ordering of the cycles in B such that $C_i \setminus (C_{i-1} \cup \cdots \cup C_1) \neq \emptyset$ for $i = 2, \ldots, c$.

Intuitively, a fundamental cycle basis can be interpreted as follows. Given an ordering satisfying the fundamentality definition, each cycle contains at least one arc that is not part of its predecessors in the ordering. It is easy to see that a strictly fundamental cycle basis is also fundamental. Indeed, since each cycle in a strictly fundamental cycle basis has a unique arc, it holds that $C_i \setminus (C_{i-1} \cup \cdots \cup C_1) \neq \emptyset$ for *any ordering* of the cycles in a strictly fundamental basis.

Lemma 5.25. Any non-basis cycle D is an integer combination of the cycles in a fundamental cycle basis B.

***Proof*.** Let the basis cycle vectors $\gamma_1, \ldots, \gamma_c$ be ordered according to the fundamentality definition. We arrange the cycle matrix Γ_B as follows. Row i contains cycle vector γ_i, and the columns are arranged such that the unique arc of basis cycle C_i is placed in column $c - i + 1$. Moreover, we direct basis cycle C_i such that entry i of γ_i equals one. Since each cycle in the basis contains *at least* one arc that is not contained in its predecessors, there may be multiple candidates for being placed in the 'unique' column. In that case, we arbitrarily choose one. Arranging the cycle matrix in this way, it is written as $\Gamma_B = [A|X]$, where A is an upper triangular matrix with all ones on the diagonal, and X represents the remaining part of the matrix.

Next, let the unique linear combination of basis cycles that yields D be denoted by $\lambda_D = (\lambda_D^1, \ldots, \lambda_D^c)$. That is, $\gamma_D = \sum_{i=1}^{c} \lambda_D^i \gamma_i$, or, equivalently

$$\gamma_D = \Gamma_B^t \lambda_D = \begin{bmatrix} A & X \end{bmatrix}^t \lambda_D = \begin{bmatrix} A^t \\ X^t \end{bmatrix} \lambda_D = \begin{bmatrix} A^t \lambda_D \\ X^t \lambda_D \end{bmatrix}.$$

Consider the first c elements of γ_D. These give $(\gamma_{D1}, \ldots, \gamma_{Dc}) = A^t \lambda_D$. The matrix A^t is lower triangular with all ones on the diagonal, and all other entries are 0 or ± 1. Therefore the vector λ_D is all-integer. ■

Note that the cycle basis B in Section 5.3.3 is not a fundamental cycle basis. This is easily seen, since each arc appears in at least two cycles of the basis. It is therefore not possible to re-order the cycles $C_1, \ldots, C_4$ such that $C_i \setminus (C_{i-1} \cup \ldots \cup C_1) \neq \emptyset$ for $i = 2, 3, 4$.

One can efficiently check whether a cycle basis B is fundamental, by trying to construct an ordering satisfying the fundamentality condition as follows. In a fundamental cycle basis, at least one arc a in B is covered by only one cycle, say C^*. This cycle becomes cycle C_c in the ordering. The cycles $B \setminus \{C^*\}$ are a fundamental cycle basis for $G \setminus \{a\}$. So, delete a from G, remove C^* from B, and repeat the procedure. If there are multiple candidate cycles that each cover a unique arc, we arbitrarily select one as C^*. In the end, we either have an ordering of B satisfying the fundamentality condition, or B is not fundamental. Note that the outcome of the fundamentality check does not depend on the choice of C^* in the case of multiple candidates, since a candidate cycle that is not chosen keeps the property that it covers a unique arc.

The next corollary classifies strictly fundamental and fundamental cycle bases as integral.

Corollary 5.26. Strictly fundamental and fundamental cycle bases are integral.

Proof. This follows from Lemmata 5.23 and 5.25. ■

5.7 Good Cycle Bases for Cyclic Railway Timetabling

The previous sections showed that it suffices to enforce the cycle periodicity constraints for all cycles in an integral cycle basis, and, moreover, that fundamental cycle bases are integral. But a graph has many different cycle bases, and the question arises whether some are better than others for formulating and applying the CPF.

Suppose that we were to solve the CPF for the CRTP by brute force enumeration of all possibilities for the vector q. Let $W_C = b_C - a_C$ be the *width* of the cycle C.

A cycle periodicity variable q_C can take $W_C + 1$ different values. For a cycle basis B, one sees that the vector $q = (q_1, \dots, q_c)$ can take

$$W(B) = \prod_{C \in B} (W_C + 1) \tag{5.16}$$

different values. We call $W(B)$ the *width* of the cycle basis B. Therefore, brutely enumerating all possible values for q is done in a minimum number of iterations when using a cycle basis with minimum width. And even when using a more sophisticated method, such as Branch&Bound or Branch&Cut, it is still sensible to formulate the CPF with a small or minimum width cycle basis. Therefore, we are interested in small width cycle bases for formulating the CPF. The remainder of this section studies various types of cycle bases, and our computational tests in Chapter 7 will show that, indeed, certain classes of cycle bases perform much better than others in solving the CPF.

5.7.1 Transforming the Cycle Basis Objective Function

The width of a cycle basis is a product and therefore a non-linear quantity. Moreover, the non-linear operation of rounding involved in computing a_C and b_C further obscures the construction of small or minimum width cycle bases.

First, in order to circumvent the product in the cycle basis width, define

$$LW(B) = \log W(B) = \log \prod_{C \in B} (W_C + 1) = \sum_{C \in B} \log(W_C + 1). \tag{5.17}$$

Since the logarithm is a monotonous transformation, a minimum width cycle basis B^* also attains the minimum for the function LW, and vice versa.

Next, consider the impact of rounding in computing a_C and b_C, and thus in computing W_C. To that end, we forget about rounding for the moment, and consider the unrounded bounds a'_C and b'_C defined by

$$a'_C = \frac{1}{T}\left(\sum_{a \in C^+} l_a - \sum_{a \in C^-} u_a\right),$$
$$b'_C = \frac{1}{T}\left(\sum_{a \in C^+} u_a - \sum_{a \in C^-} l_a\right).$$

For a cycle C we define its *unrounded width* W'_C as

$$\begin{aligned}
W'_C = b'_C - a'_C &= \frac{1}{T}\left(\sum_{a\in C^+} u_a - \sum_{a\in C^-} l_a\right) - \frac{1}{T}\left(\sum_{a\in C^+} l_a - \sum_{a\in C^-} u_a\right) \\
&= \frac{1}{T}\left(\sum_{a\in C^+} (u_a - l_a) - \sum_{a\in C^-} (l_a - u_a)\right) \\
&= \frac{1}{T}\sum_{a\in C}(u_a - l_a) \\
&= \frac{1}{T}\sum_{a\in C} w_a.
\end{aligned}$$

That means that the direction of the arcs in C does not matter for the unrounded width W'_C of a cycle C. Moreover, W'_C is a linear function of the time window widths $w_a = u_a - l_a$. The gap between the unrounded and rounded width of a cycle C equals

$$W'_C - W_C = \Big(b'_C - \lfloor b'_C \rfloor\Big) + \Big(\lceil a'_C \rceil - a'_C\Big).$$

Using that, for some $y \in \mathbb{R}, 0 \leq y - \lfloor y \rfloor < 1$ and $0 \leq \lceil y \rceil - y < 1$, we obtain

$$0 \leq W'_C - W_C < 2.$$

So, as a heuristic for approximating $LW(B)$, and thus $W(B)$, we could consider minimizing the following cycle basis objective function:

$$LW'(B) = \sum_{C\in B} \log(W'_C + 1) = \sum_{C\in B} \log(1 + \sum_{a\in C} w_a). \tag{5.18}$$

And, since the direction of the arcs does not matter in this function, $LW'(B)$ may also be defined on the underlying undirected graph U of G. In that case, minimizing $LW'(B)$ is similar to the so-called *minimum cycle basis problem* for undirected graphs. The next section describes the minimum cycle basis problem.

5.7.2 Minimum Cycle Bases and the CPF

Consider the problem of finding a so-called minimum cycle basis in an undirected graph.

Definition 5.27. Given an undirected graph $U = (N, E)$ with edge weights w_e for all $e \in E$, a *minimum cycle basis* of G is a cycle basis B^* that minimizes the cycle basis *weight*

$$\sum_{C \in B^*} \sum_{e \in C} w_e. \tag{5.19}$$

A *minimum (strictly) fundamental cycle basis* is a minimum cycle basis that is (strictly) fundamental.

Several authors studied the problem of constructing a minimum cycle basis. Horton (1987) developed an $O(m^3 n)$ algorithm for constructing a minimum cycle basis. However, the previous section showed that we need an integral cycle basis for the CPF, and Horton's algorithm may return a non-integral cycle basis. Deo et al. (1982) proved that the problem of finding a minimum strictly fundamental cycle basis[1] for the unit edge weight case is NP-complete.

These authors considered cycle bases for undirected graphs, but CRTP constraint graphs are directed. Let $w_a = u_a - l_a$. Suppose that we use the following procedure for constructing a strictly fundamental cycle basis to formulate the CPF.

Step 1. For the underlying undirected graph $U = (N, E)$ of a constraint graph $G = (N, A, l, u)$, set the edge weights to

$$w_e = w_a \text{ for all } e \in E,$$

where e is the underlying edge of arc $a \in A$.

Step 2. For the edge weights w_e, apply a heuristic to compute a small weight (strictly) fundamental cycle basis B' for U.

Step 3. Using Theorem B.1 in Appendix B, arbitrarily direct the cycles in B' to obtain a strictly fundamental cycle basis B for G.

Since wide arcs a, which yield a large increase in the width of a cycle basis, correspond to edges with large weight w_e, the above procedure expectedly constructs a cycle basis with small width. Each wide arc a generates exactly one cycle, and this is the only cycle that arc a appears in. However, the rounding that may affect the values a_C and b_C is not taken into account in this procedure.

5.7.3 Minimum Cycle Basis Algorithms

The polynomial time minimum cycle basis algorithm by Horton (1987) does not necessarily return an integral cycle basis. However, we can still apply the algorithm, and check the resulting cycle basis for integrality. Figure 5.7 shows Horton's minimum

[1] Deo et al. use the term 'minimum-length fundamental cycle set'.

```
Algorithm: MCB.
Input: Undirected graph $U = (N, E)$, edge weights $w_e$.
Output: Minimum cycle basis $B$.

for each pair of nodes $i, j \in N$ do
    Compute a shortest path $P_{ij}$ with respect to the edge weights $w_e$.
for all nodes $k \in N$ do
    for all edges $\{i, j\} \in E$ do
        Cycle $C = P_{ik} + P_{jk} + \{i, j\}$.
        if $C$ is simple then
            $B' \leftarrow C$.
Sort the cycles in $B'$ by non-decreasing weight.
$B = \emptyset$.
for all cycles $C \in B'$ do
    if $|B| < |E| - |N| + 1$ and $C$ is independent of the cycles in $B$ then
        $B \leftarrow C$.
```

Figure 5.7: The minimum cycle basis algorithm MCB (Horton, 1987).

cycle basis (MCB) algorithm. In a first step, the algorithm computes, for each edge $\{i, j\}$, and for each node k, the shortest cycle through $\{i, j\}$ and k. In a second step, the minimum cycle basis is greedily selected from this family of cycles.

Deo et al. (1982) developed several algorithms for the NP-complete problem of constructing a small weight strictly fundamental cycle basis for an undirected graph $U = (N, E)$ with unit edge weights. Their algorithms are Breadth First Search (BFS) algorithms that grow a partial spanning tree $H = (N_H, A_H)$, and follow the general scheme in Figure 5.8. Because the algorithms have been developed for the unit edge weight case, their basic strategy is to add high degree nodes to the spanning tree. In each iteration, a node i^* is selected, and all its adjacent edges $\{i^*, j\} \in E$ are added to the partial tree H, as well as the node j, as long as this does not create a cycle in H. By checking in the for-loop whether j is already in N_H, it is ensured that H remains a tree at all times.

For the selection of the next node i^*, several rules are proposed, which are mostly based on the degree of a node. These rules are the following.

SDS - Static Degree Sort. Consider the oldest node in H which does not have all its neighbors in N_H. SDS selects the highest degree neighbor of this oldest

```
Algorithm: BFS.
Input: Undirected graph $U = (N, E)$.
Output: Short strictly fundamental cycle basis $B$.

$N_H \leftarrow$ highest degree node in $U$.
while $|N_H| < |N|$ do
    Select the next node $i^*$.
    for all nodes $j$ adjacent to $i^*$ do
        if $j \notin N_H$ then
            $N_H \leftarrow j$.
            $E_H \leftarrow \{i^*, j\}$.
for all edges $\{i, j\} \notin E_H$ do
    Compute the unique path $P_{ij}$ between $i$ and $j$ in $U_H = (N_H, E_H)$.
    Cycle $C = \{i, j\} + P_{ij}$.
    $B \leftarrow C$.
```

Figure 5.8: The breadth first search algorithms BFS (Deo et al., 1982).

node, and thus yields a standard BFS tree with respect to the highest degree criterium.

DDS - Dynamic Degree Sort. DDS returns the highest degree node in the partial tree H.

UE - Unexplored Edges. When considering the degree of a node in DDS, it is not 'fair' to count neighbor nodes that are already in the partial tree H. UE takes this into account, and returns the node in H that has the highest degree with respect to its neighbors outside of H.

MBFS - Multipoint Breadth First Search. MBFS selects the highest degree node in U. This may be a node that is not part of H yet, so MBFS may build a forest H.

Egyhazy (1985) and Czech et al. (1993) propose some tie breaking rules for the above node selection rules. Based on these tie breaking rules, Deo et al. (1995) propose some new fine-tuned variants for the above node selection rules. However, since these still do not take into account edge weights, we leave them out of consideration.

Serafini and Ukovich (1989a) first proposed the following rather straightforward algorithm, which takes into account the edge weights w_e. The algorithm constructs

Algorithm: FCB.

Input: Undirected graph $U = (N, E)$, spanning tree $H = (N_H, E_H)$, edge weights w_e.

Output: Short fundamental cycle basis B.

```
Sort the edges in $E_H$ by non-decreasing weight
for all edges $\{i, j\} \notin E_H$ do
    $E = E \setminus \{i, j\}$.
    Compute the shortest path $P_{ij}$ in $U$ with respect to the edge weights $w_e$.
    Cycle $C = \{i, j\} + P_{ij}$.
    $B \leftarrow C$.
```

Figure 5.9: The fundamental cycle basis algorithm FCB (Berger, 2002).

a minimum spanning tree with respect to the edge weights w_e. Next, as in the BFS algorithms above, the non-tree arcs are iteratively added to the tree, and the resulting cycles form the cycle basis. Since it is based on the idea of constructing a minimal spanning tree, we call this the MST algorithm.

The BFS and MST algorithms construct strictly fundamental cycle bases. However, we do not need a *strictly* fundamental cycle basis for formulating the CPF. A regular fundamental cycle basis suffices. For a given spanning tree H, Berger (2002) proposes the algorithm in Figure 5.9 for constructing a fundamental cycle basis (FCB). For each non-tree arc, the FCB algorithm iteratively computes the shortest cycle through that arc in the graph G, rather than the unique cycle in the spanning tree. Since the non-tree arcs are iteratively removed from G, FCB indeed constructs a fundamental cycle basis. Moreover, by considering the arcs in decreasing order of their weights, the heaviest arc appears in only one cycle, the second-heaviest arc in at most two cycles, etc.

Given the structure of the CRTP constraint graph presented in Section 3.8, certain good cycles for the CPF can be listed. These are presented next.

5.7.4 Using the CRTP Constraint Graph Structure

The safety triangles and dwell squares, which were introduced in the previous section, are clearly two classes of good cycles for CRTP graphs. For most of the dwell squares (i, j, j', i'), the cycle periodicity variable $q_{ijj'i'}$ equals zero, and all safety triangles (i, j, k) have bounds $0 \leq q_{ijk} \leq 1$.

Section 5.3 showed that we need an integral cycle basis to formulate the CPF, and Section 5.6 showed that strictly fundamental cycle bases are integral. Therefore, it would be advantageous to construct a spanning tree that generates many safety

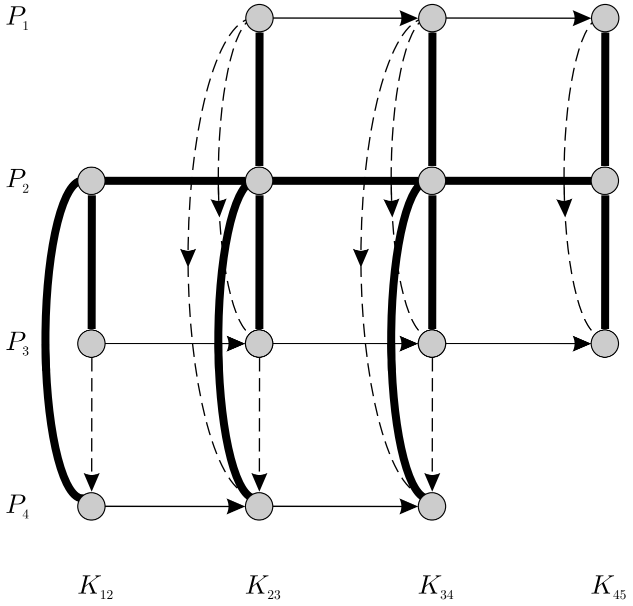


Figure 5.10: Spanning tree H of the graph in Figure 3.14.

triangles and dwell squares.

As an example, consider Figure 5.10, which contains the constraint graph from Figure 3.14. The bold arcs form a spanning tree H of the graph with two properties. First, H contains the longest train path P_t. Second, each safety clique K_n is spanned by a star tree rooted at a node in P_t. In Figure 5.10, the longest path is P_3, and each safety clique is spanned by a star tree rooted at a node in P_3. Because of this structure, the spanning tree H generates only trip squares and safety triangles.

Spanning the nodes in a CRTP graph with a tree H having the two mentioned properties results in a cycle basis that consists for a large part of trip squares and safety triangles. Therefore, we propose to set the arc weights such that a minimum spanning tree generates many dwell squares and safety triangles. Let $L(P_t)$ be the number of trip-dwell arcs in the train path P_t, and let $L_{\max}$ be the maximum of $L(P_t)$ over all trains t. We propose the arc weights below for constructing a minimum spanning tree of the constraint graph G, and call the resulting cycle basis Long Narrow Tree (LNT).

Trip-dwell arcs. For each trip-dwell arc a, set $w_a = 0$.

Safety arcs. Each safety arc a is connected to two train paths P_t and $P_{t'}$.
Set $w_a = L_{\max} - \max\{L(P_t), L(P_{t'})\}$.

Other arcs. For all other arcs a, set $w_a = L_{\max} + u_a - l_a$.

The trip-dwell arc weights ensure that all train paths are in the spanning tree. Safety arcs that are connected to long train paths receive a small arc weight, and are therefore likely to be part of the spanning tree. Finally, all other arcs receive a weight that is larger than the weight of any trip-dwell arc or safety arc.

Since the resulting spanning tree contains all train paths, the tree will generate cycles that consist of many arcs. Most of these arcs are narrow trip-dwell arcs, and therefore the resulting cycles will be long, but still quite narrow. However, since all train paths are contained in the spanning tree, the resulting cycle basis may contain only a moderate number of safety triangles.

Chapter 6

Extensions of the CRTP

This chapter presents some extensions of the CRTP. First, Section 6.1 shows how the choice of a train trip time from a time window can be modeled, thereby relaxing the assumption of fixed train trip times. Section 6.2 describes an extension that allows for more flexibility in modeling connections between multiple trains. In Section 6.3, we show how the capacity of a station can be modeled in both the CPF and the PESP. Next, we describe in Section 6.4 how the minimization of rolling stock utilization can be included into the general objective function. Finally, Section 6.5 describes a heuristic solution method for the CRTP, which is inspired by the iterative way in which planners tend to construct railway timetables in practice.

6.1 Variable Trip Times*

So far, the trip times of trains between consecutive stations were assumed to be fixed. This assumption may be too restrictive, for example, when an instance has no feasible solution. It may then be helpful to increase the solution space by allowing small deviations from the fixed trip times. This idea is illustrated by the time-space diagram in Figure 6.1. Two trains travel along the track AB, a local train L and a faster intercity train I. Suppose that, for reasons that lie outside the track AB, the departure times from A for trains L and I should be d_L and d_I, as indicated in the figure. The fixed trip times of both trains give the solid line for train L and the dotted line for train I, resulting in the infeasible situation of train I overtaking train L at the black dot. However, slowing down train I, giving the dashed line in the time-space diagram, would solve the problem.

This section describes how to incorporate variable trip times in the CRTP model.

*This section is based on Kroon and Peeters (2003).

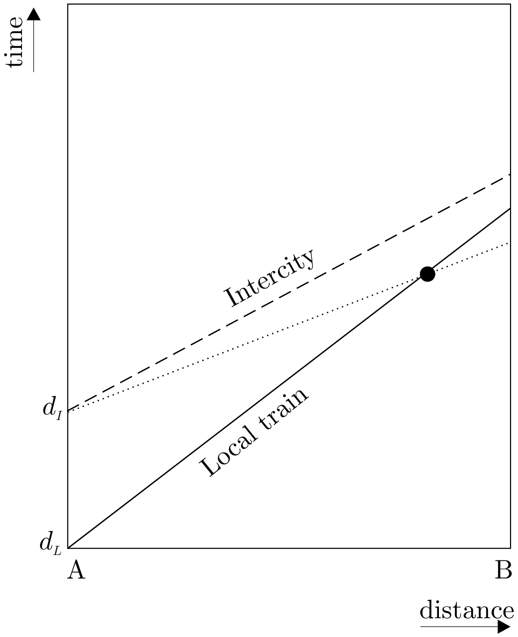


Figure 6.1: Example of a variable trip time

When introducing variable trip times, most of the constraints remain the same as in the case of fixed trip times. Exceptions are the constraints that include the trip times of the trains explicitly: trip time constraints (3.16b), safety constraints (3.16f), which prevent trains from overtaking one another, and safety constraints (3.16g), which prevent collisions on single tracks.

Suppose that a train t has a minimum trip time $\underline{r}_a^t$ and a maximum trip time $\overline{r}_a^t$ on track $a = (n, m)$. The variable trip time can then be expressed using a periodic time window as follows (see also Section 3.7.1),

$$a_m^t - d_n^t \in [\underline{r}_a^t, \overline{r}_a^t]_T. \tag{6.1}$$

This is a straightforward extension of the fixed trip time constraint (3.16b). The overtake and collision constraints (3.16f) and (3.16g) are more complicated, since these constraints include the actually chosen trip time in their time window. And, in the case of variable trip times, the actually chosen trip time is not known beforehand. In the following sections, the auxiliary decision variable ρ^t denotes the actual trip time of train t, as chosen from the time window $[\underline{r}_a^t, \overline{r}_a^t]_T$. Note that ρ^t is only used in the discussion of variable trip times, and is *not* included in the model as a decision variable.

As was mentioned above, the goal of introducing variable trip times into the model is to enlarge the solution space. Still, the trip times are preferred to be as small as possible, that is, the trains are preferred to run at the highest possible speed. This preference for small trip times can be expressed by the objective function described in Section 3.7.1, which favors small trip times.

Below, we first describe the approach by Lindner (2000) for variable trip times, and we explain why we choose to approach the problem differently. Next, we distinguish the following three cases for describing our approach to variable trip times:

(i) General case,

(ii) Disjoint trip time windows: $[\underline{r}_a^t, \overline{r}_a^t] \cap [\underline{r}_a^{t'}, \overline{r}_a^{t'}] = \emptyset$,

(iii) Opposite directions on a single track.

In cases (i) and (ii) two trains t and t' run in the same direction on the same track $a = (n, m)$, whereas in case (iii) the two trains have opposite directions on the single track $a = (n, m)$. For each of these three cases, necessary and sufficient conditions are derived under which the solutions to the extended model are proven to be correct.

6.1.1 Lindner's Approach

Lindner (2000) already proposed a method for including variable trip times into the timetabling model. His approach guarantees that two trains do not overtake each other by using additional variables $\tilde{a}_m^t$ and $\tilde{a}_m^{t'}$. Modulo T these additional variables are equivalent to the original arrival variables a_m^t and $a_m^{t'}$. However, the additional variables are not restricted to the domain $[0, T-1]$, but to the domain $[0, 2T-1]$. Further, Lindner uses the following constraints:

$$\begin{aligned}
&\tilde{a}_m^t - d_n^t \in [\underline{r}_a^t, \overline{r}_a^t], && \tilde{a}_m^{t'} - d_n^{t'} \in [\underline{r}_a^{t'}, \overline{r}_a^{t'}],\\
&d_n^{t'} - d_n^t + Tp \in [h, T-h], && \tilde{a}_m^{t'} - \tilde{a}_m^t + Tq \in [h, T-h],\\
&\tilde{a}_m^t - a_m^t = [0]_T, && \tilde{a}_m^{t'} - a_m^{t'} = [0]_T.
\end{aligned}$$

Here, p and q are integer decision variables, and the last two constraints identify the additional variables $\tilde{a}_m^t$ and $\tilde{a}_m^{t'}$ with the original variables a_m^t and $a_m^{t'}$ modulo T. As before, the parameter h denotes the minimum headway time.

Lemma 6.1. Trains t and t' do not overtake each other if and only if $p = q$.

***Proof*.** Without loss of generality, we may assume $0 \le d_n^t < d_n^{t'} < T$. Obviously, this implies $p = 0$. Now, assume that train t is overtaken by train t'. Then $\tilde{a}_m^{t'} < \tilde{a}_m^t$, which implies $q \neq 0$. The 'only if' part of the proof is similar. ■

Unfortunately, constraints of the type $p = q$ fall outside the general PESP model. Indeed, Lindner defines the JPESP (Joined constraints PESP) as the PESP including this additional type of constraint. In the remainder of this section we describe how variable trip times can be modeled alternatively, using only PESP constraints of the standard form. By using PESP constraints only, the CRTP model remains solvable by any general solution method for the PESP.

6.1.2 General Case

In the general case, there are no restrictions on the trip time windows. In particular, the trip time windows may be overlapping. That means that none of the two trains t and t' is the faster train of the two in all cases. Train t may be overtaken by train t', but it is also possible that train t' is overtaken by train t, depending on the actually chosen trip times ρ^t and $\rho^{t'}$.

Theorem 6.2. The constraints $d_n^{t'} - d_n^t \notin (-h, h)_T$ and $a_m^{t'} - a_m^t \notin (-h, h)_T$ are necessary to guarantee that

(i) trains t and t' do not overtake each other on a, and

(ii) the minimum headway times between trains t and t' at n and m are respected.

If, moreover, $\max\{\overline{r}_a^t - \underline{r}_a^{t'}, \overline{r}_a^{t'} - \underline{r}_a^t\} < 2h$, then these constraints are also sufficient to guarantee conditions (i) and (ii).

Proof. It is obvious that the constraints $d_n^{t'} - d_n^t \notin (-h, h)_T$ and $a_m^{t'} - a_m^t \notin (-h, h)_T$ are necessary and sufficient to guarantee condition (ii) on the minimum headway times. It follows that the constraints are necessary to guarantee conditions (i) and (ii).

Next, we prove that the constraints are also sufficient to guarantee condition (i) if $\max\{\overline{r}_a^t - \underline{r}_a^{t'}, \overline{r}_a^{t'} - \underline{r}_a^t\} < 2h$. To that end, suppose that $d_n^{t'} - d_n^t \notin (-h, h)_T$ and $a_m^{t'} - a_m^t \notin (-h, h)_T$, and that train t is overtaken by train t' between n and m. Obviously, this is only possible if $\rho^{t'} < \rho^t$. We write the constraint $d_n^{t'} - d_n^t \notin (-h, h)_T$ as

$$h \leq d_n^{t'} - d_n^t + Tp \leq T - h, \tag{6.2}$$

for some integer p. The term $d_n^{t'} - d_n^t + Tp$ equals the time between the departure of train t from n and the first departure of train t' from n thereafter. Therefore, the assumption that train t is overtaken means that there exists a real value $0 < \alpha < 1$ with

$$d_n^{t'} - d_n^t + Tp = (\rho^t - \rho^{t'})\alpha. \tag{6.3}$$

Combining equations (6.2) and (6.3) gives

$$h \leq (\rho^t - \rho^{t'})\alpha \leq T - h. \tag{6.4}$$

The same analysis for the constraint $a_m^{t'} - a_m^t \notin (-h, h)_T$ yields

$$h \leq (\rho^t - \rho^{t'})(1 - \alpha) \leq T - h. \tag{6.5}$$

Combining equations (6.4) and (6.5) gives

$$2h \leq (\rho^t - \rho^{t'}).$$

However, this is a contradiction, since $(\rho^t - \rho^{t'}) \leq \overline{r}_a^t - \underline{r}_a^{t'}$ and, by assumption, $(\overline{r}_a^t - \underline{r}_a^{t'}) < 2h$. It follows that condition (i) is not satisfied. In a similar way, it can be shown that train t does not overtake train t'. This demonstrates the sufficiency of the constraints. ■

The condition $\max\{\overline{r}_a^t - \underline{r}_a^{t'}, \overline{r}_a^{t'} - \underline{r}_a^t\} < 2h$ in Theorem 6.2 states that the maximum difference between the trip times of the trains should be less than the sum of the minimum headway times at n and m. The theorem can be interpreted as follows: if train t' would overtake train t, while respecting the minimum headway times, then this would give the situation sketched in Figure 6.2. It then holds that $(\rho^t - \rho^{t'}) \geq 2h$. Therefore, in order to prevent this, we need to require that $\max\{\rho^t - \rho^{t'}\} = \overline{r}_a^t - \underline{r}_a^{t'} < 2h$. The condition $\overline{r}_a^{t'} - \underline{r}_a^t < 2h$ can be interpreted similarly as preventing train t from overtaking train t'.

If the condition $\max\{\overline{r}_a^t - \underline{r}_a^{t'}, \overline{r}_a^{t'} - \underline{r}_a^t\} < 2h$ is not satisfied, that is, if the maximum difference between the trip times of the trains becomes too large, then one train may overtake the other, although the constraints $d_n^{t'} - d_n^t \notin (-h, h)_T$ and $a_m^{t'} - a_m^t \notin (-h, h)_T$ are satisfied. For example, if $\overline{r}_a^t = \underline{r}_a^{t'} + 2h$, then the following solution involving an overtake of train t by train t' would be feasible: $d_n^t = 0$, $d_n^{t'} = h$, $a_m^t = \overline{r}_a^t$, and $a_m^{t'} = \underline{r}_a^{t'} + h = \overline{r}_a^t - h$.

Moreover, if the condition $\max\{\overline{r}_a^t - \underline{r}_a^{t'}, \overline{r}_a^{t'} - \underline{r}_a^t\} < 2h$ is satisfied and the trip time windows are overlapping, that is, $[\underline{r}_a^t, \overline{r}_a^t] \cap [\underline{r}_a^{t'}, \overline{r}_a^{t'}] \neq \emptyset$, then the headway constraints $d_n^{t'} - d_n^t \notin (-h, h)_T$ and $a_m^t - a_m^{t'} \notin (-h, h)_T$ are as strong as possible. Indeed, if $\rho \in [\underline{r}_a^t, \overline{r}_a^t] \cap [\underline{r}_a^{t'}, \overline{r}_a^{t'}]$, then the following solution is feasible: $d_n^t = 0$, $d_n^{t'} = h$, $a_m^t = \rho$ and $a_m^{t'} = h + \rho$. However, the next section shows that the headway constraints can be strengthened if the trip time windows are disjoint.

Finally, whenever the condition $\max\{\overline{r}_a^t - \underline{r}_a^{t'}, \overline{r}_a^{t'} - \underline{r}_a^t\} < 2h$ is not satisfied, one may add a dummy node n' halfway on the track $a = (n, m)$. On the tracks (n, n')

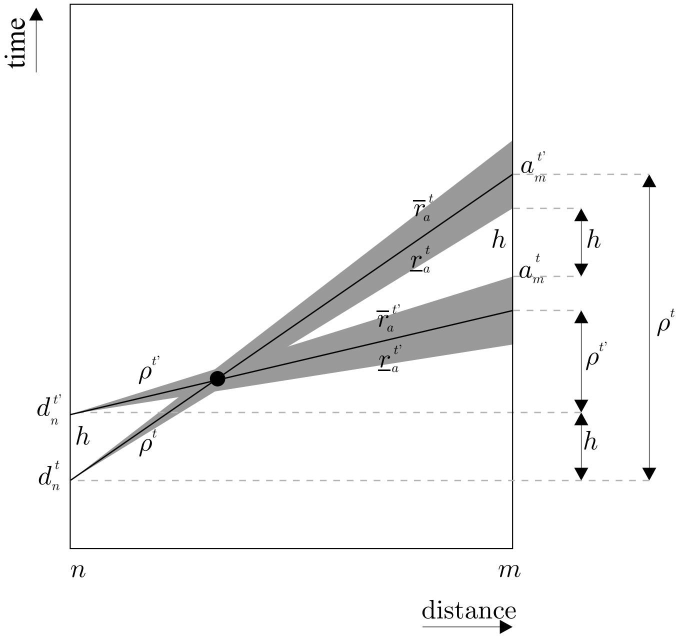


Figure 6.2: Prevent overtaking for variable trip times.

and (n', m), the trip times for trains t and t' should be between $\frac{1}{2}\underline{r}^t_a$ and $\frac{1}{2}\overline{r}^t_a$, and between $\frac{1}{2}\underline{r}^{t'}_a$ and $\frac{1}{2}\overline{r}^{t'}_a$, respectively. Moreover, the headway constraints at n' are to be respected. The condition $\max\{\overline{r}^t_a - \underline{r}^{t'}_a, \overline{r}^{t'}_a - \underline{r}^t_a\} < 2h$ concerning the track (n, m) can thus be replaced by the condition $\max\{\frac{1}{2}(\overline{r}^t_a - \underline{r}^{t'}_a), \frac{1}{2}(\overline{r}^{t'}_a - \underline{r}^t_a)\} < 2h$ for both the tracks (n, n') and (n', m). Obviously, the latter condition is weaker than the original condition. If necessary, this approach may be repeated until the requested condition is satisfied.

6.1.3 Disjoint Trip Time Windows

In the case of disjoint trip time windows, we have $[\underline{r}^t_a, \overline{r}^t_a] \cap [\underline{r}^{t'}_a, \overline{r}^{t'}_a] = \emptyset$. This means that one of the trains is the faster train of the two, whatever the actual trip times of the trains. Without loss of generality, we assume that train t' is the faster train of the two, that is, $\underline{r}^t_a > \overline{r}^{t'}_a$. Since we now only need to prevent train t' from overtaking train t, stronger conditions can be derived than in the general case.

Theorem 6.3. If $\underline{r}^t_a > \overline{r}^{t'}_a$, then the constraints $d^{t'}_n - d^t_n \notin (-h, \underline{r}^t_a - \overline{r}^{t'}_a + h)_T$ and $a^t_m - a^{t'}_m \notin (-h, \underline{r}^t_a - \overline{r}^{t'}_a + h)_T$ are necessary to guarantee that

(i) train t is not overtaken by train t' on a, and (*ii*) the minimum headway times between trains t and t' at n and m are respected.

If, moreover, $(\overline{r}_a^t - \underline{r}_a^t) + (\overline{r}_a^{t'} - \underline{r}_a^{t'}) < 2h + (\underline{r}_a^t - \overline{r}_a^{t'})$, then these constraints are also sufficient to guarantee conditions (*i*) and (*ii*).

Proof. First we prove the necessity of the constraints. If $d_n^{t'} - d_n^t \in (-h, 0]_T$ or $a_m^t - a_m^{t'} \in (-h, 0]_T$, then condition (*ii*) is not satisfied. Now suppose that $d_n^{t'} - d_n^t \in (0, \underline{r}_a^t - \overline{r}_a^{t'} + h)_T$, which means that for some integer p we have

$$0 < d_n^{t'} - d_n^t + Tp < \underline{r}_a^t - \overline{r}_a^{t'} + h. \tag{6.6}$$

Furthermore, the arrival times in m can be expressed as follows in terms of the departure times from n:

$$a_m^t = d_n^t + \rho^t + Tq \qquad \text{and} \qquad a_m^{t'} = d_n^{t'} + \rho^{t'} + Tr,$$

for some integers q and r. Using these relations, we find

$$a_m^{t'} - a_m^t + T(p + q - r) = d_n^{t'} - d_n^t + Tp + (\rho^{t'} - \rho^t). \tag{6.7}$$

Combining equations (6.6) and (6.7) gives

$$a_m^{t'} - a_m^t + T(p + q - r) < h + (\underline{r}_a^t - \rho^t) + (\rho^{t'} - \overline{r}_a^{t'}) \leq h, \tag{6.8}$$

where the latter inequality follows from $\underline{r}_a^t \leq \rho^t$ and $\rho^{t'} \leq \overline{r}_a^{t'}$.

As in the proof of Theorem 6.2, the term $d_n^{t'} - d_n^t + Tp$ in (6.6) equals the time between the departure of train t from n and the first departure of train t' from n thereafter. Now suppose that train t is *not* overtaken by train t' between n and m. That means that the difference in departure times must be greater than the difference in trip times. In other words,

$$d_n^{t'} - d_n^t + Tp + (\rho^{t'} - \rho^t) > 0.$$

This, together with (6.7) and (6.8), gives

$$0 < a_m^{t'} - a_m^t + T(p + q - r) < h. \tag{6.9}$$

However, (6.9) implies that there is insufficient headway time between the trains when they arrive in m. It follows that either (*ii*) is not satisfied, or we must reject the assumption that train t is *not* overtaken by train t', which means that (*i*) is not satisfied. Altogether, the constraint $d_n^{t'} - d_n^t \notin (-h, \underline{r}_a^t - \overline{r}_a^{t'} + h)_T$ is necessary. The necessity of the constraint $a_m^t - a_m^{t'} \notin (-h, \underline{r}_a^t - \overline{r}_a^{t'} + h)_T$ can be demonstrated similarly.

Next, we prove the sufficiency of the constraints if $(\overline{r}_a^t - \underline{r}_a^t) + (\overline{r}_a^{t'} - \underline{r}_a^{t'}) < 2h + (\underline{r}_a^t - \overline{r}_a^{t'})$. Clearly, $d_n^{t'} - d_n^t \notin (-h, \underline{r}_a^t - \overline{r}_a^{t'} + h)_T$ and $a_m^t - a_m^{t'} \notin (-h, \underline{r}_a^t - \overline{r}_a^{t'} + h)_T$ together imply that condition (ii) is satisfied, since $\underline{r}_a^t - \overline{r}_a^{t'} > 0$.

Suppose next that $d_n^{t'} - d_n^t \notin (-h, \underline{r}_a^t - \overline{r}_a^{t'} + h)_T$ and $a_m^t - a_m^{t'} \notin (-h, \underline{r}_a^t - \overline{r}_a^{t'} + h)_T$. The first one of these constraints can also be written as

$$\underline{r}_a^t - \overline{r}_a^{t'} + h \leq d_n^{t'} - d_n^t + Tp \leq T - h, \tag{6.10}$$

for some integer p. Again, the term $d_n^{t'} - d_n^t + Tp$ equals the time between the departure of train t from n and the first departure of train t' from n thereafter. Now, suppose that train t is overtaken by train t' between n and m. Then there exists a real value $0 < \alpha < 1$ with

$$d_n^{t'} - d_n^t + Tp = (\rho^t - \rho^{t'})\alpha. \tag{6.11}$$

Equations (6.10) and (6.11) together give

$$\underline{r}_a^t - \overline{r}_a^{t'} + h \leq (\rho^t - \rho^{t'})\alpha. \tag{6.12}$$

The constraint $a_m^t - a_m^{t'} \notin (0, \underline{r}_a^t - \overline{r}_a^{t'} + h)_T$ can be treated in a similar way, giving

$$a_m^t - a_m^{t'} + Tq = (\rho^t - \rho^{t'})(1 - \alpha),$$

and

$$\underline{r}_a^t - \overline{r}_a^{t'} + h \leq (\rho^t - \rho^{t'})(1 - \alpha). \tag{6.13}$$

Combining equations (6.12) and (6.13) gives

$$2\underline{r}_a^t - 2\overline{r}_a^{t'} + 2h \leq (\rho^t - \rho^{t'}) \leq \overline{r}_a^t - \underline{r}_a^{t'}, \tag{6.14}$$

where the latter inequality follows from $\rho^t \leq \overline{r}_a^t$ and $\underline{r}_a^{t'} \leq \rho^{t'}$. However, (6.14) is a contradiction, since the assumption $(\overline{r}_a^t - \underline{r}_a^t) + (\overline{r}_a^{t'} - \underline{r}_a^{t'}) < 2h + (\underline{r}_a^t - \overline{r}_a^{t'})$ implies $2\underline{r}_a^t - 2\overline{r}_a^{t'} + 2h > \overline{r}_a^t - \underline{r}_a^{t'}$. Thus condition (i) is not satisfied. This demonstrates the sufficiency of the constraints. ■

Theorem 6.3 can be interpreted using Figure 6.3. Suppose that the headway requirements for the arrival at m are respected, that is, $a_m^t - a_m^{t'} \notin (-h, h)_T$. Since

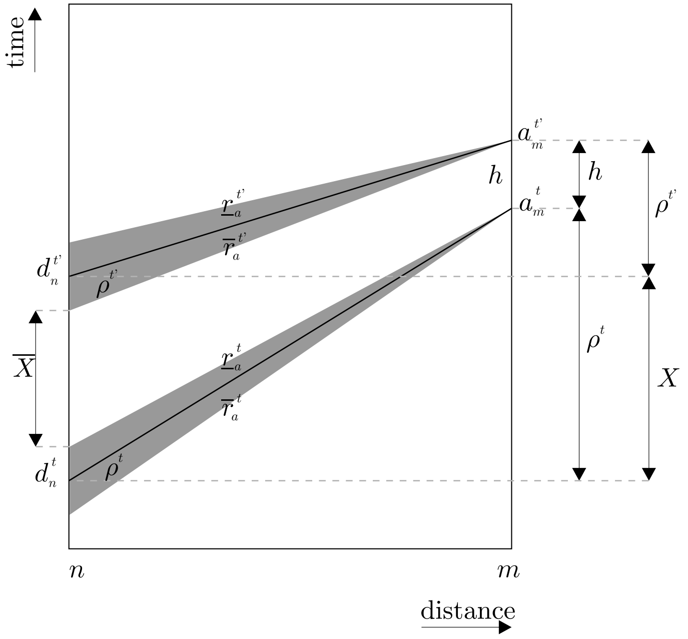


Figure 6.3: Prevent overtaking for disjoint trip time windows.

train t' is always the faster train, all we need to do to prevent overtaking is to ensure that there is enough time between the departures of the two trains from node n. From Figure 6.3 it follows that we must have

$$d_n^{t'} - d_n^t \notin X_T = (-h, \rho^t - \rho^{t'} + h)_T,$$

and that, in the extreme case, this means

$$d_n^{t'} - d_n^t \notin \bar{X}_T = (-h, \underline{r}_a^t - \overline{r}_a^{t'} + h)_T.$$

The other constraint has a similar interpretation. The condition $(\overline{r}_a^t - \underline{r}_a^t) + (\overline{r}_a^{t'} - \underline{r}_a^{t'}) < 2h + (\underline{r}_a^t - \overline{r}_a^{t'})$ can be interpreted as in Figure 6.2, when using the headway time $\bar{h} = \underline{r}_a^t - \overline{r}_a^{t'} + h$ in the condition $\overline{r}_a^t - \underline{r}_a^{t'} < 2\bar{h}$ for that figure.

Note that Theorem 6.3 is stronger than Theorem 6.2. Indeed, by assumption $\underline{r}_a^t > \overline{r}_a^{t'}$. Therefore, the constraints $d_n^{t'} - d_n^t \notin (-h, h + \underline{r}_a^t - \overline{r}_a^{t'})_T$ and $a_m^t - a_m^{t'} \notin (-h, h + \underline{r}_a^t - \overline{r}_a^{t'})_T$ are stronger than the constraints in Theorem 6.2, and the condition $(\overline{r}_a^t - \underline{r}_a^t) + (\overline{r}_a^{t'} - \underline{r}_a^{t'}) < 2h + (\underline{r}_a^t - \overline{r}_a^{t'})$ is weaker than the condition in Theorem 6.2. With examples similar to those at the end of the previous section, one can show that relaxing the condition $(\overline{r}_a^t - \underline{r}_a^t) + (\overline{r}_a^{t'} - \underline{r}_a^{t'}) < 2h + (\underline{r}_a^t - \overline{r}_a^{t'})$ allows for solutions in which trains overtake each other, and that the constraints in Theorem 6.3 are as strong as possible.

6.1.4 Opposite Directions on a Single Track

In this section we describe the situation where two trains run in opposite directions on a single track a. Train t runs from n to m, and train t' runs from m to n. Since a is a single track, meets and passes of the trains must take place at n or m. Moreover, both in n and in m a minimum headway time of h minutes is to be respected.

Theorem 6.4. The constraints $a_n^{t'} - d_n^t \notin (-h, \underline{r}_a^t + \underline{r}_a^{t'} + h)_T$ and $a_m^t - d_m^{t'} \notin (-h, \underline{r}_a^t + \underline{r}_a^{t'} + h)_T$ are necessary to guarantee that

- (i) the trains do not meet each other on a, and
- (ii) the minimum headway times between trains t and t' at n and m are respected.

If, moreover, $(\overline{r}_a^t - \underline{r}_a^t) + (\overline{r}_a^{t'} - \underline{r}_a^{t'}) < 2h + (\underline{r}_a^t + \underline{r}_a^{t'})$, then these constraints are also sufficient to guarantee conditions (i) and (ii).

***Proof*.** First we prove the necessity of the constraints. If $a_n^{t'} - d_n^t \in (-h, 0]_T$ or $a_m^t - d_m^{t'} \in (-h, 0]_T$, then condition (ii) is not satisfied. Furthermore, if $a_n^{t'} - d_n^t \in (0, \underline{r}_a^t + \underline{r}_a^{t'} + h)_T$, then there exists an integer p such that

$$0 < a_n^{t'} - d_n^t + Tp < \underline{r}_a^t + \underline{r}_a^{t'} + h. \tag{6.15}$$

Again, the term $a_n^{t'} - d_n^t + Tp$ represents the time between the departure of train t from n and the first arrival of train t' at n thereafter. Now suppose that train t does *not* meet train t' somewhere between nodes n and m. Then this implies

$$a_n^{t'} - d_n^t + Tp > \rho^t + \rho^{t'}. \tag{6.16}$$

Furthermore, a_m^t and $d_n^{t'}$ can be expressed as follows in terms of d_n^t and $a_m^{t'}$:

$$a_m^t = d_n^t + \rho^t + Tq \qquad d_m^{t'} = a_n^{t'} - \rho^{t'} + Tr.$$

Using these relations, we find

$$a_m^t - d_m^{t'} - T(p + q - r) = -(a_n^{t'} - d_n^t + Tp) + \rho^t + \rho^{t'}. \tag{6.17}$$

Combining equations (6.15) and (6.17) gives

$$a_m^t - d_m^{t'} - T(p+q-r) > -h + (\rho^t - \underline{r}_a^t) + (\rho^{t'} - \underline{r}_a^{t'}) \geq -h, \tag{6.18}$$

where the last inequality follows from $\underline{r}_a^t \leq \rho^t$ and $\underline{r}_a^{t'} \leq \rho^{t'}$. From (6.16) and (6.17) we get

$$a_m^t - d_m^{t'} - T(p+q-r) < 0. \tag{6.19}$$

Now, equations (6.18) and (6.19) together give

$$0 < d_m^{t'} - a_m^t + T(p+q-r) < h.$$

However, this means that there is insufficient headway time between the arrival of train t at m and the departure of train t' from m. It follows that either (i) is not satisfied, or we must reject the assumption that the trains do *not* meet underway between n and m, which means violating (ii). Thus the constraint $a_n^{t'} - d_n^t \notin (-h, \underline{r}_a^t + \underline{r}_a^{t'} + h)_T$ is necessary. The necessity of the constraint $a_m^t - d_m^{t'} \notin (-h, \underline{r}_a^t + \underline{r}_a^{t'} + h)_T$ can be demonstrated similarly.

Next, we prove the sufficiency of the constraints if $(\overline{r}_a^t - \underline{r}_a^t) + (\overline{r}_a^{t'} - \underline{r}_a^{t'}) < 2h + \underline{r}_a^t + \underline{r}_a^{t'}$. To that end, suppose that $a_n^{t'} - d_n^t \notin (-h, \underline{r}_a^t + \underline{r}_a^{t'} + h)_T$ and $a_m^t - d_m^{t'} \notin (-h, \underline{r}_a^t + \underline{r}_a^{t'} + h)_T$. Then, since $\underline{r}_a^t + \underline{r}_a^{t'} > 0$, also $a_n^{t'} - d_n^t \notin (-h, h)_T$ and $a_m^t - d_m^{t'} \notin (-h, h)_T$. Thus, obviously, condition (ii) is satisfied.

The constraint $a_n^{t'} - d_n^t \notin (-h, \underline{r}_a^t + \underline{r}_a^{t'} + h)_T$ can also be written as

$$\underline{r}_a^t + \underline{r}_a^{t'} + h \leq a_n^{t'} - d_n^t + Tp \leq T - h, \tag{6.20}$$

for some integer p, with $(a_n^{t'} - d_n^t + Tp)$ equal to the time between the departure of train t from n and the first arrival of train t' at n thereafter. Now, suppose that train t meets train t' underway between n and m. Then there exists a real value α with $0 < \alpha < 1$ such that

$$a_n^{t'} - d_n^t + Tp = (\rho^t + \rho^{t'})\alpha. \tag{6.21}$$

Combining equations (6.20) and (6.21) gives

$$\underline{r}_a^t + \underline{r}_a^{t'} + h \leq (\rho^t + \rho^{t'})\alpha \leq T - h. \tag{6.22}$$

The constraint $a_m^t - d_m^{t'} \notin (-h, \underline{r}_a^t + \underline{r}_a^{t'} + h)_T$ can be treated in a similar way, giving

$$a_m^t - d_m^{t'} + Tq = (\rho^t + \rho^{t'})(1 - \alpha),$$

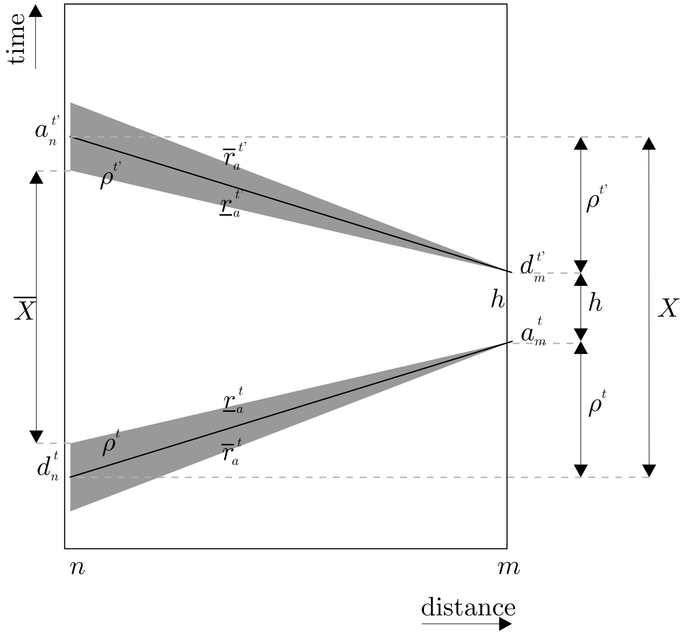


Figure 6.4: Prevent collisions on a single track.

and

$$\underline{r}_a^t + \underline{r}_a^{t'} + h \leq (\rho^t + \rho^{t'})(1 - \alpha) \leq T - h. \tag{6.23}$$

Adding equations (6.22) and (6.23) gives

$$2\underline{r}_a^t + 2\underline{r}_a^{t'} + 2h \leq (\rho^t + \rho^{t'}) \leq \overline{r}_a^t + \overline{r}_a^{t'}. \tag{6.24}$$

However, this is a contradiction, since the assumption $(\overline{r}_a^t - \underline{r}_a^t) + (\overline{r}_a^{t'} - \underline{r}_a^{t'}) < 2h + (\underline{r}_a^t + \underline{r}_a^{t'})$ implies $2\underline{r}_a^t + 2\underline{r}_a^{t'} + 2h > \overline{r}_a^t + \overline{r}_a^{t'}$. It follows that condition (i) is not satisfied. This demonstrates the sufficiency of the constraints. ■

Theorem 6.4 has the following interpretation (see Figure 6.4). Suppose that $a_m^t - d_m^{t'} \notin (-h, \underline{r}_a^t + \underline{r}_a^{t'} + h)_T$. Clearly, the headway times at m are respected. To prevent the trains from meeting between n and m, we must ensure that there is enough time between d_n^t and $a_n^{t'}$ at n. From the figure, it follows that this means

$$a_n^{t'} - d_n^t \notin X_T = (-h, \rho^t + \rho^{t'} + h)_T,$$

which in the extreme case gives

$$a_n^{t'} - d_n^t \notin \bar{X}_T = (-h, \underline{r}_a^t + \underline{r}_a^{t'} + h)_T.$$

The other constraint follows similarly. The condition $(\overline{r}_a^t - \underline{r}_a^t) + (\overline{r}_a^{t'} - \underline{r}_a^{t'}) < 2h + (\underline{r}_a^t + \underline{r}_a^{t'})$ can be derived by considering the worst case that satisfies the constraints $a_n^{t'} - d_n^t \notin (-h, \underline{r}_a^t + \underline{r}_a^{t'} + h)_T$ and $a_m^t - d_m^{t'} \notin (-h, \underline{r}_a^t + \underline{r}_a^{t'} + h)_T$. Here, the term *worst* means that both trains run at the lowest possible speed, and that one of the constraints is binding, thereby creating a situation in which a meet is most likely. Substituting $\rho^t = \overline{r}_a^t$, $\rho^{t'} = \overline{r}_a^{t'}$, and $a_n^{t'} - d_n^t = \underline{r}_a^t + \underline{r}_a^{t'} + h$ into the remaining constraint $a_m^t - d_m^{t'} \notin (-h, \underline{r}_a^t + \underline{r}_a^{t'} + h)_T$ then gives the condition.

Again, it can be shown that relaxing the condition $(\overline{r}_a^t - \underline{r}_a^t) + (\overline{r}_a^{t'} - \underline{r}_a^{t'}) < 2h + (\underline{r}_a^t + \underline{r}_a^{t'})$ may result in infeasible solutions, and that, given this sufficiency condition, the constraints in the theorem are as strong as possible.

6.1.5 Practical experiences

In the general case, the additional flexibility that is created by the variable trip times is rather small. Typically, the minimum headway time equals 3 minutes, and therefore the sufficiency condition for the general case states that the maximum difference between the trip times of the two trains should be less than 6 minutes. More flexibility may be created by adding dummy nodes, as was indicated in Section 6.1.2. In the case of disjoint trip time intervals, the *total* trip time flexibility of both trains together should not exceed 6 minutes plus the gap between the two trip time intervals, which usually gives considerable flexibility. In the case of opposite directions on a single track, there is even more flexibility for the trip times. This is due to the term $(\underline{r}_a^t + \underline{r}_a^{t'})$ with a *plus* sign in the sufficiency condition instead of the term $(\underline{r}_a^t - \overline{r}_a^{t'})$.

A prototype version of the variable trip time model has been implemented in the decision support system DONS (for a more detailed description of DONS, see Section 2.5). Using DONS, a planner may first try to construct a timetable based on fixed trip times. If a feasible timetable does not exist, then the fixed trip times can be made variable to some extent. After the variable trip times have been defined, the system checks the necessary and sufficient conditions and generates the relevant constraints. When necessary, dummy nodes may be added in order to satisfy the sufficiency conditions. Now, a feasible timetable may exist due to the extension of the solution space. Although this implementation is still rather limited, planners at NSR and Railned are quite satisfied with the extension.

6.2 Flexible Connections

Section 3.4.3 described how a connection between two specific trains can be modeled. However, it may be too restrictive to define beforehand which two trains should connect to each other. As an example, suppose that NSR decides to operate the 1900 train from the examples in Section 3.4 with frequency two, so twice per hour. We denote the two involved trains by $t_{1900,1}$ and $t_{1900,2}$. It is then desirable to have the timetabling model choose which one of the two trains $t_{1900,1}$ and $t_{1900,2}$ should offer the connection with the 800 train, rather than fixing this choice beforehand.

This section describes how to incorporate such flexible connections into the CRTP model. First, we discuss the above described situation of a connection between a train line with frequency two and a train line with frequency one. Next, the idea is generalized to the connection between a train line with frequency k, and a single train with frequency one. Finally, we present the connection between two train lines with frequency two, and discuss how to generalize the presented ideas to two train lines with general frequencies.

We use the following notation for the trains involved in the connection. The connecting trains are denoted by $t_1, \ldots, t_k$, and the feeding trains by $t'_1, \ldots, t'_{k'}$. The departure times of the connecting trains are denoted by $d_1, \ldots, d_k$, and the arrival times of the feeding trains by $a_1, \ldots, a_{k'}$. Throughout this section, we assume the time window for the connection between the trains $t_1, \ldots, t_k$ and $t'_1, \ldots, t'_{k'}$ to be equal to $[\underline{c}, \overline{c}]_T$. Further, the general synchronization constraints in Section 3.4.5 play an important part in our analysis of flexible connections.

6.2.1 Two-to-one Flexible Connections

This subsection considers the case $k = 2, k' = 1$, that is, the case of two connecting trains t_1 and t_2, and one feeding train t'. One of the trains t_1 and t_2 must offer a connection with transfer time window $[\underline{c}, \overline{c}]$ with the feeding train t'. Since there are only two connecting trains, the synchronization time window for t_1 and t_2 is denoted by $[\underline{s}, \overline{s}]_T$. The arrival time of the single feeding train t' is simply denoted by a. The following theorem describes how to model the connection between the trains t_1, t_2 and t'.

Theorem 6.5. Suppose that $\overline{c} - \underline{c} < \underline{s}$. If

$$d_i - a \in [\underline{c}, \overline{s} + \overline{c}]_T, \text{ and} \tag{6.25a}$$

$$d_i - a \in [\underline{s} + \underline{c}, \overline{c} + T]_T, \tag{6.25b}$$

for $i = 1, 2$, then exactly one of the trains t_1, t_2 provides a $[\underline{c}, \overline{c}]$ connection with the train t'.

Proof. Lemma 4.9 implies that the constraints (6.25) are equivalent to

$$d_i - a \in [\underline{c}, \overline{c}]_T, \text{ or} \tag{6.26a}$$

$$d_i - a \in [\underline{s} + \underline{c}, \overline{s} + \overline{c}]_T, \tag{6.26b}$$

for $i = 1, 2$. Recall that equality (3.10) states that $\underline{s} + \overline{s} = T$. This, together with the condition $\underline{s} > \overline{c} - \underline{c}$, ensures that the two time windows in (6.26) are disjoint.

If constraint (6.26a) is satisfied for $i = 1$, then t_1 clearly provides a connection with t'. Moreover, because of the synchronization constraint between t_1 and t_2, we know that in this case $d_2 - a \in [\underline{s} + \underline{c}, \overline{s} + \overline{c}]_T$, which means that constraint (6.26b) is automatically satisfied for $i = 2$.

The same argument holds for the case in which constraint (6.26a) is satisfied for $i = 2$. Then, train t_2 provides the connection, and constraint (6.26b) is automatically satisfied for $i = 1$.

The above implies that constraint (6.26a) can not be satisfied for both $i = 1$ and $i = 2$. It follows that exactly one of the trains t_1 and t_2 provides a $[\underline{c}, \overline{c}]$ connection with the train t'. ■

Recall from Section 3.4.6 that $\underline{s} = \frac{T}{2} - \delta$. Typically, for $T = 60$, the value of δ lies between zero and five minutes, and one hardly ever defines passenger transfer connections with a flexibility of more than 25 minutes. Therefore, the condition $\overline{c} - \underline{c} < \underline{s}$ poses no restriction in practice on defining a two-to-one flexible passenger transfer connection. However, when defining rolling stock connections, the condition may play a role.

Finally, note that a theorem similar to Theorem 6.5 can be stated for the opposite situation with $k = 1, k' = 2$.

6.2.2 Many-to-one Flexible Connections

In this subsection, we investigate the case of k connecting trains $t_1, \ldots, t_k$, and one feeding train t', that is, the case of a general k, and of $k' = 1$. One of the trains $t_1, \ldots, t_k$ must offer a connection of $\underline{c}$ to $\overline{c}$ minutes with the feeding train t'. The arrival time of the single feeding train t' is again denoted by a. The following generalization of Theorem 6.5 describes how the connection between the trains $t_1, \ldots, t_k$, and t' can be modeled. The theorem uses the general synchronization time windows $[\underline{s}_{ij}, \overline{s}_{ij}]$ described in Section 3.4.5.

Theorem 6.6. Suppose that $\overline{c} - \underline{c} < \frac{T}{k} - \delta$. If

$$d_i - a \in [\underline{c}, \overline{c} + \overline{s}_{i,i-1}]_T, \quad \text{and} \tag{6.27a}$$

$$d_i - a \in [\underline{c} + \underline{s}_{i,i+1}, \overline{c} + T]_T, \tag{6.27b}$$

for $i = 1, \ldots, k$, then exactly one of the trains $t_1, \ldots, t_k$ provides a $[\underline{c}, \overline{c}]_T$ connection with the train t'.

Proof. As in Theorem 6.5, we use Lemma 4.9, which states that the constraints (6.27) are equivalent to the following set of 'or' constraints:

$$d_i - a \in [\underline{c}, \overline{c}]_T, \quad \text{or} \tag{6.28a}$$

$$d_i - a \in [\underline{c} + \underline{s}_{i,i+1}, \overline{c} + \overline{s}_{i,i-1}]_T, \tag{6.28b}$$

for $i = 1, \ldots, k$. This equivalence only holds if the time windows in (6.28) are disjoint. This means that the following must hold for each $i = 1, \ldots, k$:

$$\overline{c} < \underline{c} + \underline{s}_{i,i+1}, \quad \text{and} \tag{6.29a}$$

$$\overline{c} + \overline{s}_{i,i-1} < \underline{c} + T. \tag{6.29b}$$

Using the frequency time window bounds (3.8), one can check that both conditions (6.29) yield $\overline{c} - \underline{c} < \frac{T}{k} - \delta$, which was our assumption.

Suppose that an arbitrary train t_j provides a $[\underline{c}, \overline{c}]_T$ connection with train t', so constraint (6.28a) is satisfied for j. Next, consider the synchronization constraints (3.8) in Section 3.4.5, their symmetrical counter parts (3.11), and the equality $\underline{s}_{ij} + \overline{s}_{ij} = T$ (3.10). Together, these yield

$$d_j - d_i \in [\underline{s}_{ij}, \overline{s}_{ij}]_T \text{ for all } i \neq j.$$

These constraints imply that the following set of constraints is automatically satisfied:

$$\begin{aligned}
d_j - a &\in [\underline{c} + \underline{s}_{j1}, \overline{c} + \overline{s}_{j1}]_T, \\
&\vdots \\
d_{j-1} - a &\in [\underline{c} + \underline{s}_{j,j-1}, \overline{c} + \overline{s}_{j,j-1}]_T, \\
d_{j+1} - a &\in [\underline{c} + \underline{s}_{j,j+1}, \overline{c} + \overline{s}_{j,j+1}]_T, \\
&\vdots \\
d_k - a &\in [\underline{c} + \underline{s}_{jk}, \overline{c} + \overline{s}_{jk}]_T.
\end{aligned}$$

So, for each $i = 1, \ldots, k, i \neq j$, we have

$$d_i - a \in [\underline{c} + \underline{s}_{j,j+1}, \overline{c} + \overline{s}_{j,j-1}]_T.$$

Because of the constraints (6.28), this means that none the trains t_i, with $i \neq j$, provides a connection with train t'.

Moreover, because of synchronization constraints (3.8), it is impossible that constraint (6.28b) is satisfied *for each* $i = 1, \ldots, k$. It follows that at least one of the trains t must satisfy (6.28a), and therefore that exactly one train provides the connection with t'. ■

The condition $\overline{c} - \underline{c} < \frac{T}{k} - \delta$ usually poses no difficulties in practice when considering transfer connections. For rolling stock connections, it may impose some restrictions. However, there are not many practical situations where a rolling stock connection is defined from one of more than two trains t to a single train t'.

As in the previous section, a similar theorem can be derived for the situation with general k' and $k = 1$.

6.2.3 Two-to-two Flexible Connections

The analysis for the situation with general k and k' is quite complex, although the basic idea is the same as in the previous two subsections. Therefore, we consider the case $k = k' = 2$, with two connecting trains t_1 and t_2, and two feeding trains t'_1 and t'_2. We have the following two synchronization constraints in this case:

$$d_2 - d_1 \in [\underline{s}, \overline{s}]_T, \tag{6.30a}$$

$$a_2 - a_1 \in [\underline{s}', \overline{s}']_T. \tag{6.30b}$$

Since these are the only synchronization constraints, we omit the indices on the synchronization time window bounds.

In the case $k = k' = 2$, *two* connections with time window $[\underline{c}, \overline{c}]$ should be established between the trains t and t'. These two connections can be realized in two ways:

(i) Train t_1 connects with train t'_1, and train t_2 connects with train t'_2.

(ii) Train t_1 connects with train t'_2, and train t_2 connects with train t'_1.

The problem is to decide how the four trains should connect to one another. Clearly, this problem is modeled by the following constraints:

$$d_1 - a_1 \in [\underline{c}, \overline{c}]_T \text{ and } d_2 - a_2 \in [\underline{c}, \overline{c}]_T, \tag{6.31a}$$

$$\text{or}$$

$$d_1 - a_2 \in [\underline{c}, \overline{c}]_T \text{ and } d_2 - a_1 \in [\underline{c}, \overline{c}]_T. \tag{6.31b}$$

Here, constraints (6.31a) correspond to connection (i), and constraints (6.31b) to connection (ii). Suppose that train t_1 connects to t'_1, which means that the leftmost constraint in (6.31a) is satisfied. Using the synchronization constraint (6.30b), and the fact that $\underline{s}' + \overline{s}' = T$, we see that in this case the following constraint is automatically satisfied:

$$d_1 - a_2 \in [\underline{c} + \underline{s}', \overline{c} + \overline{s}']_T. \tag{6.32}$$

Similarly, if train t_1 connects with train t'_2, then the leftmost of constraints (6.31b) is satisfied, and the synchronization constraint between t'_1 and t'_2 yields that the constraint $d_1 - a_1 \in [\underline{c}+\underline{s}', \overline{c}+\overline{s}']_T$ is automatically satisfied. Using a similar argument for the two rightmost constraints in (6.31), we derive that (6.31) is equivalent to

$$d_1 - a_1 \in [\underline{c}, \overline{c}]_T \text{ or } d_1 - a_1 \in [\underline{c} + \underline{s}', \overline{c} + \overline{s}']_T, \text{ and} \tag{6.33a}$$

$$d_1 - a_2 \in [\underline{c}, \overline{c}]_T \text{ or } d_1 - a_2 \in [\underline{c} + \underline{s}', \overline{c} + \overline{s}']_T, \text{ and} \tag{6.33b}$$

$$d_2 - a_1 \in [\underline{c}, \overline{c}]_T \text{ or } d_2 - a_1 \in [\underline{c} + \underline{s}', \overline{c} + \overline{s}']_T, \text{ and} \tag{6.33c}$$

$$d_2 - a_2 \in [\underline{c}, \overline{c}]_T \text{ or } d_2 - a_2 \in [\underline{c} + \underline{s}', \overline{c} + \overline{s}']_T. \tag{6.33d}$$

The only problem is that the situation in which both trains t_1 and t_2 connect with the *same* train in t' is a solution to (6.33). However, this solution is prevented by the synchronization constraint (6.30a) between t_1 and t_2.

Finally, we apply Lemma 4.9 to the system (6.33), which gives

$$d_1 - a_1 \in [\underline{c}, \overline{c} + \overline{s}']_T \text{ and } d_1 - a_1 \in [\underline{c} + \underline{s}', \overline{c} + T]_T, \text{ and} \tag{6.34a}$$

$$d_1 - a_2 \in [\underline{c}, \overline{c} + \overline{s}']_T \text{ and } d_1 - a_2 \in [\underline{c} + \underline{s}', \overline{c} + T]_T, \text{ and} \tag{6.34b}$$

$$d_2 - a_1 \in [\underline{c}, \overline{c} + \overline{s}']_T \text{ and } d_2 - a_1 \in [\underline{c} + \underline{s}', \overline{c} + T]_T, \text{ and} \tag{6.34c}$$

$$d_2 - a_2 \in [\underline{c}, \overline{c} + \overline{s}']_T \text{ and } d_2 - a_2 \in [\underline{c} + \underline{s}', \overline{c} + T]_T. \tag{6.34d}$$

Note that the values $\underline{c}$, $\overline{c}$, $\underline{s}'$, and $\overline{s}'$ must be such that the time windows in (6.33) are disjoint, since else we may not apply Lemma 4.9.

Using both the synchronization constraints of t and t' is the key idea for defining many-to-many flexible connection constraints. However, for more than two-to-two connections, the analysis becomes quite difficult, and so does the check of the disjoint time window condition, under which Lemma 4.9 can be applied. Therefore, we do not further describe many-to-many flexible connections. For a more extensive analysis of many-to-many flexible connections, we refer to Zuidwijk and Kroon (2000).

6.3 Modeling Station Capacity in the CRTP

The CRTP so far considered stations to be black boxes. It returns train arrival and departure times, and leaves the construction of feasible platform assignments and routings through the stations to be carried out in a later phase (see Zwaneveld et al., 1996, Kroon et al., 1997, Zwaneveld, 1997). This section describes how the capacity of a station can be modeled to some extent. However, our station capacity model requires constraints that fall outside the PESP formulation (4.2). The results in this section are also more generally applicable to model the capacity of a general node in the railway network.

Let U_n be the capacity of node $n \in \mathcal{N}$, expressed as the maximum number of trains that can be in node n at the same time instant. Usually, U_n represents the number of platforms in a station, or, more generally, the number of trains a station can handle concurrently. This section describes how to place an upper bound U_n on the number of trains that is concurrently present at a station. Throughout the section, we assume that, when two trains are present at a station n at the same time instant, then each of the two trains consumes one unit of station capacity. So, for a station n and two trains t and t', both t and t' consume one unit of station capacity if $d_n^t = a_n^{t'}$. When U_n represents the number of platforms in station n, this assumption means that each platform can host at most one train, and that each train needs a platform.

In practice, one often sees that some platforms are dedicated to certain groups or types of trains. For example, some platforms may be used only for trains heading in a certain direction, or for intercity trains or local trains. Using the modeling procedure in this section, one can also limit the number of trains for each platform group.

Below, we first describe a station capacity extension for the CPF. Next, a station capacity extension for the PESP is presented.

6.3.1 Station Capacity in the CPF

A *station square* (i, j, k, l) in a station graph G_n^S is a cycle (i, j, k, l) consisting of two dwell arcs (i, j) and (k, l), and of the two arcs (k, j) and (i, l), see Figure 6.5(a). Recall that the trains t and t' have dwell time windows $[\underline{d}_n^t, \overline{d}_n^t]$ and $[\underline{d}_n^{t'}, \overline{d}_n^{t'}]$, respectively. If the arcs (i, l) and (k, j) are not defined in G_n^S, we may add them with the

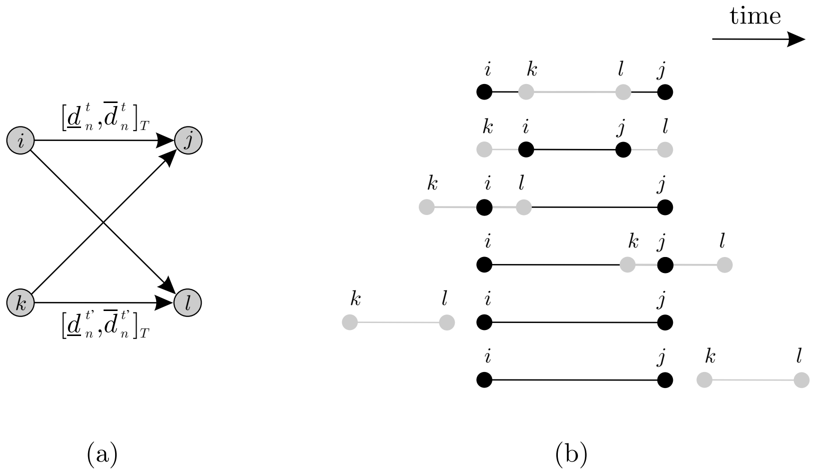


Figure 6.5: (a) Station square (i, j, k, l), (b) Possible dwelling sequences for the trains t and t'.

trivial time window $[0, T-1]$. The following lemma is the basis of our station capacity extension. It allows for determining when two trains are present in a station at the same time.

Lemma 6.7. Consider a pair of trains t and t' dwelling at a station n, with dwell arcs (i, j) and (k, l), respectively. Assume that $\overline{d}^t_n, \overline{d}^{t'}_n < \frac{1}{2}T$. If the arcs (i, l) and (k, j) are not defined, add them with the trivial time window $[0, T-1]$. Then the trains t and t' are concurrently present in station n if and only if $q_C = 0$ for the cycle $C = (i, j, k, l)$.

***Proof*.** First, assume that the arcs (i, l) and (k, j) both have time window $[0, T-1]$, and compute the cycle bounds on C:

$$a_C = \left\lceil \frac{1}{T} \left(\underline{d}^t_n + \underline{d}^{t'}_n - 2(T-1) \right) \right\rceil = -2 + \left\lceil \frac{\underline{d}^t_n + \underline{d}^{t'}_n + 2}{T} \right\rceil ,$$

$$b_C = \left\lfloor \frac{1}{T} \left(\overline{d}^t_n + \overline{d}^{t'}_n \right) \right\rfloor = \left\lfloor \frac{\overline{d}^t_n + \overline{d}^{t'}_n}{T} \right\rfloor .$$

By our general assumption on time windows, $\underline{d}^t_n, \underline{d}^{t'}_n \geq 0$. Using this, together with the assumption $\overline{d}^t_n, \overline{d}^{t'}_n < \frac{1}{2}T$, we obtain the cycle bounds $a_C = -1, b_C = 0$, so $q_C \in \{-1, 0\}$.

Recall that the events i, j, k, l correspond to railway events as follows:

$$\begin{aligned} i &= a_n^t, & k &= a_n^{t'}, \\ j &= d_n^t, & l &= d_n^{t'}. \end{aligned}$$

Figure 6.5(b) illustrates the possible cyclic sequences for the dwelling of the trains t and t' at n. This figure, together with the above railway event interpretation of the nodes, shows that in each possible cyclic sequence, except for the cyclic sequence $S = i \rightarrow j \rightarrow k \rightarrow l$, the trains t and t' are concurrently present at station n. Only for the cyclic sequence S, it is not clear beforehand whether the two trains will be concurrently present at the station or not. Moreover, Lemma 5.11 yields that the cyclic sequence S takes place if and only if $q_C = -1$.

In the cyclic sequence S, the trains are present in the station at the same time instant when events i and l, or j and k, or all four events, take place at the same time instant. Consider this situation where either i and l, or j and k, or all four events, take place concurrently. The cycle periodicity constraint for C reads

$$x_{ij} - x_{kj} + x_{kl} - x_{il} = Tq_C. \tag{6.35}$$

If i takes place at the same time as l, then $x_{il} = 0$, and (6.35) becomes $x_{ij} - x_{kj} + x_{kl} = Tq_C$. Using $\overline{d}_n^t, \overline{d}_n^{t'} < \frac{1}{2}T$ for x_{ij} and x_{kl}, and the time window $[0, T-1]$ for x_{kj}, we obtain $-T + 1 \leq x_{ij} - x_{kj} + x_{kl} < T$. So, if l takes place at the same time as i, then $q_C = 0$. Analogously, $q_C = 0$ when j takes place at the same time as k, and when all four events take place at the same time.

So, the cyclic sequence S takes place if and only if $q_C = -1$, and, moreover, within S it is impossible for any of the four events to take place at the same time. Therefore, the trains t and t' are *not* concurrently present in the station if and only if $q_C = -1$. Thus, it follows that t and t' *are* concurrently present in n if and only if $q_C = 0$. Finally, it is easily seen that the proof also holds when the arcs (i, l) and (k, j) have narrower time windows than $[0, T-1]$. ■

The assumption on the upper bound of the dwell time windows is quite reasonable in practice, since trains hardly ever dwell at a station for more than half the cycle time.

Theorem 6.8 below states how station capacity can be modeled in the CPF. For a station n with capacity U_n, we consider each possible set of $U_n + 1$ trains that dwell at the station. If t_n trains dwell at the station, then there are $\binom{t_n}{U_n+1}$ such sets. For such a set of $U_n + 1$ trains, let S be the set of all the station squares between the dwell arcs of those trains. Note that S contains a total of $\frac{1}{2}U_n(U_n + 1)$ station squares. Further, let S_n be the set consisting of all such sets S for the station n.

Theorem 6.8. Consider a station n with station graph G_n^S. For each pair of dwell arcs (i,j) and (k,l), if the arcs (i,l) and (k,j) are not defined in G_n^S, add them with the trivial time window $[0, T-1]$. Assume that for all trains t dwelling at the station, $\overline{d}_n^t < \frac{1}{2}T$. Then the following constraints ensure that the station capacity U_n is respected:

$$\sum_{C \in S} q_C \leq -1 \text{ for all } S \in S_n. \tag{6.36}$$

Proof. By Lemma 6.7, two trains involved in a dwell square C are present at the station concurrently if and only if $q_C = 0$, and otherwise $q_C = -1$. Consider some set of $U_n + 1$ trains, and the corresponding set S. If these $U_n + 1$ trains are all present at the station concurrently, then all variables $q_C, C \in S$, have the value zero, and constraint (6.36) is violated for this set S.

Conversely, if constraint (6.36) is violated for some set S, then all variables $q_C, C \in S$, have the value zero. That means that there exists a set of $U_n + 1$ trains, which are all pairwise concurrently present at the station. Thus, there exists a time instant at which $U_n + 1$ trains are concurrently present at the station. ■

6.3.2 Station Capacity in the PESP

In the PESP model, the basic idea is the following. For every train t, we count the trains that are present at station n at the same time as t, and limit that number by the capacity U_n. The analysis in the PESP model is based on the interpretation of the variables p_{ij} described in Section 4.3.2. There, we argued that for a periodic constraint with $u_{ij} \leq T - 1$, the integer variable p_{ij} can be interpreted as indicating the sequence in which the events i and j take place on the linear axis $[0, T-1]$.

More precisely, p_{ij} was shown to indicate the sequence of the events i and j on the linear axis $[0, T-1]$ as follows:

$$p_{ij} = \begin{cases} 0 & \text{if event } j \text{ takes place after, or at the same time as event } i, \\ 1 & \text{if event } j \text{ takes place before event } i. \end{cases}$$

The value $p_{ij} = 0$ does not distinguish between, on one hand, i and j taking place at the same time, and, on the other hand, j taking place after i. Therefore we also define the following constraint on the event pair (i,j):

$$1 \leq v_j - v_i + T\bar{p}_{ij} \leq T, \tag{6.37}$$

where $\bar{p}_{ij}$ is a new integer decision variable. Because constraint (6.37) has a time window with width $T-1$, it poses no extra restriction on the variables v_i and v_j. Moreover, $\bar{p}_{ij}$ can be restricted to take only the values $\{0,1\}$. For $\bar{p}_{ij}$, we have the sequence interpretation

$$\bar{p}_{ij} = \begin{cases} 0 & \text{if event } j \text{ takes place after event } i, \\ 1 & \text{if event } j \text{ takes place before, or at the same time as event } i. \end{cases}$$

Adding constraint (6.37) does not make matters much more complex, since any choice for p_{ij} has the following strong implications for $\bar{p}_{ij}$:

$$\begin{aligned} p_{ij} = 1 \Rightarrow &\quad \bar{p}_{ij} = 1, \\ p_{ij} = 0 \Rightarrow &\quad \begin{cases} \bar{p}_{ij} & = 1 \text{ if and only if } v_i = v_j, \\ \bar{p}_{ij} & = 0 \text{ otherwise.} \end{cases} \end{aligned} \tag{6.38}$$

Constraint (6.37), and in particular the integer variable $\bar{p}_{ij}$, is used in Theorem 6.9 to count the number of events that take place concurrently with the arrival or the departure of a train t'.

Suppose that a station graph G_n^S with arrival and departure node sets N_n^a and N_n^d contains all arcs

$$(i,j) \text{ with } i \in N_n^a, j \in N_n^d, \text{ and} \tag{6.39}$$

$$(i,j) \text{ with } i < j, i,j \in N_n^a. \tag{6.40}$$

So G_n^S contains an arc from every arrival node to every departure node, and an arc between each pair of arrival nodes, directed from the lower indexed node to the higher indexed one. If G_n^S does not contain all the arcs defined above, add them with the trivial time window $[0, T-1]$. For an arrival node i, let A_i^a be the set of arcs to and from other arrival nodes, and let A_i^d be the set of arcs from i to departure nodes, not including the dwell arc incident to i.

Theorem 6.9. Consider a station graph G_n^S as described above. For each $i \in N_n^a$, and for each arc $(i,j) \in A_i^a$, if $l_{ij} = 0$, add the periodic constraint

$$1 \leq v_j - v_i + T\bar{p}_{ij} \leq T,$$

with $\bar{p}_{ij} \in \{0,1\}$. Then the following constraints ensure that a station capacity of U_n is respected:

$$1+\sum_{(i,j)\in A_i^a}\bar{p}_{ij}+\sum_{(j,i)\in A_i^a}(1-p_{ji})-\sum_{(i,j)\in A_i^d}p_{ij}+\sum_{(i,j)\in A_n^{\text{dwell}}}p_{ij}\leq U_n \text{ for all } i\in N_n^a. \tag{6.41}$$

Proof. The proof is based on the sequence interpretation of the variables p_{ij} and $\bar{p}_{ij}$ on the linear axis $[0,T]$. Whenever the proof mentions 'before' or 'after', this is meant to consider the linear axis $[0,T]$ only.

Consider a train t, and the corresponding arrival event $i\in N_n^a$. Of all trains t' with corresponding arrival event $j>i$, the first sum term in (6.41) counts the number arriving at n before, or at the same time as train t does. By using $\bar{p}_{ij}$, the trains t' arriving at the same time as t are also counted. Similarly, the second sum term in (6.41) counts the number of trains t' arriving at n before, or at the same time as train t does, for all trains t' with corresponding arrival event $j<i$. Since the counting is done through the term $(1-p_{ij})$, the summation includes concurrencies. The third sum term counts the number of trains that leave n before t arrives. Since trains that leave at the same time as t arrives should not be counted as departed, we use p_{ij} here. The fourth term, finally, counts the number of trains that are dwelling at station n between time instants T and 0. These trains can therefore be seen as the stock of trains present at the station at time instant 0.

Thus, adding the stock of trains and the trains arriving before the arrival of t, adding one to incorporate for train t itself, and subtracting the trains leaving before t arrives, we obtain the total number of trains present at station n concurrently with train t. For any train t, the total number of trains in station n concurrently with t is limited by U_n. ■

6.4 The Rolling Stock Circulation Objective

The train cycles presented in Section 3.8.1 directly relate to the number of rolling stock compositions that are required to operate a cyclic railway timetable. This section describes that relation, and shows how to use it to define the rolling stock objective function F_s, which was introduced in Section 3.7.

We start with a description of the most basic case, in which the rolling stock circulation is fixed beforehand. This fixation is such that a rolling stock composition first operates some connection, then waits for some time at its destination station, and subsequently operates the same connection in the opposite direction. Next, we describe the case in which the model has more freedom in choosing the rolling stock circulation.

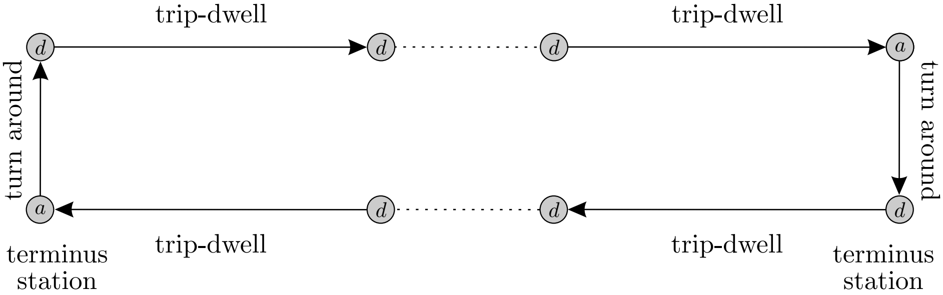


Figure 6.6: A train cycle C

6.4.1 Fixed Rolling Stock Circulation

Consider the case described in Section 3.8.1, in which a rolling stock composition first carries out the journey from terminus A to terminus B, and subsequently the reverse journey from B to A. This situation corresponds to the train cycle C in Figure 6.6 (see also Figure 3.13). For the time being, we omit the subscript tt' on the train cycle C. Suppose that a train departs from terminus A at time instant τ, travels to the other terminus B, dwells there for some time, departs again for the opposite journey, arrives at A, dwells there for some time, and departs again for B at time instant τ'. Measuring the elapsed time from τ until τ' in whole minutes (so *not* modulo T), we have that $\tau' - \tau = \sum_{a \in C} x_a$, because C is a directed cycle. Since $\sum_{a \in C} x_a = Tq_C$, the term Tq_C can be seen as the time it takes a train to 'travel along the cycle C'.

Because of the integrality of q_C, it will take a train either T or $2T$ or $3T$ etc. minutes to travel along its train cycle C. Moreover, the timetable is cyclic, which means that every T minutes a train departs from terminus A for terminus B. So, if it takes a train T minutes to travel along the cycle C in a certain timetable, then the same rolling stock composition can operate that trip in the next cycle period. If, however, it takes a train $2T$ minutes to travel along the cycle C, then the rolling stock composition will not be back in A in time to execute the next trip, and a second rolling stock composition is needed. And if it takes a train $3T$ minutes to travel along C, then both the first and the second rolling stock compositions will not be back in A after T minutes, and we need a third rolling stock composition. Extending this argument, one sees that the integer variable q_C for a train cycle C exactly measures the number of rolling stock compositions required to operate the journeys from A to B, and from B to A.

Therefore, we can use the variables q_C for train cycles C to construct a timetable that minimizes the required number of rolling stock compositions. Let C^t be the set of all train cycles in an instance $G = (N, A, l, u)$. The minimization of the number of required rolling stock compositions is then achieved by minimizing the objective function

$$F_s = \sum_{C \in C^t} q_C. \tag{6.42}$$

The function F_s simply adds the number of trains needed for each train cycle $C \in C^t$.

6.4.2 Choosing the Rolling Stock Circulation

This section describes the case of two train lines t and t', which are each operated in both directions. Train line t is operated in one direction by train t_1, and in the other direction by train t_2. Similarly, train line t' is operated by the trains t'_1 and t'_2. We consider the situation in which, for a certain terminus station, the train lines are allowed to either turn on themselves, or to turn on the other train line. That is, we allow the following to take place at the terminus station:

(*i*) Both train lines turn on themselves, which means that train t_1 turns on train t_2, and train t'_1 turns on train t'_2.

(*ii*) The train lines turn on one another, which means that train t_1 turns on train t'_2, and train t'_1 turns on train t_2.

We show how to model these situations such that the model can choose the minimum combined rolling stock circulation.

We assume that the departures of the trains t_2 and t'_2 from the terminus are related by a synchronization constraint with time window $[\underline{s}, \overline{s}]$. Further, we assume that all turn around time windows are equal to $[\underline{c}, \overline{c}]$. Finally, we assume that, at the other termini of the train lines t and t', both lines turn on themselves. These assumptions clarify the analysis, but are not crucial.

Figure 6.7 illustrates the problem by means of a small constraint graph. The cycles C and C' are the train cycles for trains t and t', respectively, for circulation (*i*). Node a represents the arrival of train t_1 at the terminus, and node d the departure of train t_2 from the terminus. Similarly, nodes a' and d' represent the arrival and departure of trains t'_1 and t'_2. The dashed arc from node d to node d' is the synchronization arc. The four double-lined arcs represent the choice arcs for the turning of the train lines t and t'. Cycle C_1 in Figure 6.8(a) shows the combined circulation (*ii*).

The double-lined arcs leaving node a show that train t_1 can turn on train t_2, which is represented by the arc (a, d), or train t_1 can turn on train t'_2, which is represented by arc (a, d'). Similarly, arc (a', d') represents the turning of train t'_1 on train t'_2, and arc (a', d) the turning of train t'_1 on train t_2. We will explain below why these turn around arcs are double-lined. The two leftmost and single-lined turn around

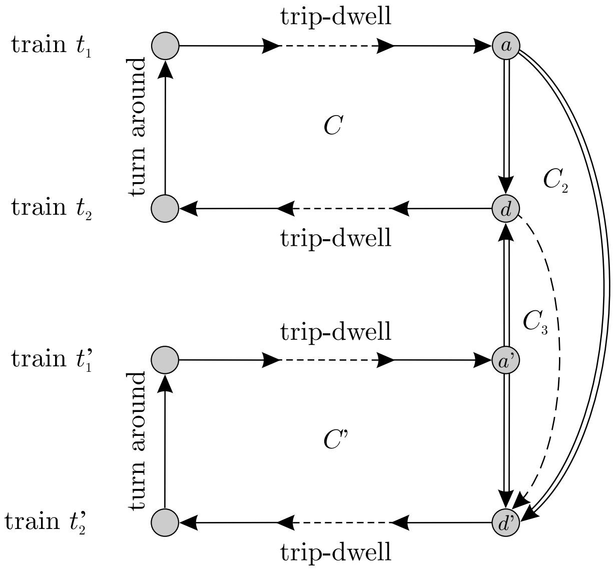


Figure 6.7: Combined rolling stock circulation for two trains t and t'.

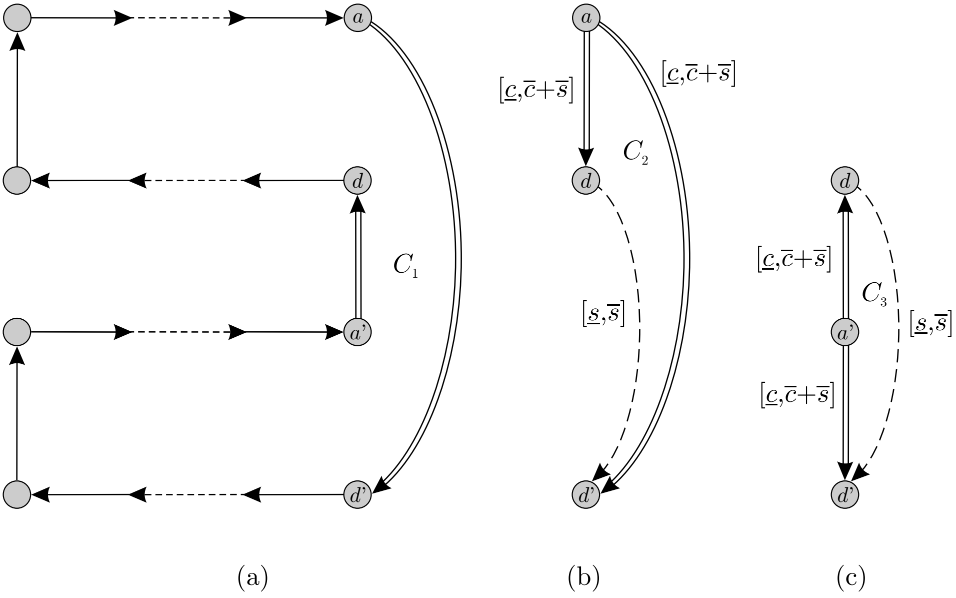

a
d
C_1
a'
d'
(a)
a
$[\underline{c},\overline{c}+\overline{s}]$
$[\underline{c},\overline{c}+\overline{s}]$
C_2
d
$[\underline{s},\overline{s}]$
d'
(b)
d
$[\underline{c},\overline{c}+\overline{s}]$
C_3
$[\underline{s},\overline{s}]$
a'
$[\underline{c},\overline{c}+\overline{s}]$
d'
(c)

Figure 6.8: (a) Rolling stock circulation (*ii*) corresponds to cycle C_1, (b) Cycle C_2 with time windows, (c) Cycle C_3 with time windows.

arcs represent the turning of the train lines at the other termini. These arcs do not play an important part in our analysis, and are included because they are part of the train cycles C and C'.

Next, we describe how the choice between the rolling stock circulations (i) and (ii) can be modeled. At the end of Section 3.4.3, we explained that turn around constraints can be seen as a special type of connection constraint, since they model a connection between the train that ends at the terminus, and the train that starts at the terminus. So, the choice between the circulations (i) and (ii) is quite similar to the two-to-two flexible connection in Section 6.2.3. Indeed, as in Section 6.2.3, the choice between the two above circulations is modeled by the constraints

$$d - a \in [\underline{c}, \overline{c}]_T \text{ and } d' - a' \in [\underline{c}, \overline{c}]_T, \tag{6.43a}$$

$$\text{or}$$

$$d - a' \in [\underline{c}, \overline{c}]_T \text{ and } d' - a \in [\underline{c}, \overline{c}]_T. \tag{6.43b}$$

And, as in Section 6.2.3, these constraints can be written as

$$d - a \in [\underline{c}, \overline{c} + \overline{s}]_T \text{ and } d - a \in [\underline{c} + \underline{s}, \overline{c} + T]_T, \text{ and} \tag{6.44a}$$

$$d - a' \in [\underline{c}, \overline{c} + \overline{s}]_T \text{ and } d - a' \in [\underline{c} + \underline{s}, \overline{c} + T]_T, \text{ and} \tag{6.44b}$$

$$d' - a \in [\underline{c}, \overline{c} + \overline{s}]_T \text{ and } d' - a \in [\underline{c} + \underline{s}, \overline{c} + T]_T, \text{ and} \tag{6.44c}$$

$$d' - a' \in [\underline{c}, \overline{c} + \overline{s}]_T \text{ and } d' - a' \in [\underline{c} + \underline{s}, \overline{c} + T]_T, \tag{6.44d}$$

under the condition that $\overline{c} - \underline{c} < \underline{s} = \frac{1}{2}T - \delta$. The system (6.44) is the reason for the double-lined arcs in Figure 6.7. Each of the double-lined arcs in fact represents two arcs, one for the leftmost constraints in (6.44), and one for the rightmost constraints.

In the further analysis below, we use the CPF for the CRTP. Therefore, it is important to note that, for each of the four double-lined turn around arcs, we have two tension variables, since there are in fact two arcs. For a double arc between two nodes i and j, we denote the two tension variables by x_{ij} and $\bar{x}_{ij}$, with time windows $x_{ij} \in [\underline{c}, \overline{c} + \overline{s}]$, and $\bar{x}_{ij} \in [\underline{c} + \underline{s}, \overline{c} + T]$. Moreover, this means that the constraint graph in Figure 6.7 contains more cycles than it seems at first sight. For example, the train cycle C in fact consists of two cycles, since there are two arcs between nodes a and d.

For our rolling stock circulation choice, however, we only need to consider the cycles that contain the tension variables x_{ij}. This follows from the observation that the variables x_{ij} lie in the time windows $[\underline{c}, \overline{c} + \overline{s}]$, so they represent the actual turn around times. Note that the other tension variables $\bar{x}_{ij}$ are still needed for correctly modeling the choice between circulations (i) and (ii). They are just not needed for determining the number of rolling stock compositions. In the remainder of this section, we therefore only consider the tension variables x_{ij} for double arcs.

Now that it is clear how to model the choice between the circulations (i) and (ii), we proceed with the problem of determining the minimum combined circulation. Let γ and γ' be the cycle vectors for cycles C and C', and let q and q' be the corresponding integer variables. Further, for each of the cycles $C_i, i = 1, 2, 3$, let γ_i be the corresponding cycle vector, and let q_i be the corresponding integer variable. As was explained in the previous section, the number of train compositions needed to operate circulation (i) equals $q + q'$. Similarly, the number of train compositions for circulation (ii) equals q_1.

Next, consider the cycles C_2 and C_3 in Figure 6.8, and direct these such that the synchronization arc appears forwardly in both cycles. Then, both cycles consist of the forward synchronization arc with time window $[\underline{s}, \overline{s}]$, and of a forward and a backward turn around arc. Recall from the above that we consider these turn around arcs as corresponding to tension variables x_{ij}, with time windows $[\underline{c}, \overline{c} + \overline{s}]$. Using Lemma 5.7, we obtain the following bounds on the integer variables q_2 and q_3:

$$\left\lceil \frac{\underline{c} - \overline{c} + \underline{s} - \overline{s}}{T} \right\rceil \leq q_2, q_3 \leq \left\lfloor \frac{\overline{c} - \underline{c} + 2\overline{s}}{T} \right\rfloor .$$

Since $\underline{s} + \overline{s} = T$, and $\overline{s} = \frac{1}{2}T + \delta$, and further assuming that $\overline{c} - \underline{c} < T - 2\delta$, we find that $q_2, q_3 \in \{0, 1\}$.

Finally, we apply Lemma 5.11 to the cycles C_2 and C_3, to see that

$$q_2 = \begin{cases} 1 \text{ if and only if the cyclic sequence } a \rightarrow d' \rightarrow d \text{ takes place,} \\ 0 \text{ if and only if the cyclic sequence } a \rightarrow d \rightarrow d' \text{ takes place.} \end{cases}$$

$$q_3 = \begin{cases} 1 \text{ if and only if the cyclic sequence } a' \rightarrow d' \rightarrow d \text{ takes place,} \\ 0 \text{ if and only if the cyclic sequence } a' \rightarrow d \rightarrow d' \text{ takes place.} \end{cases}$$

The cycle periodicity constraint for C_2 reads $x_{ad} + x_{dd'} - x_{ad'} = Tq_2$. And, because of (6.43), we know that either $x_{ad} \in [\underline{c}, \overline{c}]$ and $x_{ad'} \in [\underline{c} + \underline{s}, \overline{c} + \overline{s}]$, or $x_{ad'} \in [\underline{c}, \overline{c}]$ and $x_{ad} \in [\underline{c} + \underline{s}, \overline{c} + \overline{s}]$. One can inspect that the latter case corresponds to $q_2 = 1$, and the former to $q_2 = 0$. Since a similar argument holds for cycle C_3, the above relation can also be stated as

$$q_2 = \begin{cases} 1 \text{ if and only if circulation } (ii) \text{ is operated,} \\ 0 \text{ if and only if circulation } (i) \text{ is operated.} \end{cases}$$

$$q_3 = \begin{cases} 1 \text{ if and only if circulation } (i) \text{ is operated,} \\ 0 \text{ if and only if circulation } (ii) \text{ is operated.} \end{cases}$$

Note that this implies $q_3 = 1 - q_2$.

This means that we can use either q_2 or q_3 as a binary variable that indicates which of the circulations (i) and (ii) is operated. Arbitrarily choosing q_2 for this role, we can thus minimize the rolling stock circulation in this example by minimizing the non-linear expression

$$(1 - q_2)(q + q') + q_2 q_1. \tag{6.45}$$

This expression states that if circulation (i) is operated, the objective is to minimize $q + q'$, and otherwise the objective is to minimize q_1.

Next, we rewrite the non-linear objective (6.45) to a linear objective function, which requires the reformulation of some cycles in terms of other cycles. To that end, let C_4 be the clockwise directed cycle formed by the four double-lined arcs in Figure 6.7, and let γ_4 and q_4 be its cycle vector and integer variable. One can check that $\gamma_1 = \gamma + \gamma' + \gamma_4$, and thus we have $q_1 = q + q' + q_4$. That means that (6.45) is equivalent to

$$(1 - q_2)(q + q') + q_2(q + q' + q_4) = q + q' + q_2 q_4.$$

Further, it holds that $\gamma_4 = \gamma_3 - \gamma_2$, and thus that $q_4 = q_3 - q_2$. Together with the earlier found equality $q_3 = 1 - q_2$, this gives $q_4 = 1 - 2q_2$. Substituting this into the above expression, we obtain

$$q + q' + q_2 q_4 = q + q' + q_2(1 - 2q_2).$$

In the latter expression, the term $q_2(1 - 2q_2)$ has value zero whenever $q_2 = 0$, and value -1 whenever $q_2 = 1$. For the considered situation in Figure 6.7, the minimum combined rolling stock circulation is therefore obtained by the linear objective function

$$\text{Minimize } q + q' - q_2. \tag{6.46}$$

The above analysis holds for a general pair of train lines t and t', that are related by a synchronization constraint, and for which the choice of the rolling stock circulation is modeled by a system of turn around constraints as (6.44). We did assume that

$\overline{c} - \underline{c} < T - 2\delta$, but this assumption is hardly restrictive in practice, especially since we already have the general time window assumption $\overline{c} - \underline{c} \leq T - 2$. So, the above model is less restrictive than it seems. For example, the same modeling approach can be applied to the turn around times at the other termini for the trains t and t', that is, to the leftmost turn around arcs in Figure 6.7. This offers another degree of freedom in constructing a timetable that minimizes the rolling stock utilization. However, for the above approach to be applicable, we do need to decide beforehand which pairs of trains are allowed to turn on one another.

6.5 A Cycle Fixation Heuristic

In Section 2.1 we explained that, in practice, planners construct timetables by iteratively fixing the schedule of a train. Typically, a planner first schedules international and intercity trains, interregional trains next, and finally completes the timetable by scheduling the local trains. Also in the DONS DSS, planners usually first enter some intercity trains, compute a timetable, add some more intercity trains, again compute a timetable, etc. Next, they iteratively add interregional trains, and finish with the local trains. Clearly, the timetable computed after adding some new trains may differ completely from the timetable computed in the previous step. If the subsequently computed timetables remain the same for some iterations during the construction process, then the planner tends to develop an ‘intuitive feeling’ for this timetable. Therefore, significant changes in subsequent timetables are considered undesirable by practitioners.

The cycle fixation heuristic described in this section is inspired by this practical planning method. Rather than trying to compute a timetable for a complete instance in one run, we propose to use a multi-stage approach. The stages need not correspond to train types. For example, one may first plan the timetable for a certain part of the network, and add other parts of the network in subsequent stages.

6.5.1 An Illustrative Example of Fixing Cycles

We start the description of the cycle fixation heuristic with the example of first planning a timetable for the intercity network, then fixing the sequence of the intercity trains, adding all other trains, and computing a timetable for the complete network. The sequence of the intercity trains is fixed by fixing the cycle integer variables for all safety triangles in the intercity network constraint graph.

Let the CRTP instance representing the complete railway system be given by the constraint graph $G = (N, A, l, u)$, and let $\bar{G} = (\bar{N}, \bar{A}, \bar{l}, \bar{u})$ be the subgraph of

G corresponding to the intercity network. So all nodes $i \in \bar{N}$ represent arrival and departure events for intercity trains only, any arc $a \in \bar{A}$ connects two nodes in $\bar{N}$, and the vectors $\bar{l}$ and $\bar{u}$ contain the time window lower and upper bounds for the arcs in $\bar{A}$.

Step 1: Construct a timetable for $\bar{G}$. First, use the CPF to construct a timetable for the intercity instance $\bar{G}$. Let the solution be denoted by $(\bar{x}, \bar{q})$.

Step 2: Fix the sequence of the intercity trains. Section 5.4 described the relation between safety triangles in safety cliques and the cyclic sequencing of trains. There, Lemma 5.13 stated that the departure sequence from a station n of three trains $i, j,$ and k relates to the triangle (i, j, k) in the safety clique K_n as follows:

$$\bar{q}_{ijk} = \begin{cases} 0 & \text{if and only if events } i, j, k \text{ occur in the cyclic sequence } i \to j \to k, \\ 1 & \text{if and only if events } i, j, k \text{ occur in the cyclic sequence } i \to k \to j. \end{cases}$$

As a consequence, fixing the variables $\bar{q}_{ijk}$ for all safety triangles $(i, j, k) \in \bar{G}$ results in fixing the sequence in which the trains leave the stations. Let $\bar{K}$ be the set of all safety cliques in $\bar{G}$. We shall use the values $\bar{q}_{ijk}$ for $(i, j, k) \in \bar{K}$ to fix the intercity train sequences.

Step 3: Construct a timetable for G. We compute a timetable for G by solving the CPF for G, while fixing the intercity train sequences to the sequences in the timetable obtained in Step 2. For each safety triangle $(i, j, k) \in \bar{K}$, and for a cycle basis B, let $(\lambda^1_{ijk}, \ldots, \lambda^c_{ijk})$ be the linear combination of basis cycles that yields the triangle (i, j, k). We solve the following problem:

$$\begin{aligned} \text{Minimize} \quad & F(x) & & \\ \text{satisfying} \quad & \sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = T q_C & \text{for all } C \in B & \quad (6.47\text{a}) \\ & \sum_{C \in B} \lambda^C_{ijk} q_C = \bar{q}_{ijk} & \text{for all } (i, j, k) \in \bar{K} & \quad (6.47\text{b}) \\ & l_a \le x_a \le u_a & \text{for all } a \in A & \quad (6.47\text{c}) \\ & x_a \in \mathbb{R} & \text{for all } a \in A & \quad (6.47\text{d}) \\ & q_C \in \mathbb{Z} & \text{for all } C \in B & \quad (6.47\text{e}) \end{aligned}$$

In the above formulation, constraint (6.47b) fixes the intercity train sequences to those obtained in the timetable for $\bar{G}$.

Note that, because of the fixing of the intercity train sequences through constraints (6.47b), the formulation (6.47) may not be feasible, even if the original instance G does have a feasible solution.

6.5.2 Formal Description of the Heuristic

The above example can be generalized in two directions. First, instead of using a two-stage approach, one may choose to use multiple stages of iteratively fixing train sequences, for example, first high speed and international trains, then intercity trains, then interregional trains, and finally local trains. Second, the fixation in a stage does not have to concern all trains of a certain type. For example, for some part of the railway network, we may choose to consider all trains in each stage, and to iteratively increase the size of the considered network part. These two possibilities for the fixation in a stage are discussed in more detail in the next section.

For an instance $G = (N, A, l, u)$, let us consider S stages, numbered $1, \ldots, S$, and thus a sequence of constraint graphs $G^1, \ldots, G^S$, with $G^1 \subset G^2 \subset \cdots \subset G^S = G$. For stage s, graph G^s is defined as $G^s = (N^s, A^s, l^s, u^s)$. As in the example, the vectors l^s and u^s contain the time window lower bounds and upper bounds for the arcs in A^s. Moreover, for each stage s, let C^s be the total set of cycles for which the integer variables are fixed in that stage. So, C^s contains the cycles for which solution values were obtained in the stages $1, \ldots, s-1$. The structure of this so-called *cycle fixation set* C^s reflects the aspects of the timetable that are fixed when proceeding from one stage to the next.

For a cycle $D \in C^s$, the fixed value of the integer variable is denoted by $\bar{q}_D$. For formulating the CPF in stage s, we use a cycle basis B^s. For a cycle $D \in C^s$, and for a cycle basis B^s, let $(\lambda_D^1(s), \ldots, \lambda_D^c(s))$ be the linear combination of basis cycles in B^s that yields D. The cycle fixation heuristic iteratively solves the following mathematical program for $s = 1, \ldots, S$:

$$\begin{array}{lll} \text{Minimize} & F(x) & \\ \text{satisfying} & \displaystyle\sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = T q_C & \text{for all } C \in B^s \quad (6.48a) \\ & \displaystyle\sum_{i=1}^{c} \lambda_D^i(s) q_i = \bar{q}_D & \text{for all } D \in C^s \quad (6.48b) \\ & l_a^s \le x_a \le u_a^s & \text{for all } a \in A^s \quad (6.48c) \\ & x_a \in \mathbb{R} & \text{for all } a \in A^s \quad (6.48d) \\ & q_C \in \mathbb{Z} & \text{for all } C \in B^s \quad (6.48e) \end{array}$$

This program is just the CPF, formulated with the cycle basis B^s, and with the extra constraint (6.48b). This extra constraint requires the fixation of the cycle integer variables for the cycles in C^s to the values that were obtained in the previous $s-1$ stages.

6.5.3 Choosing Stages and Cycle Fixation Sets

We argued that, by choosing the cycle fixation sets C_s properly, one can fix certain properties of the timetable when going from one stage to the next. This section describes two possible choices for these sets. It shows that the number of stages S is closely related to the cycle fixation sets C^s.

Fixing cyclic sequences within train types Rather than considering intercity trains in the first stage and all other trains in the second stage, suppose that we are given an ordered classification $t^1, \ldots, t^S$ of S train types. The first type t^1 represents the most important train type, and the last type t^S represents the least important one. The most important train type t^1 corresponds to stage 1, the least important train type to stage S, and the types in between to the stages $2, \ldots, S-1$. The cycle fixation sets are constructed such that in stage s the sequence of all trains of types $t^1, \ldots, t^{s-1}$ is fixed, as in Step 2 of the example.

Fixing cyclic sequences within parts of the network A second option for choosing the stages, and the cycle fixation sets, is to start with a small part of the railway network, to construct a timetable, and to fix the sequences of the trains for that part. Next, increase the size of the considered part iteratively, and after each iteration fix the sequence of all trains for that part of the network. The number of stages S then equals the number of increases until arriving at the complete railway network. Since we iteratively increase the size of the network, we not only construct a sequence of constraint graphs $G^1, \ldots, G^S$, but also a sequence of railway networks, denoted by $\mathcal{G}^1 \subset \cdots \subset \mathcal{G}^S = \mathcal{G}$, where the network for stage s is defined as $\mathcal{G}^s = (\mathcal{N}^s, \mathcal{A}^s)$ (see Section 3.5).

Chapter 7

Computational Results

We tested the models and ideas presented in the previous chapters on several real-life railway timetabling instances. This chapter reports on the computational results of these experiments. We first describe the test instances in Section 7.1. Next, Section 7.2 presents some preprocessing techniques for reducing the sizes of the corresponding constraint graphs, and for strengthening the time windows. In Section 7.3, we report on our findings for solving the PESP formulation for the CRTP. Section 7.4 discusses the experimental results of the cycle basis algorithms presented in Section 5.7. Next, based on the thus constructed cycle bases, we formulate the CPF, and compare the computation times for solving the model for the various cycle bases. Section 7.6 describes our findings for solving the CPF with the robustness objective function, and Section 7.7 reports on the results for the rolling stock utilization objective. The constraint violation objective function is the subject of Section 7.8. In Section 7.9, we investigate the effects of adding cutting planes to the CPF, and Section 7.10 describes the results of our experiments for the cycle fixation heuristic. Finally, Section 7.11 compares the quality of our solutions with the solutions obtained by the CADANS algorithm (Schrijver and Steenbeek, 1994).

For solving the Mixed Integer Programs (4.2) and (5.3), we used the MIP solver CPLEX 7.5 (CPLEX website, 2002). Unless stated otherwise, we used the default settings for all CPLEX parameters. All experiments were performed on an AMD Athlon XP 1500+, with 512 MB memory, operating under Linux kernel 2.4.10.

7.1 Description of the Instances

Our computational experiments considered three real-life instances, which were obtained from NSR and Railned. The three instances are based on the Dutch railway system for 1997/1998, which is quite similar to the 2001/2002 railway system. For all three instances, we consider a cycle time of one hour, that is, $T = 60$ minutes.

IC97 Instance IC97 contains all trains for the Dutch 1997/1998 intercity network. The instance consists of 50 stations and 15 train lines. All trains have frequency one, but large parts of the network are served by multiple train lines. In that case, frequency constraints between these multiple lines are stated, which define a 30 minute service on the majority of the routes. These frequencies are assumed to be perfect, which means that the corresponding frequency constraints are equality constraints. Connection constraints are defined such that one can travel between any two intercity stations with only a brief transfer time. The instance also contains many combining and splitting events between trains.

ICIR97 Instance ICIR97 represents the 1997/1998 intercity and interregional networks for the Netherlands. For the intercity network part, it is equal to the IC97 instance. It additionally consists of 11 interregional trains, and 24 stations that are served by the interregional trains, but not by the intercity trains. Most of the interregional trains have perfect frequency two. Again, connection constraints are defined so as to offer convenient and swift travel between most stations.

NH97 The NH97 instance consists of all trains in the 1997/1998 timetable for the part of the Netherlands north of the line Leiden–Amsterdam, known as North-Holland. It contains 50 stations, and 10 train lines of various train types, namely intercity trains, interregional trains, local trains, and one cargo train. Since the instance does not contain international trains, there are no fixed arrival or departure time constraints. All train lines but the cargo train have frequency two, with bandwidth $\delta = 2$. Connection constraints are defined to enable transfers between several trains.

The IC97 and ICIR97 instances are defined too tightly, and are therefore infeasible. As such, we can use them to test the objective function F_v (see Section 3.7). Moreover, the IC97 and ICIR97 instances are feasible when we increase the upper bounds on the dwell and connection time windows. So, after increasing these upper bounds, we can still use the objective functions F_t, F_r, and F_s. In fact, all computations in this chapter for the IC97 and ICIR97 instances with the F_t, F_r, and F_s objective functions were performed with the increased upper bounds on the dwell and connection time windows.

The ICIR97 instance is the largest and most complex test instance. Indeed, we were not able to compute any integer optimal solution for ICIR97. Therefore, we used the cycle fixation heuristic from Section 6.5 in order to obtain good solutions for the ICIR97 instance.

For more detailed information on the instances, we refer to Appendix C, and to Table 7.1 in the next section. Moreover, Appendix C also shows a visualization of the constraint graph for the IC97 instance.

7.2 Preprocessing

We first present some techniques for shrinking the usually very large constraint graphs as they are obtained from DONS. Next, we describe an algorithm for strengthening the remaining time windows. Finally, the CPF is re-formulated such that all cycle periodicity integer variables have lower bound zero.

7.2.1 Shrinking the Constraint Graph

Our instances are obtained from the cyclic railway timetabling decision support system DONS (see Section 2.5). The initial size of the instances can be reduced drastically by applying some basic shrinking techniques. By deleting nodes and arcs from the constraint graph G, we may also delete some cycles. This in turn reduces the number of integer variables, which can considerably improve the solution process.

Removing parallel arcs. Instances obtained from DONS typically contain many parallel arcs. For example, they contain safety constraints between trains that belong to the same train line. But trains within a train line are typically already separated by their frequency constraints. We intersect the time windows of the parallel arcs, and replace them by a single arc with the intersected time window. In doing so, it is important to check that the resulting intersected time window is connected (see Section 4.3).

Nodes with degree one or two. Nodes with degree one in the constraint graph can be deleted, since they are not included in any cycle. A tension corresponding to a deleted arc can be calculated in a post-processing phase. The adjacent arcs of a node with degree two can be contracted, which means that they are merged into a single arc. The time window of the contracted arc is the sum of the original two time windows. Let a node j with degree two have adjacent arcs (i, j) and (j, k), and periodic time window constraints

$$v_j - v_i \in [l_{ij}, u_{ij}]_T,$$
$$v_k - v_j \in [l_{jk}, u_{jk}]_T.$$

Then the arcs (i, j) and (j, k) can be contracted into an arc (i, k) which represents the constraint

$$v_k - v_i \in [l_{ij} + l_{jk}, u_{ij} + u_{jk}]_T.$$

A contracted arc can be expanded during post-processing to calculate the tensions for the two original arcs. As with removing parallel arcs, we only execute a contraction if the time window of the contracted arc is not disjoint. One may model the disjoint time window of a contracted arc by two parallel arcs, as was explained in Section 4.3. But that would mean removing one arc because of the contraction, and adding a new arc to model the disjoint time window.

Substituting trip time arcs. The DONS instances contain many trip time constraints. Since a trip time constraint in the basic model is an equality constraint, either the involved departure time variable, or the arrival time variable can be substituted out. We perform the substitutions such that all arrival time variables are eliminated. The substitution of trip time arcs also has another impact. Consider the small track graph in Figure 7.1(a) (see Section 3.8 for a description of the track graph). There, three trains are running along the track $(1, 2)$, resulting in three un-merged train paths P_1, P_2, and P_3. The two displayed safety cliques K_1 and K_2 are for leaving node 1 and entering node 2, respectively. Figure 7.1(b) shows the track graph after substituting the trip time arcs, which results in parallel safety arcs, namely the safety arcs at either end of the substituted trip arcs. And these parallel safety arcs can be removed as described above, which results in the graph in Figure 7.1(c).

The sizes of the constraint graphs can be reduced considerably by applying these shrinking techniques. Table 7.1 summarizes some statistics on the sizes of the instances, and shows the impact of the preprocessing techniques. The table shows the number of nodes and arcs in the constraint graph, both before and after shrinking. The total number of arcs is split into the various corresponding constraint types. One clearly sees the reduction in the number of trip arcs and safety arcs. The last row of the table shows the number of cycles in a cycle basis of the constraint graph. The number of basis cycles is reduced considerably by the shrinking phase, which yields a reduction in the number of integer variables in the CPF.

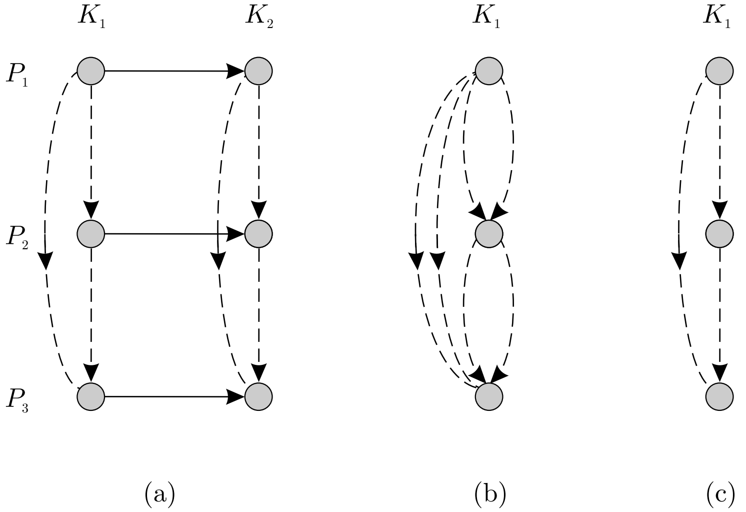


Figure 7.1: (a) Track graph, (b) after substituting trip arcs, (c) after removing parallel safety arcs.

	original			preprocessed		
	IC97	NH97	ICIR97	IC97	NH97	ICIR97
nodes	1475	532	2553	205	222	455
arcs	3342	1766	7426	523	958	1657
trip arcs	1258	310	2088	0	0	0
dwell arcs	172	180	376	160	180	367
frequency arcs	42	70	162	42	70	162
connection arcs	34	24	39	34	24	39
fixed arcs	5	0	5	5	0	5
safety arcs	1831	1182	4756	282	684	1084
cycles	1868	1235	4874	319	737	1203

Table 7.1: Brief statistics on the sizes of IC97, NH97, and ICIR97, before and after shrinking the constraint graph.

```
Algorithm: Strengthening periodic time windows.
Input: PESP instance $G = (N, A, l, u)$ and $T$.
Output: Strongest time windows $[l_{ij}, u_{ij}]$.

for all nodes $i, j \in N$ with $i \neq j$ do
    set $S_{ij} = \begin{cases} [l_{ij}, u_{ij}] & \text{if } (i,j) \in A, \\ [T - u_{ij}, T - l_{ij}] & \text{if } (j,i) \in A, \\ [0, T-1] & \text{otherwise.} \end{cases}$
found = true.
while found do
    found = false.
    for all nodes $i, j, k \in N$ with $i \neq j \neq k$ do
        $S' = S_{ij} \oplus S_{jk}$.
        if $S_{ik} \not\subseteq S'$ then
            $S_{ik} \leftarrow S_{ik} \cap S'$.
            found = true.
```

Figure 7.2: Strengthening the periodic time windows.

7.2.2 Strengthening Time Windows

For strengthening the time windows, we use the algorithm proposed by Schrijver and Steenbeek (1994), Nachtigall (1999), Lindner (2000), which resembles the Floyd-Warshall all-pairs shortest path algorithm (see, for example, Ahuja et al., 1993).

The algorithm views each time window $[l_{ij}, u_{ij}]_T$ as a set of numbers

$$S_{ij} = \{x_{ij} \text{ modulo } T \mid x_{ij} \in \{l_{ij}, \dots, u_{ij}\}\}.$$

For two of these sets S_{ij} and S_{jk}, it uses the binary operator $\oplus$ which is defined as

$$S_{ij} \oplus S_{jk} = \{x_{ij} + x_{jk} \text{ modulo } T \mid x_{ij} \in S_{ij}, x_{jk} \in S_{jk}\}.$$

So, for the nodes i and k, the set $S' = S_{ij} \oplus S_{jk}$ expresses the time window that is implied by the path (i, j, k) and the time windows $[l_{ij}, u_{ij}]$ and $[l_{jk}, u_{jk}]$. Any feasible solution for the nodes i and k has to respect both the time window represented by S_{ik}, and the implied time window represented by S'. Therefore, if $S_{ik} \not\subseteq S'$, then we can replace S_{ik} by $S_{ik} \cap S'$. Figure 7.2 shows the algorithm for strengthening the periodic time windows.

	IC97	NH97	ICIR97
strengthened arcs	57	408	97
avg$(u_{ij} - l_{ij})$			
before	50.9	53.7	52.0
after	18.5	43.8	19.6

Table 7.2: The results of strengthening the time windows for the IC97, NH97, and ICIR97 instances.

Nachtigall (1999) showed that the algorithm has a worst case time complexity of $O(T|N|^4)$. We applied the algorithm to the three test instances, which took a computation time of a few seconds. The results are shown in Table 7.2. The first row in the table shows the number of arcs for which the time window improves. For these arcs, the second row shows the average time window width before strengthening, and the third row shows the average time window width after strengthening. Almost all improved time windows are safety time windows, which explains the high average width before strengthening. For the IC97 and ICIR97 instances, not too many arcs are improved, but the realized improvements are more than half of the time window width. For the NH97 instance, the average improvement is less, but about two third of all safety arcs are improved.

7.2.3 Zero Lower Bounds

Finally, we re-formulate the CPF by substituting for each cycle periodicity integer variable $q_C = \bar{q}_C + a_C$. This gives the following formulation for the CPF:

$$\begin{array}{lllr}
\textbf{CPF:}\ \text{Minimize} & F(x) & & \\
\text{subject to} & \sum_{a \in C^+} x_a - \sum_{a \in C^-} x_a = T\bar{q}_C + Ta_C & \text{for all } C \in B & (7.1a) \\
& l_a \leq x_a \leq u_a & \text{for all } a \in A & (7.1b) \\
& 0 \leq \bar{q}_C \leq b_C - a_C & \text{for all } C \in B & (7.1c) \\
& x_a \in \mathbb{R} & \text{for all } a \in A & (7.1d) \\
& \bar{q}_C \in \mathbb{Z} & \text{for all } C \in B & (7.1e)
\end{array}$$

Because of the zero lower bounds in (7.1c), this formulation contains a binary variable for each cycle $C \in B$ with width $W_C = b_C - a_C = 1$. This has the advantage that CPLEX can apply any special solution approaches for binary variables to these variables.

	Best integer	Best LP	Nodes	Root LP+cuts	Root LP	Optimal
IC97						
F_t	4049	3880.63	231647	8.03	0	3942
$F_{t,q}$	737	14.13	74539	12.3	0	154
F_r	11	0	385340	0	0	1
NH97						
F_t	1428	1414	186812	8	0	1418
$F_{t,q}$	180	0	161646	0	0	48
F_r	189	0	140592	0	0	?
ICIR97						
F_t	6437	6230.28	104300	2.28	0	6340*
$F_{t,q}$	2830	2.07	50452	2.07	0	344*
F_r	-	0	76694	0	0	?

Table 7.3: Results after 3600 seconds computation time using the PESP model for the IC97, NH97, and ICIR97 instances (* best found solution).

7.3 Solving the PESP formulation

We first solved the IC97, NH97, and ICIR97 instances using the PESP formulation (4.2). We used three objective functions, namely minimizing passenger travel time, both as the linear function F_t and as the quadratic function $F_{t,q}$, and minimizing the robustness function F_r. None of the instances could be solved to optimality within a time limit of 3600 seconds. The results after 3600 seconds of computation time are shown in Table 7.3. The first column shows the value of the best found integer feasible solution, the second column the value of the LP relaxation, and the third column the number of nodes in the Branch&Cut tree. The fourth column displays the value of the root LP relaxation after CPLEX has added cutting planes, and the fifth column shows the root LP value without these cuts. The final column shows the value of the optimal solution, which we obtained in later experiments for the CPF model. One sees that the PESP model may be quite far from the optimal solution after 3600 seconds. We also executed the PESP model without a time limit, but after several hours of computing all available memory had been used, and the model had not yet been solved to optimality.

In the next sections, we show that the CPF model yields a much better performance. Therefore, the remainder of this chapter focuses on the CPF (5.3).

7.4 Results of the Cycle Basis Algorithms

We applied the cycle basis algorithms from Section 5.7 to the three instances IC97, NH97, and ICIR97. This section illustrates the results for the IC97 instance through tables and figures. Similar tables and figures for the other two instances can be found in Appendix D. The results in this section are based on joint work with Van den Braber (2001), who studied the BFS algorithms by Deo et al. (1982) for CRTP problems. The section further reports on the MCB, FCB, MST, and LNT algorithms.

First, we applied the MCB algorithm by Horton (1987), and checked whether the resulting cycle basis was fundamental. The arc costs for the MCB were set equal to the time window widths. Next, we executed the LNT and MST algorithms, and applied Berger's (2002) FCB algorithm to the LNT and MST spanning trees. We name the resulting cycle bases LNF and MF, respectively.

The BFS algorithms by Deo et al. (1982) are designed for the unit arc weight case, and their basic idea is to iteratively add high degree nodes to a partial spanning tree. However, the structure of a CRTP instance is such that high degree nodes are typically nodes that appear in large safety cliques. So, the BFS algorithms yield a spanning tree consisting of many safety arcs, which is likely to result in a quite wide strictly fundamental cycle basis.

Following Van den Braber (2001), we also executed the BFS algorithms with the restriction that safety arcs are not allowed to be included. This yields the algorithms SDSr, DDSr, UEr, MBFSr, where the 'r' stands for restricted. In fact, the restricted algorithms have been implemented by running the 'regular' BFS algorithms on a constraint graph from which all safety arcs have been deleted. Once the spanning tree has been constructed, the safety arcs are inserted in the graph again, and the strictly fundamental cycle basis is computed.

The results of the algorithms for the IC97 instance are reported in Table 7.4, and Figures 7.3 and 7.4. The results for the instances NH97 and ICIR97 can be found in Tables D.1 and D.2, and in Figures D.1–D.4. The first column in Table 7.4 shows the log-width $LW(B)$ of the cycle bases. The average width of the cycles in B is given by $\text{avg}(W_C)$, in the second column, and the maximum width by $\max(W_C)$, in the third column. We define the length of a cycle C as $L_C = |C|$, that is, as the total number of arcs in C. The length of a cycle basis B is simply $\sum_{C \in B} L_C$, and is shown in the fourth column. The average length of the cycles in B is given by $\text{avg}(L_C)$, in the fifth column, and the maximum length by $\max(L_C)$, in the sixth column.

The leftmost vertical axis in Figure 7.3 depicts $LW(B)$, and the rightmost vertical axis depicts $L(B)$. For each cycle basis, the white bar in Figure 7.3 shows the value

	$LW(B)$	avg(W_C)	max(W_C)	$L(B)$	avg(L_C)	max(L_C)
MCB	119	0.54	4	2473	7.8	22
LNF	136	0.66	4	1843	5.8	40
MF	128	0.61	4	1868	5.9	40
LNT	220	1.52	5	5630	17.6	56
MST	211	1.42	5	5332	16.7	52
SDSr	235	1.50	4	3624	11.4	42
DDSr	217	1.39	6	3046	9.5	43
UEr	210	1.35	5	3336	10.5	34
MBFSr	215	1.42	9	3387	10.6	46
SDS	298	1.70	2	2072	6.5	17
DDS	268	1.49	6	1793	5.6	23
UE	227	1.16	5	1358	4.3	22
MBFS	282	1.69	9	1753	5.5	25

Table 7.4: Cycle basis algorithms results for IC97.

$LW(B)$. The other, multiply colored, bar shows the length $L(B)$. Moreover, this bar also shows the arc type composition of the cycle basis. For example, the black part of the second bar shows that the MCB cycle basis contains about 1150 dwell arcs. Finally, Figure 7.4 shows how many cycles of a certain width each cycle basis contains. Note that the total number of cycles is equal for each cycle basis.

A first surprising result is that, for all three instances, the MCB cycle bases turn out to be fundamental. Less surprisingly, they are the bases with smallest width, and also have small length. Moreover, the MCB cycle bases consist almost completely of cycles that are either fixed beforehand, because their width equals zero, or of cycles with width equal to one.

The LNT and MST algorithms yield the longest cycle bases, and this goes together with a quite large width. The LNT and MST bases also have much more variety in the distribution of the widths over the cycles, and contain a significant number of cycles with widths up to five.

The MF and LNF cycle bases resemble the MCB cycle basis. They are both narrow and relatively short, and also consist mostly of cycles with widths zero and one.

The restricted BFS algorithms yield cycle bases that are wider than the above described cycle bases. The lengths of the restricted BFS cycle bases lie between those of the LNT and MST on one hand, and the MCB, LNF, and MF cycle bases on the other hand. With respect to the distribution of the widths of the cycles, they resemble the LNT and MST cycle bases.

Finally, the basic BFS algorithms yield a second somewhat surprising result.

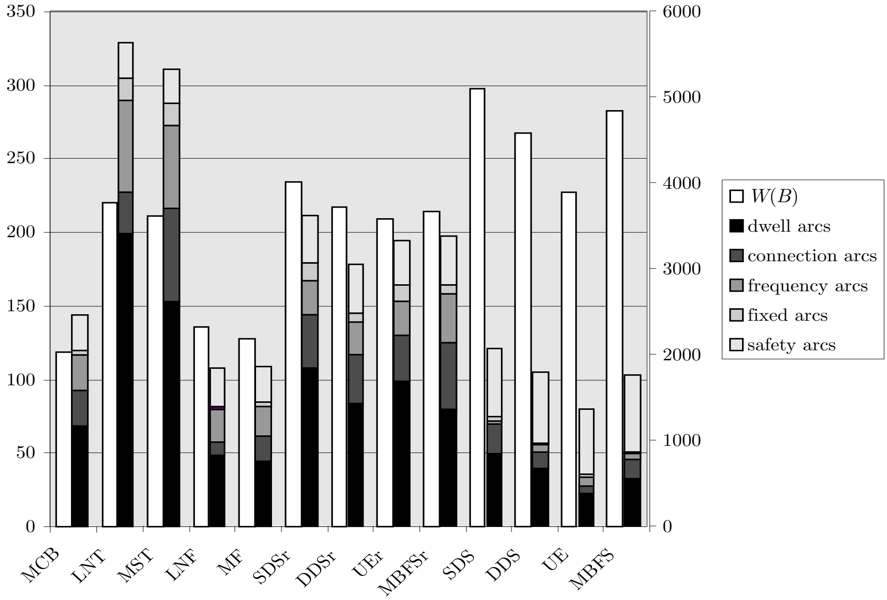


Figure 7.3: Cycle bases widths and lengths for the IC97 instance.

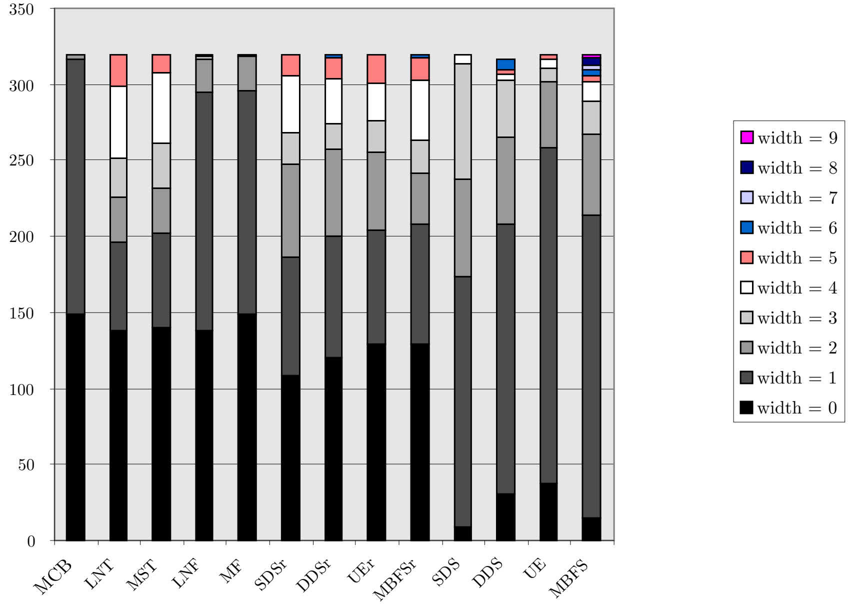


Figure 7.4: Widths distribution of the basis cycles for the IC97 instance.

By combining a small length with a large width, they show that the length and the width of a cycle basis are not necessarily positively correlated. Moreover, they mainly consist of cycles with widths up to three. Table D.1 for the NH97 instance shows that the basic BFS cycle bases may be less wide than their restricted counterparts.

As for the arc composition of the cycle bases, we see that most cycle bases contain about the same number of safety arcs. Only the basic BFS cycle bases contain more safety arcs, which is explained by their focus on high degree nodes. Further, the length of the longer cycle bases is mostly accounted for by dwell arcs and frequency arcs, and, to a lesser extent, by connection arcs. This arc composition explains how the LNT and MST cycle bases combine a large length with a relatively small width, since dwell arcs, frequency arcs, and connection arcs typically have narrow time windows.

7.5 Comparing the Cycle Bases for Solving the CPF

The goal of studying various cycle bases for the CPF is to improve the solution process. We argued in Section 5.7 that small width cycle bases are likely to solve the CPF faster, since they yield a smaller solution space. To test this statement, we formulated and solved the CPF for each of the computed cycle bases. For this test, we used the objective functions F_t for linearly minimizing the passenger travel time, and $F_{t,q}$ for quadratically minimizing the passenger travel time.

In an initial set of computational experiments, we found that the wide cycle bases constructed by the basic BFS algorithm by Deo et al. perform very poorly in solving the model. Their computation times exceeded our 3600 seconds time limit for the instances IC97 and NH97. Therefore, these algorithms are not reported in this section.

The results for the remaining cycle bases, namely MCB, LNT, MST, LNF, MF, and the restricted BFS cycle bases, are discussed below. The tests yielded the following optimal solution values z_{opt}:

$$\begin{aligned}
z_{\text{opt}} &= 3942 \text{ for the objective function } F_t \text{ for IC97,}\\
z_{\text{opt}} &= 154 \text{ for the objective function } F_{t,q} \text{ for IC97,}\\
z_{\text{opt}} &= 1418 \text{ for the objective function } F_t \text{ for NH97,}\\
z_{\text{opt}} &= 48 \text{ for the objective function } F_{t,q} \text{ for NH97.}
\end{aligned}$$

For the linear objective functions, these solution values equal the total travel time of the trains in the system. For the quadratic objective function, the solution values

	Time(s)	Nodes	Root LP+cuts	Root LP
MCB	659	240088	3924	3892
LNT	39	12618	3919.26	3893
MST	31	9080	3931.12	3893
LNF	388	137576	3929.21	3891
MF	47	17580	3932.66	3893
SDSr	64	25715	3917.9	3889
DDSr	64	25941	3924.47	3891
UEr	44	18561	3924.43	3891
MBFSr	62	29246	3915.77	3883

Table 7.5: Solving the IC97 instance for various cycle bases, with the linear passenger travel time objective F_t.

	Time(s)	Nodes	Root LP+cuts	Root LP
MCB	1087	173280	76.98	25
LNT	113	12204	89.44	26
MST	152	15830	88.78	26
LNF	985	114322	97.61	24
MF	169	22865	110.6	26
SDSr	197	24280	103.99	21
DDSr	140	19312	103.98	24
UEr	131	17299	103.98	24
MBFSr	139	17589	100.84	16

Table 7.6: Solving the IC97 instance for various cycle bases, with the quadratic passenger travel time objective $F_{t,q}$.

measure, quadratically, how much the dwell and connection tension variables $x_a \in A^d \cup A^c$ exceed their time window lower bounds l_a.

The computational results for the IC97 instance are shown in Tables 7.5 and 7.6. The results for the NH97 instance can be found in Tables D.3 and D.4 in Appendix D. Each table shows the computation time in seconds, the number of nodes in the Branch&Cut tree, the root LP relaxation value after adding violated cuts by CPLEX, and the root LP relaxation value without these cuts. Note that, for the NH97 instance, the LP relaxation value equals the optimal solution value. We were not able to obtain integer feasible solutions for the ICIR97 instance in a reasonable amount of time.

The MCB cycle basis performs unexpectedly poorly in solving IC97 and NH97. The wider LNT and MST cycle bases solve both instances much faster than the small width MCB cycle basis. Moreover, for the IC97 instance, the restricted BFS cycle bases perform quite well in solving the CPF. In combination with the above mentioned poor performance of the basic BFS cycle bases, we conclude that a small width cycle bases does not guarantee fast computation times. In fact, the small width of the MCB cycle basis prevents it from containing long cycles. And the tables suggest a positive influence of the cycle basis length on computation times. Because of the larger length of the LNT, MST, and restricted BFS cycle bases, there is a lot of overlap between the cycles in the basis. Therefore, fixing an integer variable q_C in the Branch&Cut process has more influence on the other, not yet fixed, cycles, than in a short cycle basis. This positive effect seems to be stronger than the theoretical disadvantage of a larger width.

7.6 The Robustness Objective Function

This section reports on the computational results for the robustness objective function F_r, which was described in Section 3.7.2. Based on the results in the previous section, we only used the LNT and MST cycle bases for our robustness computations. As with the passenger travel time objectives from the previous section, we were not able to obtain an integer feasible solution for the ICIR97 instance in a reasonable amount of computation time. Therefore, this section only reports on the IC97 and NH97 instances.

We used a quadratic robustness objective function. Recall that the objective function F_r penalizes values in the safety time windows which are close to the time window bounds. Safety time windows are quite wide, since the headway time $h = 3$ in our instances, with cycle time $T = 60$ minutes. Therefore, we used Remark 3.2 with $\lambda_{\max} = 2$. So, the approximation of the quadratic function is exact for the subwindow $[\mu_a - \lambda_{\max}, \mu_a + \lambda_{\max}]_T$ of a safety time window $[l_a, u_a]_T$. For the remainder of the safety time window, we use the last linear approximation.

Table 7.7 shows the results for the robustness experiments. The table displays the same information as the previous tables. It is clear that the robustness objective takes more time to solve than the passenger travel time objective. Indeed, we were not able to obtain an integer optimal solution for the NH97 instance within a 3600 seconds time limit. Table 7.1 indicates an explanation for these results. The instances contain more safety arcs than dwell arcs and connection arcs. This especially holds for the NH97 instance. Therefore, there are more variables that influence the objective function F_r than for the objective function F_t. Moreover, these variables correspond to wide safety time windows, and thus have a quite wide range.

	Time(s)	Nodes	Best integer	Root LP+cuts	Root LP
IC97					
LNT	3385	359110	1	0	0
MST	1730	355695	1	0	0
NH97					
LNT	*	176743	137	0	0
MST	*	268770	149	0	0

Table 7.7: Solving the IC97 and NH97 instances for the robustness maximization objective function F_r (* = 3600 seconds time limit or 1 million nodes limit exceeded).

	Time(s)	Nodes	Root LP	Root LP+cuts
MST	3.1	937	47.75	50
LNT	5.66	1628	50.1	52

Table 7.8: Minimizing the rolling stock objective function F_s for the IC97 instance.

7.7 The Rolling Stock Objective Function

We applied the rolling stock objective function F_s to the IC97 and ICIR97 instances, but were not able to solve the ICIR97 instance within our one hour time limit. The rolling stock objective was not applied to the NH97 instance, since this instance considers a part of the network only. Therefore, it does not contain the termini for most trains, and misses the turn around constraints.

We used the basic version of F_s for the case in which each train turns on itself, as described in Section 6.4.1. The international trains were excluded from the objective. We solved the IC97 instance with the MST and LNT cycle bases, since these cycle bases yielded the best results for the passenger travel time objective. The optimal solution for the IC97 instance consists of 55 rolling stock composition, and this solution is computed in a few seconds. For the rolling stock objective, Table 7.8 shows the same information as before.

7.8 The Constraint Violation Objective Function

As was described in Section 7.1, the IC97 and ICIR97 instances are defined too tightly, which causes them to be infeasible. For the computations in the previous sections, we adjusted the upper bounds of the dwell time windows and connection time windows, so as to obtain feasible instances.

Let us describe this adjustment in more detail. For all dwell and connection arcs

$a \in A^d \cup A^c$, we introduced a new time window $[l'_a, u'_a]$ defined by

$$\begin{aligned} l'_a &= l_a, \\ u'_a &= \begin{cases} u_a + 5 \text{ if } a \in A^d, \\ u_a + 2 \text{ if } a \in A^c. \end{cases} \end{aligned}$$

After this relaxation, both the IC97 and the ICIR97 instance were feasible.

Therefore, the quadratic objective function $F_{t,q}$ from Section 7.5 can be interpreted as the constraint violation objective F_v for the extra space in the relaxed time windows $[l'_a, u'_a], a \in A^d \cup A^c$. The only difference between $F_{t,q}$ and F_v is that $F_{t,q}$ also penalizes values in the original time window $[l_a, u_a]$, whereas F_v only penalizes values between u_a and u'_a. However, since $F_{t,q}$ is a quadratic function, values between u_a and u'_a are penalized much heavier than values in the original time window. Therefore, we shall use the function $F_{t,q}$ to illustrate the constraint violation objective function F_v.

A closer inspection of the results for the IC97 instance with the objective function $F_{t,q}$ yields that, in the optimal solution, only two dwell arcs use the extra offered space in the relaxed time windows $[l'_a, u'_a]$, and both arcs have an optimal solution value $x_a = u_a + 2$. All other dwell arcs and connection arcs, 197 in total, do not use the extra offered space. Moreover, of these 197 arcs, three dwell arcs and two connection arcs take the optimal solution values $x_a = u_a$.

This result shows the benefit of the constraint violation approach. Rather than iteratively trying to change some time windows in order to make the instance feasible, we relaxed all the commercial constraints in the instance. Subsequently, the solution of the CPF for the relaxed instance immediately indicates that only two dwell time windows need to be relaxed by two minutes in order to arrive at a feasible timetable.

7.9 Adding Cutting Planes

Next, we further investigated the MST and LNT cycle bases by adding cycle cuts and change cycle cuts to the CPF (see Section 5.5.1). First, since Horton's algorithm selects its basis cycles greedily from a list of candidates, this candidates list is filled with cycles C for which a_C is close to b_C. From these candidates, we selected all non-basis cycles D with $b_D - a_D \leq 1$, and added the cycle cuts for them to the formulation. Second, we added all safety triangles (see Sections 5.4.2 and 5.4.4), which also have width at most one. Third, we added the change cycle cuts for all basis cycles.

	Time (s)	Nodes	Root LP+cuts	Root LP
F_t ($z_{\text{opt}} = 3942$)				
LNT	36.32	11983	3922.42	3843.00
LNT+saf	34.71	8516	3922.43	3843.00
LNT+Horton	26.27	4295	3922.48	3843.00
LNT+change	40.13	10250	3922.06	3845.80
LNT+all	51.48	5957	3922.06	3845.80
MST	31.10	9254	3926.12	3843.00
MST+saf	23.69	5135	3926.65	3843.00
MST+Horton	16.12	2545	3926.65	3843.00
MST+change	19.54	4609	3924.85	3845.72
MST+all	45.06	5088	3925.65	3845.72
$F_{t,q}$ ($z_{\text{opt}} = 154$)				
LNT	93.76	11029	89.44	26.00
LNT+saf	168.05	17101	89.42	26.00
LNT+Horton	56.10	5220	90.73	26.00
LNT+change	109.78	10008	94.05	29.83
LNT+all	128.93	7554	96.02	29.83
MST	126.64	13737	91.44	26.00
MST+saf	102.90	12544	92.52	26.00
MST+Horton	62.07	5427	93.51	26.00
MST+change	229.99	22166	99.93	29.59
MST+all	141.50	9898	98.82	29.59
F_r ($z_{\text{opt}} = 1$)				
LNT	4706.95	*	0	0
LNT+saf	290.83	51655	0	0
LNT+Horton	33.96	3466	0	0
LNT+change	2600.03	445989	0	0
LNT+saf+Horton+change	36.99	3119	0	0
MST	1729.62	355695	0	0
MST+saf	782.85	136950	0	0
MST+Horton	6085.68	694728	0	0
MST+change	152.78	26123	0	0
MST+all	828.10	70466	0	0

Table 7.9: MST and LNT cycle basis with safety triangle, Horton, and change cycle cuts, for the IC97 instance.

Table 7.9 shows the results of adding the safety triangle cuts and the Horton cuts for the IC97 instance. The results for the NH97 instance can be found in Table D.5. The safety triangle cuts are indicated by '+saf', the Horton cuts by '+Horton', and the change cycle cuts by '+change'. Moreover, we added all three cut types, which is indicated by '+all'. The basic instance without cuts is also included in the table. Further, the table shows the same information as the previous tables. Table D.5 also shows the best integer solution for the NH97 instance with objective function F_r, since we were not able to solve this instance to optimality.

For the IC97 instance, the computation times are clearly reduced in most cases by adding the safety triangle, Horton, and change cycle cuts. The same holds for the number of Branch&Cut tree nodes. Surprisingly, the root LP value only changes for the change cycle cuts. However, after CPLEX has added additional violated cuts, the LP value does change for all cut types, and the overall effect of the safety cuts and Horton cuts is positive. For the NH97 instance, however, the solution time only increases in about half the cases, even though the number of Branch&Cut tree nodes decreases.

7.10 The Cycle Fixation Heuristic

We were not able to solve the ICIR97 instance with the CPF. Therefore, we applied the cycle fixation heuristic from Section 6.5 to this instance. We used a two-stage approach. In the first stage, the constraint graph consisted only of the intercity network. That is, the first stage consisted of solving the IC97 instance, and the previous sections showed that this can be done quite quickly with the CPF. As cycle fixation set, we chose all safety triangles in the IC97 instance, which means that the cyclic sequences of the intercity trains on the tracks were fixed. Having fixed the cycle periodicity integer variables for the safety triangles, we solved the complete instance in the second stage.

We tested the cycle fixation heuristic with the linear and quadratic passenger travel time objectives F_t and $F_{t,q}$, for both the MST and LNT cycle bases. Moreover, we further tested the effectiveness of the cutting planes from the previous section by adding them to the CPF for the second stage of the cycle fixation heuristic.

The results for the first stage of the heuristic are simply the results that were presented before for the IC97 instance. We solved the second stage CPF with CPLEX, with a time limit of 3600 seconds. The results for the second stage are shown in Figures 7.5 and 7.6. The figures show how the value of the best found integer feasible solution changes over the 3600 seconds time limit. The bottom-most dashed line in both figures represents the best LP relaxation value that was found in the test.

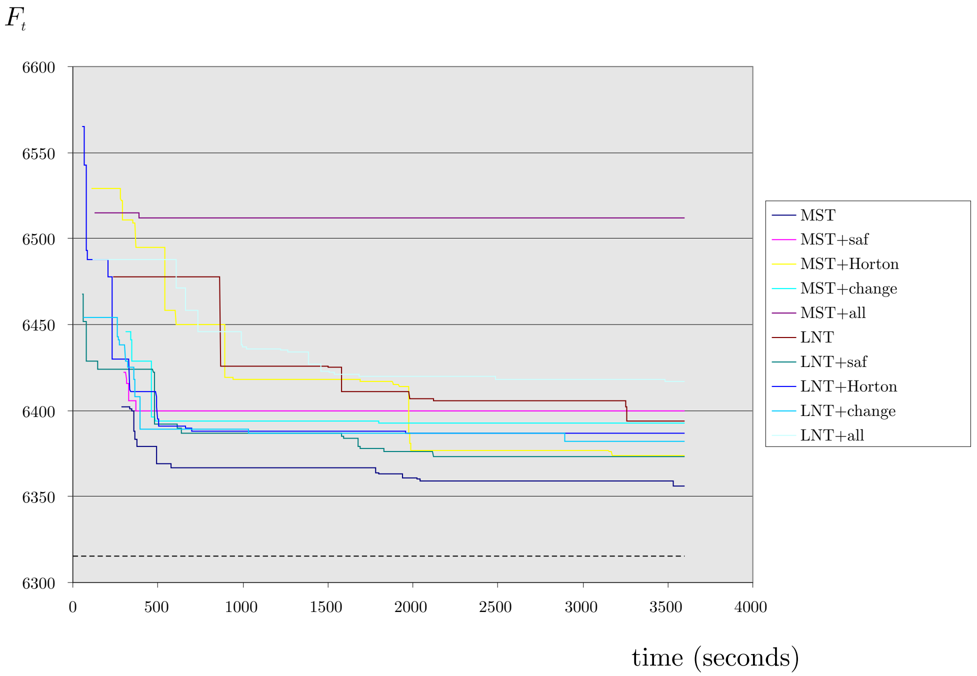


Figure 7.5: Solving the second stage CPF for the ICIR97 instance with objective F_t.

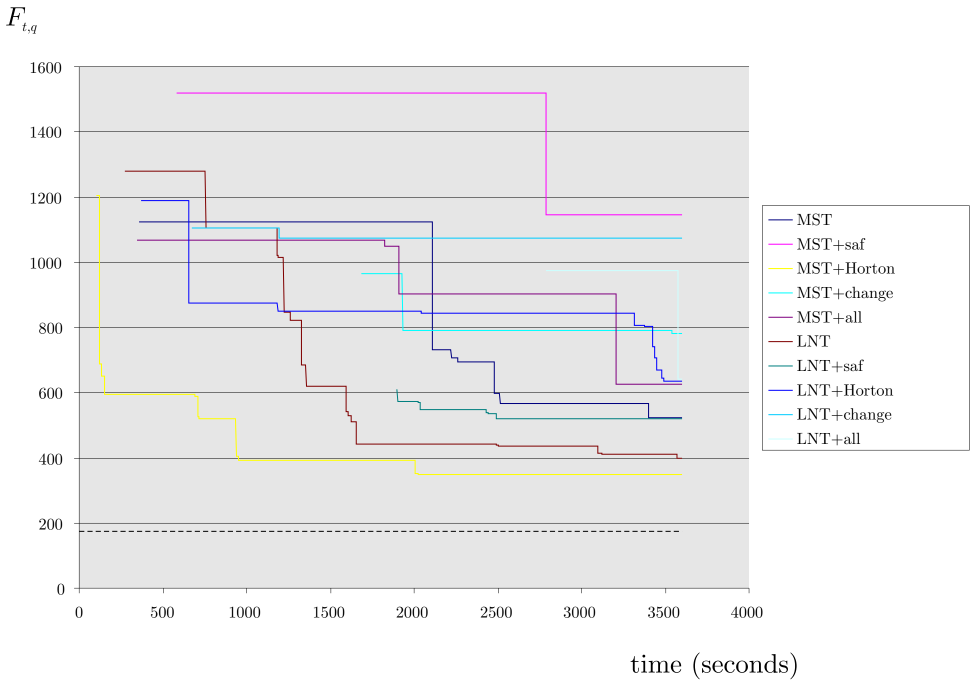


Figure 7.6: Solving the second stage CPF for the ICIR97 instance with objective $F_{t,q}$.

	Time(h)	Nodes (mln)	Best Integer	Best LP
F_t				
LNT	18.3	6.6	6347	6315.1
MST	22.6	9.7	6340	6323.8
$F_{t,q}$				
LNT	13.3	2.9	344	170.54
MST	24.1	4.1	399	180.1

Table 7.10: Solving the second stage CPF of the cycle fixation heuristic for the ICIR97 instance.

The figures show that we were not able to solve the second stage CPF to optimality within the 3600 seconds time limit. We did, however, obtain quite a lot of integer feasible solutions. In the next section, we compare these solutions with the solutions obtained by the CADANS algorithm. There, it is shown that the quality of the best found second stage solution is actually quite good.

The question arises how the solution process proceeds after the first hour of computation time. In order to answer that question, we solved the second stage CPF for both passenger travel time objectives, with the MST and LNT cycle bases, on a Pentium III 1.2 Ghz machine, with 2 GB memory. The solution process was executed until all available memory was used. Table 7.10 shows the results for this experiment. The first column shows the computation time when the memory ran out, the second column the number of nodes in the Branch&Cut tree, the third column the best found integer feasible solution, and the fourth column the best found LP relaxation value. For none of the instances, the second stage CPF could be solved to optimality within the 2 GB memory limit. After 13 to 24 hours of computation time, CPLEX ran out of memory. The table shows that, at this point, the gap between the LP relaxation and the best found integer feasible solution is still not closed. However, the best found integer feasible solutions are again quite a bit better than the solutions obtained after one hour of computation time.

Finally, we also applied the cycle fixation heuristic for the rolling stock objective function F_s. However, for this objective, the heuristic fails to find an integer feasible solution for the second stage CPF within 3600 seconds of computation time.

7.11 A Comparison to CADANS

Finally, we solved each of the three instances using the CADANS algorithm (Schrijver and Steenbeek, 1994). CADANS solved each instance quite quickly, in two to five seconds time. We also carried out a post-optimization step, which consists of

	IC97		NH97		ICIR97	
	CADANS	z_{opt}	CADANS	z_{opt}	CADANS	z_{best}
F_t	4422	3942	1424	1418	8570	6340
$F_{t,q}$	203	154	66	48	643	344
F_r	1	1	111	137*	160	-
F_s	64	55	-	-	145	-

Table 7.11: Solution values of the CADANS solutions for the IC97, NH97, and ICIR97 instances (* best found solution).

computing the cycle periodicity variables corresponding to the CADANS solution, substituting those variables into the CPF, and solving the remaining LP problem. The latter took less than a second.

The results are shown in Table 7.11. For the IC97 and NH97 instances, the table also shows the optimal solution values that we obtained by solving the CPF. For the ICIR97 instance, the best solution found by the cycle fixation heuristic is shown.

CADANS solves the problem of finding a feasible solution, and does not take an objective function into account. Therefore it is not surprising that, in most cases, our solutions are better than those obtained by CADANS. Most notably, our solutions are much better for the passenger travel time objective for the IC97 and ICIR97 instances. As was shown is the previous sections, it takes more computation time to find these better solutions.

Whenever CADANS fixes a certain variable v_i, there may be multiple values left to choose from. It then selects the value that, roughly speaking, corresponds to the middle of the time windows that are adjacent to node i. This may explain why, for the robustness objective, the CADANS solution for the IC97 instance is optimal, and why the CADANS solution for the NH97 instance is better than the best solution we obtained with the CPF in one hour.

7.12 Summary of the Computational Results

In this chapter, we reported on the performance of the models and the solution techniques in this thesis in optimizing three real-life cyclic railway timetabling instances that were obtained from NSR and Railned.

As a first result, we found that it is crucial to preprocess the constraint graphs, since this yields a considerable reduction of the sizes of the three constraint graphs. Also, the periodic time window strengthening algorithm showed to be quite effective in shrinking some of the time windows in the instances.

Next, we tested the PESP formulation for the CRTP. The results of these tests were a bit disappointing, since we were not able to solve the PESP formulation to optimality within a one hour time limit for any of our test instances. Further tests showed that the PESP formulation is not able to solve any of the instances to optimality in several hours of computing time, after which all available memory had been used by the Branch&Cut process.

Therefore, the remainder of the chapter tested various aspects of the CPF model for the CRTP. To that end, we started with comparing the cycle bases obtained by the various cycle basis algorithms presented in Section 5.7. We surprisingly found that, for all three test instances, the MCB algorithm yields a fundamental cycle basis. Less surprisingly, these cycle bases are the least wide cycle bases that we obtained. In their basic form, the BFS algorithms construct quite wide cycle bases. After adjusting these algorithms by forbidding safety arcs to be included in the spanning tree, the width of their cycle bases was reduced considerably. The MST and LNT algorithms construct cycle bases that are wider than the MCB and BFS cycle bases. The FCB algorithm reduced the width of each of the strictly fundamental cycle bases constructed by the BFS, MST, and LNT algorithms.

Next, we compared the performance of the various cycle bases for solving the CPF with the travel time objective functions F_t and $F_{t,q}$, for the IC97 and NH97 instances. Quite surprisingly, the MST and LNT cycle bases outperformed the other cycle bases in solving the CPF, even though the MST and LNT cycle bases are much wider. This surprising result may be explained by the fact that the MST and LNT cycle bases are much longer than the other cycle bases. Overall, we were able to compute timetables with minimum travel times for the IC97 and NH97 instances within 30 seconds to one hour computation time, depending on the quality of the cycle basis. We were not able to solve the ICIR97 instance to optimality with any of the cycle bases within the one hour time limit.

The quadratic robustness objective function F_r turned out to be much harder to solve. For the IC97 instance, it took between half an hour and one hour to solve the CPF with the objective function F_r. We were not able to compute optimal robust timetables for the NH97 within one hour, and the quality of the timetables that we did obtain was rather poor. The hardness of the robustness objective function may be explained by its dependence on the safety arcs, which are quite wide, and form the majority of the arcs in our test constraint graphs.

The rolling stock objective function F_s, on the other hand, was much easier to solve. We computed rolling stock optimal timetables within a few seconds of computation time for the IC97 instance. This may be explained by the fact that the objective function F_s is stated in terms of the integer variables q_C, whereas the other objective functions all depend on the tension variables x_a. The rolling stock objective function is therefore the only objective function that directly contains the integer variables. Since the NH97 instance considers only a part of the Netherlands, and therefore only parts of some of the involved train lines, it makes no sense to minimize the required number of rolling stock compositions for this instance.

The constraint violation objective function F_v was illustrated through its similarity to the quadratic travel time objective function $F_{t,q}$. For the IC97 instance, which was initially infeasible, the constraint violation objective function showed that two dwell time windows in the instance need to be relaxed with two minutes each, in order to obtain a feasible timetable. Moreover, this result was obtained in between two and 22 minutes of computation time. This may be much faster than the rather cumbersome process of manually detecting the constraints that need to be relaxed.

Next, we tested the cutting planes of Section 5.5 for the MST and LNT cycle bases. Strangely, the value of the root LP relaxation hardly changes when we add the cutting planes. However, the cutting planes do strengthen the CPF formulation for the IC97 instance such that CPLEX is able to add other cutting planes, which do improve the root LP relaxation value. For the NH97 instance, the root LP relaxation value without cuts already equals the optimal solution value. Still, also for this instance, the added cuts help in pruning the Branch&Cut tree, and therefore reduce the computation times. We conclude that the overall effect of adding cutting planes to the CPF is a clear reduction of the computation times for the IC97 and NH97 instances.

We were not able to solve the large ICIR97 instance in a reasonable amount of computation time in any of the above described experiments. Therefore, we applied our cycle fixation heuristic to the ICIR97 instance for computing timetables with minimum travel times. Although the heuristic is not able to solve the resulting second stage CPF to optimality within a time frame of several hours, it does find quite good solutions quite quickly. We therefore conclude that the cycle fixation heuristic is a promising candidate for the practical optimization of cyclic railway timetables.

Finally, we compared the quality of our solutions with the quality of the solutions that were obtained by CADANS (Schrijver and Steenbeek, 1994), the algorithm that is currently in use at NSR and Railned for the strategic planning of cyclic railway timetables. CADANS is quite fast in practice, but does not take an objective function into account, and returns the first feasible timetable that it finds. The aim of the comparison was therefore not to show that our methods find higher quality timetables, but rather to gain insight into the trade-off between the quality of the obtained timetables, and the required computation time. We conclude that the quality of the timetables obtained by our methods is quite a bit higher than the quality of the

CADANS timetables. With respect to the robustness objective, however, CADANS finds quite high quality timetables, which we explain by the algorithm's search strategy. The price for our higher quality timetables is an increase in the required computation time. Solving the CPF takes between a few seconds and an hour, depending on the objective function, whereas CADANS solves all three instances within a few seconds.

Chapter 8

Conclusions

An increasing demand for mobility, the introduction of competition in the European railways, and performance and punctuality problems have made the railways in the Netherlands and their timetables an often discussed topic. We argued in Chapter 1 that optimization methods for constructing cyclic railway timetables form a valuable tool for both railway operators and railway infrastructure managers. The research in this thesis therefore provides aid in solving the problems caused by the still growing mobility and the resulting congested railways.

This concluding chapter first presents an overview of the main results of the thesis. We next discuss how these results provide answers to the research questions posed in Chapter 1. Finally, we reflect on the limitations of the thesis, and propose some directions for further research on optimizing cyclic railway timetables.

8.1 Main Results

In discussing the main results of the thesis, we distinguish between results regarding the modeling of cyclic railway timetables, results with respect to the cycle periodicity formulation, and results for cycle bases for the cycle periodicity formulation.

Modeling Cyclic Railway Timetables. Having sketched the organizational environment for constructing railway timetables in Chapter 2, Chapter 3 subsequently described the requirements that a railway timetable should meet, and provided extensive examples for modeling these requirements using periodic time window constraints. In particular, the modeling of synchronization and safety requirements, and the structure of the model's constraint graph representation were discussed in detail. As such, we believe to have provided a valuable contribution, since most literature describes the modeling of cyclic timetables only briefly, and without going into details.

Chapter 3 further presented the linear objective functions of minimizing passenger travel times, maximizing timetable robustness, and minimizing the amount of constraint violation for infeasible instances, and showed how to approximate a quadratic variant of these objectives. Later, in Chapter 6, we also introduced an objective function for minimizing the required number of rolling stock compositions. From the computational tests in Chapter 7, we conclude that these objective functions yield a substantial increase in the quality of the obtained timetables, compared to models and methods that only consider the feasibility of timetables. The computational tests further showed that the constraint violation objective is quite useful for handling infeasible problem instances. It may however take quite some time to compute an optimal timetable, but we argued that the increase in timetable quality justifies the possibly large computation times.

Several extensions to the basic model from Chapter 3 were proposed in Chapter 6. The extensions for incorporating variable train trip times and flexible connections into the model use periodic constraints only, and can therefore be added to the basic model without disturbing its structure. We also proposed a station capacity extension, which allows for limiting the number of trains concurrently present in a network node.

The Cycle Periodicity Formulation. Chapter 4 described the PESP, an existing model in literature which generalizes our cyclic railway timetabling model. Chapter 5 introduced periodic tensions as a natural counterpart of periodic constraints. Using these periodic tensions, the PESP was transformed into the Cycle Periodicity Formulation (CPF). Whereas the PESP is formulated in terms of arrival and departure times, which are related through periodic time window constraints, the CPF is based on process times, and on the periodicity of cycles in the constraint graph. The computational experiments in Chapter 7 showed that the PESP formulation performs quite poorly, and that the CPF is the superior mathematical formulation for optimizing cyclic railway timetables.

Using the sequencing corollary presented in Chapter 4, Chapter 5 explored the relation between certain cycles in the constraint graph, and the sequence in which trains travel along tracks. Based on these train sequences, we derived some cutting planes for the CPF, which were shown to belong to the class of so-called cycle cuts. In most of our experiments in Chapter 7, computation times could be reduced by adding these cycle cuts, and change cycle cuts, to the CPF.

In Chapter 6, we described a cycle fixation heuristic for solving the CPF. The heuristic was inspired by the typical practical process of first constructing a timetable for the most important trains, and then iteratively adding other classes of trains, while changing the timetable for the previous train classes as little as possible. Based on this approach, and on the train sequencing result from Chapter 5, we proposed a cycle fixation heuristic that iteratively adds classes of trains, while keeping the train sequences of the previously scheduled train classes fixed. Within an hour of computation time, the cycle fixation heuristic yielded quite good timetables for our largest test instance, which we were not able to solve using exact solution methods.

Cycle Bases for the CPF. In principle, cycle periodicity has to be required in the CPF for every single cycle, and a constraint graph may contain exponentially many cycles. In Chapter 5 we showed, however, that it suffices to require cycle periodicity for the cycles in an integral basis of the cycle space of the constraint graph only. Moreover, an example illustrated that a non-integral cycle basis may return a feasible solution to the CPF, which is in fact an incorrect timetable.

Several aspects of cycle bases for the CPF were explored in Chapter 5. It was shown that strictly fundamental and fundamental cycle bases are integral, and that they can thus be used for formulating the CPF. We argued that algorithms for the minimum cycle basis problem can be used as heuristics for constructing good cycle bases for the CPF. Based on the specific structure of the constraint graph, two minimum spanning trees weight functions were proposed for computing good strictly fundamental cycle bases. We also described how these strictly fundamental cycle bases can be improved by considering the corresponding regular fundamental cycle bases. Our computational tests in Chapter 7 show that it is essential to choose a good cycle basis for formulating and solving the CPF. More specifically, the MST and LNT cycle basis algorithms compute cycle bases that yield fast computation times for our objective functions.

8.2 Answering the Research Questions

Having summarized the results of the thesis, we now return to the research questions defined in Chapter 1.

> *What are the criteria for assessing the quality of a timetable, and how can they be modeled?*

Several criteria for evaluating timetables were discussed in Section 3.2. We distinguished between the criteria of customer satisfaction, timetable robustness, and cost control. We argued that these criteria may be conflicting, and delineated each of them with respect to our purposes and assumptions. Within our assumptions, the only variable for customer satisfaction is travel time, which should be reduced in order to improve customer satisfaction. Timetable robustness is also of importance for customer satisfaction. The robustness of a timetable can be improved by increasing the time buffer between trains that possibly interfere with one another in case of a disturbance. Although the decisions that influence the main cost components of a railway system are not taken at the timetabling level, one can still consider cost efficient timetables to some extent. We translated the cost criterium into reducing the number of rolling stock compositions that are required to operate a timetable. Sections 3.7 and 6.4 described how to model these criteria as objective functions in our cyclic railway timetabling model.

> *What adjustments need to be made to over-tightly defined timetable requirements in order to obtain a feasible timetable, and how can these adjustments be modeled?*

As an answer to this question, we proposed to relax the timetable requirements, or some subset of them. Then, one can use an objective function which aims at minimizing the violation of the initial requirements. The computational tests in Chapter 7 suggest that this method can be quite effective.

> *How can the models arising from the previous two questions be solved in a reasonable amount of time?*

We used periodic constraints for modeling cyclic railway timetables in a natural way. For solving the resulting model, we propose to use the transformed CPF. The thesis shows that the cycle basis used for formulating the CPF has a significant impact on the computation times. More specifically, we found the best performance for strictly fundamental cycle bases, stemming from a minimum spanning tree with respect to an arc weight function based on the time window widths and the structure of the constraint graph. We further found that the computation times can be reduced by adding cycle cuts to the CPF. With the use of a good cycle basis and cycle cuts, the two smaller test instances can be solved to optimality in computation times ranging from less than a minute to half an hour. However, a good cycle basis and cycle cuts do not suffice for solving our largest test instance in a reasonable amount of time. Therefore, we propose to use the cycle fixation heuristic for solving large instances.

The above answers to the sub-questions lead us to the following answer to the main research question.

> *How can mathematical models and solution methods support the construction of high quality cyclic railway timetables?*

We discussed the practical process of railway timetabling in Chapter 2. Keeping the practice of railway timetabling in mind, and having answered the three sub-questions above, we conclude that our cyclic railway timetabling model and the CPF can support in constructing good cyclic railway timetables, when applying them with good cycle bases and cycle cuts. For large timetabling problems, a heuristic such as the proposed cycle fixation heuristic will be required to yield solutions in a practically reasonable amount of time.

8.3 Limitations of the Thesis and Recommendations for Further Research

Since the thesis describes a mathematical model for cyclic railway timetable optimization, a clear limitation of our research is formed by the level of detail in the cyclic railway timetabling model. In our opinion, the current model adequately incorporates those timetable requirements that are of importance for tactical and strategic studies. Still, there will always be a desire by practitioners to further increase the level of detail. In particular, an increase in the level of detail would be required in order to extend the model towards the operational planning level.

A second limitation regards the practical applicability of the research, which is hampered by the large computation times for large timetabling problems, even when using a heuristic solution method. The fact that we have to resort to heuristics is in principle not a big problem, since the objective functions for our model are somewhat rough. Therefore, it does not make sense in a practical setting to spend a lot of computation time to search for the exact optimal solution. However, it is problematic that our cycle fixation heuristic takes very much time to solve the resulting second-stage CPF to optimality for large timetabling problems. Therefore, we see the further investigation of fast heuristics as an important direction for further research. More specifically, it would be interesting to investigate whether the good performance of the CADANS algorithm by Schrijver and Steenbeek (1994) can be combined with the optimization features of our methods.

On the same topic of improving the computational performance, a further study of the CPF would also be very useful. Here, our thoughts go in the direction of new classes of cutting planes, alternative relaxations that may yield good lower bounds in a Branch&Bound or Branch&Cut procedure, and methods for quickly finding good quality integer feasible solutions.

The above limitations and recommendations only consider the solution of the mathematical puzzle of constructing an optimal cyclic timetable. A different, but very interesting direction of further research is the study of the application of cyclic railway timetable optimization in a practical setting. This requires a clear graphical user interface, for example for specifying the type of objective function, and the objective function weights. With respect to the further specification of an instance, a user interface similar to the one of DONS, discussed in Section 2.5, could be used. In the same direction, the visualization of constraint graphs, such as presented in Appendix C.4, and of obtained timetables, through tables and time-space diagrams, could further support timetable planners in understanding the bottlenecks and interdependencies in a timetabling problem instance.

Bibliography

Ahuja, R., Magnanti, T., Orlin, J. (1993). *Network flows: theory, algorithms, and applications*. Prentice Hall, Englewood Cliffs.

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

AVV (Adviesdienst Verkeer en Vervoer) (2000). *Mobiliteit verkend – Feiten achter het NVVP (Mobility explored – Facts behind the NVVP)*. In Dutch.

AVV (Adviesdienst Verkeer en Vervoer) website (2002). `http://www.rws-avv.nl/vv2020/`. In Dutch.

Berger, F. (2002). *Personal communication*.

Bollobás, B. (1998). *Modern graph theory*, volume 184 of *Graduate Texts in Mathematics*. Springer, Berlin.

Brännlund, U., Lindberg, P., Nõu, A., Nillson, J.-E. (1998). *Railway timetabling using lagrangian relaxation*. Transportation Science, 32(4):358–369.

Burkard, R., Çela, E., Pardalos, P., Pitsoulis, L. (1998). *The quadratic assignment problem*. In Du, D., Pardalos, P. (editors), *Handbook of Combinatorial Optimization*, volume 2, pages 241–337. Kluwer Academic Publishers, Dordrecht.

Bussieck, M., Winter, T., Zimmermann, U. (1997). *Discrete optimization in public rail transportation*. Mathematical Programming, 79(3):415–444.

Caprara, A., Fischetti, M., Guida, P., Monaci, M., Sacco, G., Toth, P. (2001). *Solution of real–world train timetabling problems*. In *Proceeding of HICSS 2001*. `http://www.hicss.hawaii.edu/diglib.htm`.

Caprara, A., Fischetti, M., Toth, P. (2002). *Modeling and solving the train timetabling problem*. Operations Research, 50(5):851–861.

Çela, E. (1998). *The quadratic assignment problem: theory and algorithms*. Kluwer Academic Publishers, Dordrecht.

Cordeau, J.-F., Toth, P., Vigo, D. (1998). *A survey of optimization models for train routing and scheduling*. Transportation Science, 32(4):380–404.

CPLEX website (2002). `http://www.ilog.com/products/cplex/`.

Czech, Z., Konopka, M., Majewski, B. (1993). *Parallel algorithms for finding a sub-optimal fundamental cycle set in a graph*. Parallel Computing, 19:961–971.

Daduna, J., Voss, S. (1995). *Practical experiences in schedule synchronization*. In Daduna, J.R., Branco, I., Paixão, J. (editors), *Computer Aided Transit Scheduling*, number 430 in Lecture Notes in Economics and Mathematical Systems. Springer, Berlin.

DEO, N. (1982). *Graph theory with applications to engineering and computer science*. Series in Automatic Computation. Prentice–Hall, Englewood Cliffs.

DEO, N., KUMAR, N., PARSONS, J. (1995). *Minimum length fundamental cycle set: new heuristics and an empirical study*. Congressus Numerantium, 107:141–154.

DEO, N., PRABHU, G.M., KRISHNAMOORTHY, M.S. (1982). *Algorithms for generating fundamental cycles in a graph*. ACM Transactions on Mathematical Software, 8:26–42.

DOMSCHKE, W. (1989). *Schedule synchronization for public transit networks*. OR Spektrum, 11:17–24.

EGYHAZY, C. (1985). *An algorithm for generating the minimum length fundamental cycles in a graph*. Congressus Numerantium, 50:219–230.

FORD, R., HAYDOCK, D. (1992). *Signalling and timetabling*. In HARRIS, N., GODWARD, E. (editors), *Planning Passenger Railways: A Handbook*, chapter 11, pages 117–129. Transportation Publishing Company, Glossop.

GAREY, M., JOHNSON, D. (1979). *Computers and intractability – a guide to the theory of NP-completeness*. Freeman and Company, New York.

GERTSBAKH, I., SERAFINI, P. (1991). *An interactive approach based on the periodic event scheduling problem and the deficit function approach*. European Journal of Operational Research, 50:298–309.

GLEISS, P. (2001). *Short cycles – minimum cycle bases from chemistry and biochemistry*. Ph.D. thesis, University of Vienna, Vienna, Austria.

GOVERDE, R. (1999). *Improving punctuality and transfer reliability by railway timetable optimization*. In BOVY, P. (editor), *Proceedings TRAIL 5th Annual Congress*, volume 2. Delft University Press, Delft.

GOVERDE, R., SOTO Y KOELEMEIJER, G. (2000). *Performance evaluation of periodic railway timetables: Theory and algorithms*. Studies in Transportation Science 2, TRAIL, Delft, The Netherlands.

GRÖTSCHEL, M., JÜNGER, M., REINELT, G. (1984). *A cutting plane algorithm for the linear ordering problem*. Operations Research, 32(6):1195–1220.

——— (1985). *Facets of the linear ordering polytope*. Mathematical Programming, 33:43–60.

HARTVIGSEN, D., ZEMEL, E. (1989). *Is every cycle basis fundamental?*. Journal of Graph Theory, 13(1):117–137.

HASSIN, R. (1996). *A flow algorithm for network synchronization*. Operations Research, 44(4):570–579.

HOOGHIEMSTRA, J. (1996). *Design of regular interval timetables for strategic and tactical railway planning*. In ALLAN, J., BREBBIA, C.A., HILL, R.J., SCIUTTO, G. (editors), *Computers in Railways V*, volume 1, pages 393–402. WIT Press, Ashurst.

HOOGHIEMSTRA, J., KROON, L., ODIJK, M., SALOMON, M., ZWANEVELD, P. (1999). *Decision support systems support the search for win-win solutions in railway network design*. Interfaces, 29(2):15–32.

HORTON, J. (1987). *A polynomial-time algorithm to find the shortest cycle basis of a graph*. SIAM Journal on Computing, 16(2):358–366.

HURKENS, C. (1996). *Een polyhedrale aanpak van treinrooster problemen (A polyhedral approach to railway timetabling problems)*. Unpublished. In Dutch.

KROON, L., PEETERS, L. (2003). *A variable trip time model for cyclic railway timetabling*. Transportation Science. Forthcoming.

KROON, L., ROMEIJN, H., ZWANEVELD, P. (1997). *Routing trains through railway stations: complexity issues*. European Journal of Operational Research, 98(3):485–498.

LICHTENEGGER, M. (1990). *Der taktfahrplan (The periodic timetable)*. Ph.D. thesis, Technical University Graz, Graz, Austria. In German.

LIEBCHEN, C. (1998). *Optimierungsverfahren zur Erstellung von Taktfahrplänen (Optimization methods for constructing periodic timetables)*. Master's thesis, Technical University Berlin, Berlin, Germany. In German.

LIEBCHEN, C., PEETERS, L. (2002a). *On cyclic timetabling and cycles in graphs*. Technical Report 761/2002, Institut für Mathematik, Berlin, Germany.

——— (2002b). *Some practical aspects of periodic timetabling*. In CHAMONI, P., LEISTEN, R., MARTIN, A., MINNEMANN, J., STADTLER, H. (editors), *Operations Research Proceedings 2001*. Springer, Berlin.

LINDNER, T. (2000). *Train schedule optimization in public rail transportation*. Ph.D. thesis, Technical University Braunschweig, Braunschweig, Germany.

MARDON, R. (1990). *Optimal cycle and cut bases in graphs*. Ph.D. thesis, Northwestern University, Evanston, Illinois, USA.

NACHTIGALL, K. (1994). *A branch and cut approach for periodic network programming*. Technical Report 29, Hildesheimer Informatik-Berichte, Hildesheim, Germany.

——— (1996a). *Cutting planes for a polyhedron associated with a periodic network*. Technical Report 17, DLR Interner Bericht, Braunschweig, Germany.

——— (1996b). *Periodic network optimization with different arc frequencies*. Discrete Applied Mathematics, 69:1–17.

——— (1999). *Periodic network optimization and fixed interval timetables*. Habilitation thesis, DLR, Braunschweig, Germany.

NACHTIGALL, K., VOGET, S. (1996). *A genetic algorithm approach to periodic railway synchronization*. Computers and Operations Research, 23(5):453–463.

——— (1997). *Minimizing waiting times in integrated fixed interval timetables by upgrading railway tracks*. European Journal of Operational Research, 103(3):610–627.

NEMHAUSER, G., WOLSEY, L. (1988). *Integer and combinatorial optimization*. John Wiley & Sons, New York.

NS (NEDERLANDSE SPOORWEGEN) (2001). *Op het spoor van de klant (On the customer's track)*. In Dutch.

ODIJK, M. (1996). *A constraint generation algorithm for the construction of periodic railway timetables*. Transportation Research, 30(6):455–464.

——— (1997). *Railway timetable generation*. Ph.D. thesis, Delft University of Technology, Delft, The Netherlands.

PRINS, M. (1998). *Het vervoersproces van planning tot uitvoering (The transportation process from planning to operation)*. Internal NS Verkeersleiding document. In Dutch.

ROCKAFELLAR, R. (1984). *Network flows and monotropic optimization*. John Wiley & Sons, New York.

SCHRIJVER, A. (1986). *Theory of linear and integer programming*. John Wiley & Sons, New York.

——— (1998). *Routing and timetabling by topological search*. Documenta Mathematica, Extra volume Proceedings ICM III:687–695.

SCHRIJVER, A., STEENBEEK, A. (1993). *Spoorwegdienstregelingontwikkeling (Timetable construction)*. Technical report, CWI, Amsterdam, The Netherlands. In Dutch.

——— (1994). *Dienstregelingontwikkeling voor Railned (Timetable construction for Railned)*. Technical report, CWI, Amsterdam, The Netherlands. In Dutch.

SERAFINI, P., UKOVICH, W. (1989a). *A mathematical model for periodic event scheduling problems*. SIAM Journal on Discrete Mathematics, 2(4):550–581.

——— (1989b). *A mathematical model for the fixed-time traffic control problem*. European Journal of Operational Research, 42:152–165.

VAN DEN BRABER, M. (2001). *Analyse van circuitbases ten behoeve van het optimaliseren van dienstregelingen (Analysis of cycle bases for timetable optimization)*. Master's thesis, Free University Amsterdam, Amsterdam, The Netherlands. In Dutch.

VOORHOEVE, M. (1993). *Rail scheduling with discrete sets*. Technical report, Eindhoven University of Technology, Eindhoven, The Netherlands.

V&W (MINISTERIE VAN VERKEER EN WATERSTAAT) (2001). *Van A naar beter – nationaal verkeers- en vervoerplan 2001–2020, (National traffic and transportation plan 2010–2020)*. In Dutch.

WEIGAND, W. (1983). *The man-machine dialogue and timetable planning*. Rail International, 3:8–25.

WYCKOFF, D. (1976). *Railroad management*. Lexington, Massachusetts.

ZUIDWIJK, R., KROON, L. (2000). *Integer constraints for train series connections*. Research Report 05-LIS, ERIM, Rotterdam, The Netherlands.

ZWANEVELD, P. (1997). *Railway planning - routing of trains and allocation of passenger lines*. Ph.D. thesis, Rotterdam School of Management, Ph.D-Series in General Management 25, Rotterdam, The Netherlands.

ZWANEVELD, P., KROON, L., ROMEIJN, H., SALOMON, M., DAUZÈRE-PÉRÈS, S., VAN HOESEL, C., AMBERGEN, H. (1996). *Routing trains through railway stations: model formulation and algorithms*. Transportation Science, 30(3):181–194.

Appendix A

Notation

This appendix summarizes the notation that is used in the thesis. It first contains a list of the used abbreviations. Next, it lists all sets, indices for these sets, parameters, and decision variables that are used in the thesis. Finally, the notation for cycles and paths is described, and illustrated with a small example.

A.1 Abbreviations

AVV	Advisory unit on traffic and transport (Adviesdienst Verkeer en Vervoer) of the V&W.
BOT	Basic One–hour Timetable, the timetable for one standard hour.
DONS	Designer Of Network Schedules, the railway timetabling decision support system used at Railned and NSR.
NS	Netherlands Railways (Nederlandse Spoorwegen).
NSR	NS Travelers, the NS passenger railway operator (NS Reizigers).
V&W	The Dutch Ministry of Transportation (Ministerie van Verkeer en Waterstaat).
PESP	Periodic Event Scheduling Problem.
CPF	Cycle Periodicity Formulation.
CRTP	Cyclic Railway Timetabling Problem.
MCB	Minimum Cycle Basis algorithm, by Horton (1987).
MST	Minimum Spanning Tree algorithm for constructing a strictly fundamental cycle basis.

LNT	Long Narrow Tree algorithm for constructing a strictly fundamental cycle basis.
FCB	Fundamental Cycle Basis improvement algorithm for any strictly fundamental cycle basis, by Berger (2002).
MF	The cycle basis constructed by applying FCB to the MST cycle basis.
LNF	The cycle basis constructed by applying FCB to the LNT cycle basis.
BFS	Breadth First Search algorithm for strictly fundamental cycle bases, by Deo (1982).
UE	Unexplored Edges node selection rule for the BFS algorithm.
SDS	Static Degree Sort node selection rule for the BFS algorithm.
DDS	Dynamic Degree Sort selection rule for the BFS algorithm.
MBFS	Multi-point BFS selection rule for the BFS algorithm.

A.2 Sets

$\mathcal{G} = (\mathcal{N}, \mathcal{A} \cup \mathcal{A}^s)$	The railway network graph, consisting of nodes $\mathcal{N}$, regular tracks $\mathcal{A}$, and single tracks $\mathcal{A}^s$.
$\mathcal{N}$	The set of nodes in the railway network.
$\mathcal{A}$	The set of regular tracks $a = (n, m)$ in the railway network, with $n, m \in \mathcal{N}$.
$\mathcal{A}^s$	The set of single tracks $a = (n, m)$ in the railway network, with $n, m \in \mathcal{N}$.
$\mathcal{T}$	The set of trains.
$\mathcal{N}^t \subseteq \mathcal{N}$	The set of nodes that train t visits.
$\mathcal{A}^t \subseteq \mathcal{A} \cup \mathcal{A}^s$	The set of tracks that train t travels along.
$\mathcal{T}_a$	The set of all train pairs that travel along track a in the same direction.
$\mathcal{T}_a^s$	The set of all train pairs that travel along single track a in opposite directions.
$\mathcal{F}_n^d, \mathcal{F}_n^a$	The sets of fixed departure and arrival time trains for network node n.
$\mathcal{S}_n$	The set of train pairs for which the departure times need to be synchronized at network node n.
$\mathcal{C}_n$	The set of train pairs for which a connection or turn around constraint is required at network node n.

P_t	The train path of train t.
$C_{tt'}$	The train cycle for trains t and t'.
K_a	The safety clique for track a.
N_n^a, N_n^d	The arrival and departure node sets of the bipartite dwell-connection graph.
G_n^S	The station graph for network node n.
$G = (N, A)$	The (constraint) graph.
B	The cycle basis of a graph G.
C_i	A cycle in a cycle basis B.
D	A non-basis cycle.

A.3 Indices

n, m	The index for network nodes in $\mathcal{N}$.
$a = (n, m)$	The index for tracks in $\mathcal{A} \cup \mathcal{A}^s$.
i, j, k	The index for nodes in graph G.
a	The index for arcs in graph G.

A.4 Parameters

T	The cycle time of the timetable.
c	The number of cycles in a cycle basis B of G.
h	The general headway time upon departure and arrival for any node.
r_a^t	The trip time of train t for track a.
$[\underline{r}_a^t, \overline{r}_a^t]$	The trip time window for train t along track a.
$[\underline{d}_n^t, \overline{d}_n^t]$	The dwell time window for train t at node n.
$[\underline{f}_n^t, \overline{f}_n^t]$	The fixed departure or arrival time window for train t at node n.
$[\underline{s}_n^{tt'}, \overline{s}_n^{tt'}]$	The time window for the synchronization of trains t and t' at node n.
$[\underline{c}_n^{tt'}, \overline{c}_n^{tt'}]$	The time window for the connection or turn around constraint between trains t and t' at node n.
$[l_{ij}, u_{ij}]$	The time window for arc (i, j).
$[\underline{f}_i, \overline{f}_i]$	The time window for event i.
$[l'_{ij}, u'_{ij}]$	The relaxed time window for arc (i, j) for objective function F_v.

w_a	The objective function weight for arc a.
μ_a	The middle of the time window for arc a.
a_C	The lower bound for cycle C.
b_C	The upper bound for cycle C.
l	The vector of time window lower bounds l_{ij}.
u	The vector of time window upper bounds u_{ij}.
a	The vector of cycle lower bounds a_C.
b	The vector of cycle upper bounds u_C.

A.5 Variables

a_n^t	The arrival time of train t at node n.
d_n^t	The departure time of train t from node n.
v_i	The time instant at which event i takes place.
p_{ij}	The integer variable modeling the cyclic nature of the time window constraint involving events i and j.
F_t	The linear passenger travel time objective function.
$F_{t,q}$	The quadratic passenger travel time objective function.
F_r	The timetable robustness objective function.
F_s	The rolling stock objective function.
F_v	The constraint violation objective function.
σ_a	The used space of the relaxed time window $[l'_a, u'_a]$ for objective function F_v.
δ_a	The auxiliary variable for the robustness objective function F_r.
v	The vector of variables v_i.
p	The vector of variables p_{ij}.
x	The vector of variables x_a.
q	The vector of variables q_C.

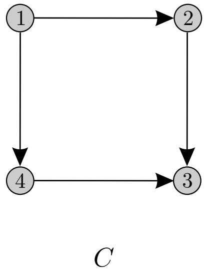


Figure A.1: Example on the notations for cycles

A.6 Paths and Cycles

Throughout the thesis, we only consider simple cycles which need not be directed. A cycle $C = (1, 2, \dots, k, 1)$ visits the nodes $1, 2, \dots, k$ in that order. With respect to this direction, C consists of forward and backward arcs. The sets of forward and backward arcs are denoted by C^+ and C^-, respectively, and are defined as follows:

$$\begin{aligned} C^+ &= \big\{(i,j) \mid i < j, i, j \in C\big\}, \\ C^+ &= \big\{(i,j) \mid i > j, i, j \in C\big\}. \end{aligned}$$

Whenever there is no confusion possible, we also denote by C the set of node pairs as they appear in the direction of the cycle. Formally,

$$C = \big\{(i,j) \mid (i,j) \in C^+\big\} \cup \big\{(j,i) \mid (i,j) \in C^-\big\}.$$

Usually there is no reason for confusion, since we either consider arcs $(i,j) \in C$ or nodes $i \in C$.

As an example, consider the cycle $C = (1, 2, 3, 4, 1)$ with a clockwise direction in Figure A.1. This gives

$$\begin{aligned} C^+ &= \big\{(1,2), (2,3)\big\}, \\ C^- &= \big\{(1,4), (4,3)\big\}, \\ C \;\; &= \big\{(1,2), (2,3), (3,4), (4,1)\big\}. \end{aligned}$$

We use a similar notation for paths. Throughout the thesis, only simple paths are considered. A path from s to t is denoted by P_{st}. The path P_{st} is directed from s to t, and the sets of forward and backward arcs in it by P_{st}^+ and P_{st}^- respectively. We also use P_{st} to denote the set of node pairs as they appear in the path.

Appendix B

Cycle Bases of Graphs

This appendix briefly describes the concepts of cycle spaces and cycle bases for undirected and directed graphs, and illustrates the presented theory with some examples. We also show the relation between cycle bases for undirected and directed graphs. For an in-depth coverage of the subject we refer the interested reader to Deo (1982), Bollobás (1998).

B.1 The Cycle Space of Undirected Graphs

In an undirected graph $U = (N, E)$, a cycle C is encoded by a so-called *cycle vector* φ_C defined as

$$\varphi_{Ce} = \begin{cases} 1 & \text{if } e \in C, \\ 0 & \text{otherwise.} \end{cases}$$

Arithmetic for cycle vectors in undirected graphs is considered over the field GF(2). As an example, consider the undirected graph U in Figure B.1. The two triangle cycles C_1 and C_2 give the cycle vectors

$$\begin{array}{cc} & \begin{array}{ccccc} \scriptstyle 1 & \scriptstyle 2 & \scriptstyle 3 & \scriptstyle 4 & \scriptstyle 5 \end{array} \\ \varphi_1 = & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix}, \\ \varphi_2 = & \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \end{bmatrix}. \end{array}$$

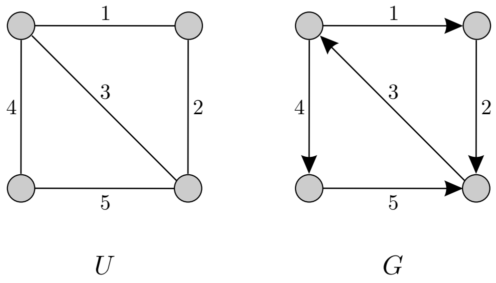


Figure B.1: Examples of encoding cycles in graphs

The cycle vector of the outer square cycle C_4 is given by

$$\varphi_3 = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 \end{bmatrix},$$

and the three cycles are related as

$$\varphi_3 = \varphi_1 + \varphi_2 \text{ modulo } 2.$$

In other words, when two cycles C_1 and C_2 both contain a common edge, then this edge cancels out in their sum.

The *cycle space of an undirected graph* $U = (N, E)$ is the space spanned by the $\{0, 1\}$ cycle vectors φ_C of cycles $C \in U$. A *cycle basis* is a basis of the cycle space of U. A cycle basis B of an undirected graph U can be constructed by the following procedure. First, construct a spanning tree H of U. Adding a non-tree edge e to the edges in H creates a unique cycle C. The non-tree edge e is said to *generate* C. The set of cycles B generated by all non-tree edges is a cycle basis of U. From this procedure it is clear that the cycle space of an undirected graph has dimension $c = |A| - |N| + 1$, since a spanning tree of U contains $|N| - 1$ edges. For a cycle basis $B = \{C_1, \ldots, C_c\}$ with cycle vectors $\varphi_1, \ldots, \varphi_c$, the *cycle matrix* Φ_B is the $c \times |A|$ matrix with the cycle vectors $\varphi_1, \ldots, \varphi_c$ as rows.

B.2 The Cycle Space of Directed Graphs

In a directed graph $G = (N, A)$, a cycle C is not required to be directed, so it may contain forward and backward arcs. Therefore, a cycle in a directed graph is encoded by a $\{0, \pm 1\}$ cycle vector γ_C. Choosing an arbitrary direction for the cycle, γ_C is defined as

$$\gamma_{Ca} = \begin{cases} 1 & \text{if } a \text{ is a forward arc in } C, \\ -1 & \text{if } a \text{ is a backward arc in } C, \\ 0 & \text{if } a \notin C. \end{cases}$$

Contrary to the undirected case, arithmetic is performed over the field $\mathbb{Q}$ for cycles in directed graphs. In the graph G in Figure B.1, the two triangles can be described by the cycle vectors

$$\begin{array}{rc} & \begin{array}{ccccc} \scriptstyle 1 & \scriptstyle 2 & \scriptstyle 3 & \scriptstyle 4 & \scriptstyle 5 \end{array} \\ \gamma_1 = & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix}, \\ \gamma_2 = & \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \end{bmatrix}. \end{array}$$

The outer square now contains both forward and backward arcs, and directing that cycle clockwise, it is encoded as

$$\gamma_3 = \begin{bmatrix} 1 & 1 & 0 & -1 & -1 \end{bmatrix}.$$

These three cycles relate as

$$\gamma_3 = \gamma_1 - \gamma_2.$$

Note that in this case, the sum of cycles C_1 and C_2 makes no sense, since it does not construct a cycle:

$$\gamma_1 + \gamma_2 = \begin{bmatrix} 1 & 1 & 2 & 1 & 1 \end{bmatrix}.$$

The *cycle space of a directed graph* $G = (N, A)$ is the space spanned by the $\{0, \pm 1\}$ cycle vectors γ_C of cycles $C \in G$. A *cycle basis* B of G is a basis of the cycle space of G. As for the undirected case, a cycle basis for a directed graph G can be constructed by first computing a spanning tree H of G. This spanning tree is not required to be directed. The cycles constructed by iteratively adding a non-tree arc to the arcs in H together form a cycle basis of G. Hence, the dimension of the cycle space of a directed graph also equals $c = |A| - |N| + 1$. The *cycle matrix* Γ_B corresponding to a set of cycle basis vectors $\gamma_1, \ldots, \gamma_c$ is the $c \times |A|$ matrix with $\gamma_1, \ldots, \gamma_c$ as rows.

B.3 Relation between Cycle Bases for Undirected and Directed Graphs

Consider a directed graph $G = (N, A)$, and let $U = (N, E)$ be the underlying undirected graph. The *projection* of a cycle $C \in G$ is the corresponding undirected cycle $C' \in U$. So, for the cycle vectors γ and φ of C and C', it holds that $\varphi = |\gamma|$. The theorem below describes the relation between cycle bases of directed and undirected graphs.

Theorem B.1. Consider a directed graph $G = (N, A)$ with underlying undirected graph $U = (N, E)$. Suppose that the set of undirected cycles $C'_1, \ldots, C'_c \in U$ are the projections of a set of cycles $C_1, \ldots, C_c \in G$. If $C'_1, \ldots, C'_c$ form a cycle basis of U, then $C_1, \ldots, C_c$ form a cycle basis of G.

Proof. Let $\varphi_t, \ldots, \varphi_c$ be the cycle vectors of $C'_1, \ldots, C'_c$. Since $C'_1, \ldots, C'_c$ form a cycle basis of U, we have that

$$\sum_{i=1}^{c} \lambda_i \varphi_i = 0 \text{ modulo } 2 \Leftrightarrow \lambda_i = 0 \text{ modulo } 2 \text{ for all } i = 1, \ldots, c. \tag{B.1}$$

Next, let $\gamma_1, \ldots, \gamma_c$ be the cycle vectors of $C_1, \ldots, C_c$. As remarked above, the cycle spaces of U and G both have dimension c. Therefore, if $\gamma_1, \ldots, \gamma_c$ do *not* form a basis of the cycle space of G, there must exist some $\lambda = (\lambda_1, \ldots, \lambda_c) \neq 0$ such that

$$\sum_{i=1}^{c} \lambda_i \gamma_i = 0. \tag{B.2}$$

We assume that λ is integer, and that it contains at least one odd λ_i. Both assumptions are without loss of generality, since one can always construct an integer λ by multiplying by a sufficiently large number, and if all λ_i's are even, one can repeatedly divide them by 2 until at least one λ_i is odd.

Using the fact that γ_i modulo $2 = |\gamma_i| = \varphi_i$, expression (B.2) taken modulo 2 gives

$$\sum_{i=1}^{c} \lambda_i \varphi_i = 0 \text{ modulo } 2. \tag{B.3}$$

Since λ is integer, with at least one odd element, (B.3) contradicts (B.1). It follows that $\gamma_1, \ldots, \gamma_c$ must form a basis of G. ■

Appendix C

Test Instances

C.1 The IC97 instance

The IC97 instance consists of the following intercity trains:

200 International train Amsterdam–Emmerich (on the German border).

500 Rotterdam/The Hague–Groningen/Leeuwarden.

600 International train Amsterdam–Essen (on the Belgian border).

700 Amsterdam/Schiphol–Groningen/Leeuwarden.

800 Haarlem–Maastricht.

900 Haarlem–Eindhoven.

1500 The Hague–Heerlen.

1600 Amsterdam/Schiphol–Enschede.

1700 Rotterdam/The Hague–Enschede.

1900 The Hague–Venlo.

2100 Amsterdam–Vlissingen.

2300 International train Amsterdam–Bad Bentheim (on the German border).

2400 Amsterdam–Dordrecht.

3000 Amsterdam–Arnhem.

9300 International Thalys train Amsterdam–Essen (on the Belgian border).

The international 200, 600, 2300, and 9300 trains travel on from the Dutch border to, respectively, Cologne, Brussels, Berlin, and Paris. However, these parts of the routes are not included in the instance.

All trains are operated with frequency one, except for the 3000 train, which has frequency two. Moreover, synchronization constraints with a bandwidth $\delta = 0$ are defined for trains that travel along the same route. Thereto, the departure times of trains are synchronized at:

Amersfoort Between the 500 and 700 trains, and between the 1600 and 1700 trains.

Amsterdam Between the 600 and 9300 trains, and between the 800 and 900 trains.

Eindhoven Between the 800 and 1500 trains.

The Hague Between the 1500 and 1900 trains, between the 2100 and 2400 trains, and between all four of these trains.

Utrecht Between the 500 and 1700 trains, and between the 800, 900, and 3000 trains.

For all stops, the dwell time windows are set to $[1, 3]$. Moreover, all connection time windows are set to $[2, 5]$. The instance contains connections at:

Amersfoort Between the 500 and 1600 trains, and between the 700 and 1700 trains.

Eindhoven Between the 800 and 1900 trains, and between the 900 and 1500 trains.

Rotterdam Between the 600 and 1700 trains, and between the 500 and 9300 trains.

Utrecht Between the 500 and 3000 trains, and between the 1700 and 3000 trains.

Finally, all trains, synchronizations, and connections are for both directions.

C.2 The ICIR97 instance

The ICIR97 instance consists of all intercity trains, synchronizations, and connections in the IC97 instance. It moreover contains the following interregional trains:

2000 Rotterdam/The Hague–Arnhem.

2200 Amsterdam–Breda.

2600 Amsterdam–The Hague.

3400 Hoorn–The Hague.

3500 Amsterdam–Eindhoven.

3600 Roosendaal–Zwolle.

3700 Eindhoven–Venlo.

3900 Amsterdam–Lelystad.

5600 Utrecht–Zwolle.

5800 Amersfoort–Hoofddorp.

6700 Heerlen–Maastricht.

All trains are operated with frequency two, except for the 3700 and 6700 trains, which have frequency one. Moreover, synchronization constraints with a bandwidth $\delta = 0$ are defined for trains that travel along the same route. Additional to the synchronizations in the IC97 instance, the departures of trains are synchronized at:

Eindhoven Between the 1900 and 3700 trains.

Utrecht Between the 2000 and 3000 trains, and between the 800, 900, and 3500 trains.

For all stops, the dwell time windows are set to $[1, 3]$. Moreover, all connection time windows are set to $[2, 5]$. The instance contains additional connections to the IC97 instance at:

Breda Between the 1500 and 3600 trains, and between the 1900 and 3600 trains.

Deventer Between the 1600 and 3600 trains, and between the 1700 and 3600 trains.

Leiden Between the 2100 and 3400 trains, and between the 2400 and 3400 trains.

Roosendaal Between the 600 and 3600 trains.

Zwolle Between the 500 and 3600 trains, and between the 700 and 3600 trains.

As for the IC97 instance, all additional interregional trains, synchronizations, and connection are for both directions.

C.3 The NH97 instance

The NH97 instance consists of the part of the Netherlands north of the line Leiden-Amsterdam. It therefore contains parts of train lines, and not the train lines on their complete journey. It consists of the following trains:

800 Intercity train Amsterdam–Zandvoort.

3000 Intercity train Amsterdam–Schagen.

3200 Interregional train Amsterdam–Hoorn.

3400 Interregional train Leiden–Haarlem.

3500 Interregional train Haarlem–Hoorn

3900 Local train Amsterdam–Hoorn.

4700 Local train Alkmaar–Amsterdam.

4800 Local train Amsterdam–Uitgeest.

5400 Interregional train Leiden–Amsterdam.

7300 Local train Uitgeest–Amsterdam.

9900 Cargo train Uitgeest–Amsterdam.

All trains are operated with frequency two, with a synchronization bandwidth $\delta = 2$, except for the 9900 cargo train, which has frequency one. The widths of the dwell time windows vary between zero and 4 minutes, depending on the station status. The cargo trains does not stop at all. Moreover, the instance contains connection constraints with time window $[2, 5]$ at Haarlem between the 3400 and 4800 trains, and between the 3400 and 3500 trains.d All trains and connections are again operated in both directions.

C.4 A Visualization of the IC97 Constraint Graph

This section presents a visualization of the constraint graph of the IC97 instance. Since the nodes of the constraint graph correspond to events that take place at railway network nodes, each constraint graph node can been drawn at the geographical coordinates of the corresponding railway network node. Typically, many events are defined for a specific railway network node, and the corresponding constraint graph nodes are thus all drawn at the same location. To increase the visibility of the drawing, we therefore shifted the coordinates of each node a bit. This shift is such, that all nodes belonging to the same train path are shifted by the same amount. Moreover, we always shift a node towards the north east corner of the drawing.

The arcs in the drawing are coded as follows:

Solid For trip-dwell arcs.

Dashed For connection arcs.

Dotted For safety arcs and frequency synchronization arcs.

Note that this means that the train paths consist of solid lines. Further, all safety arcs and connection arcs have been drawn as bezier curves. Figure C.1 shows the resulting drawing of the IC97 constraint graph.

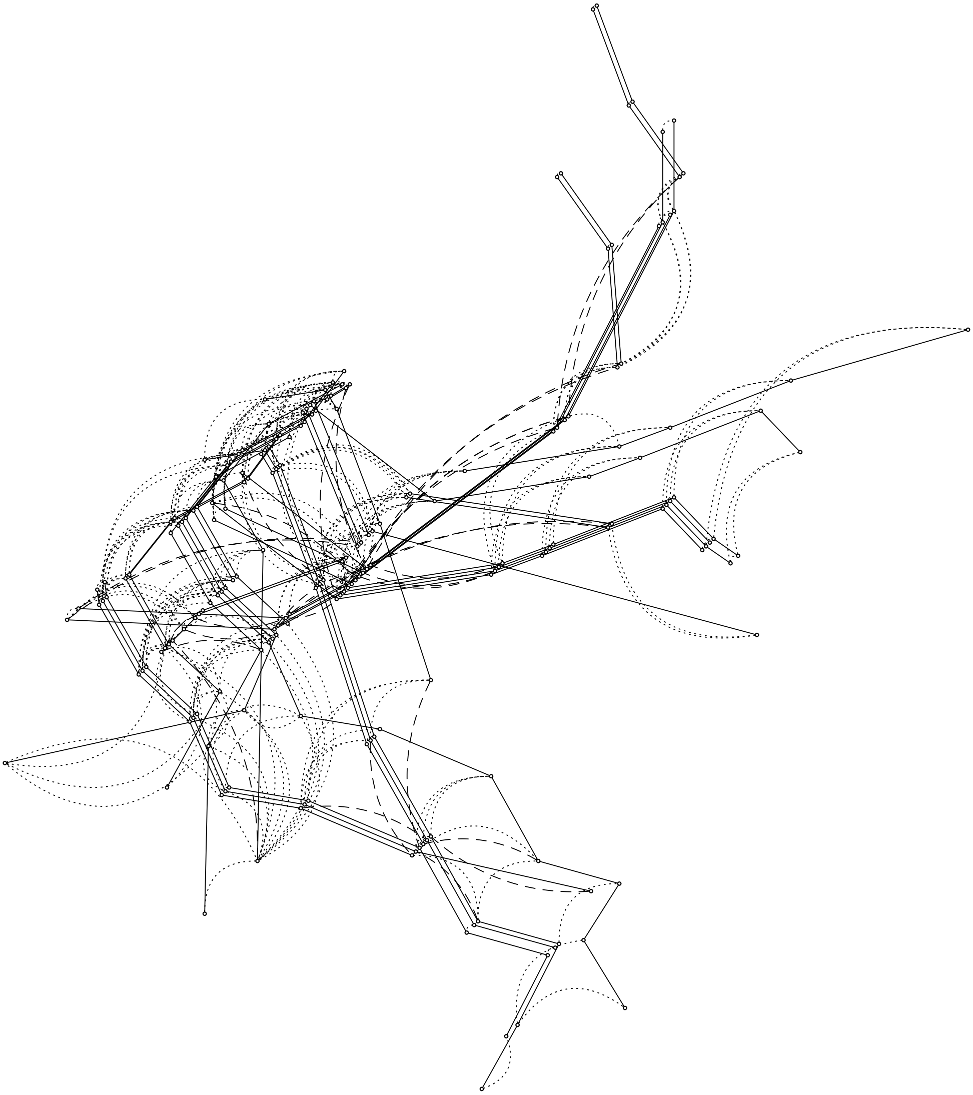


Figure C.1: Constraint graph for the IC97 instance.

Appendix D

Computational Details

This appendix contains the computational results for the NH97 and ICIR97 instances.

	$LW(B)$	avg(W_C)	max(W_C)	$L(B)$	avg(L_C)	max(L_C)
MCB	191	0.38	3	2767	3.8	11
LNF	193	0.39	5	4666	6.3	32
MF	201	0.41	6	4482	6.1	32
LNT	598	1.69	8	11048	15.0	57
MST	595	1.66	8	10520	14.3	54
SDSr	648	1.74	7	3902	9.8	29
DDSr	699	2.21	14	5009	6.8	33
UEr	660	1.88	11	4646	6.3	31
MBFSr	664	1.83	7	4728	6.4	17
SDS	509	1.12	5	2795	3.8	11
DDS	495	1.07	5	2651	3.6	12
UE	471	1.01	5	2531	3.4	13
MBFS	459	0.97	5	2554	3.5	14

Table D.1: Cycle basis algorithms results for the NH97 instance.

	$LW(B)$	avg(W_C)	max(W_C)	$L(B)$	avg(L_C)	max(L_C)
MCB	621	0.75	2	5485	4.6	19
LNF	648	0.84	4	6138	5.1	50
MF	613	0.78	4	5893	4.9	38
LNT	1080	2.08	5	22976	19.1	69
MST	1026	1.93	5	21498	17.9	58
SDSr	1210	2.33	9	15278	12.7	38
DDSr	1096	2.15	11	13780	11.5	39
UEr	982	1.68	7	11183	9.3	44
MBFSr	1114	2.15	10	13097	10.9	32
SDS	1155	1.78	4	7366	6.1	17
DDS	1035	1.55	6	5966	5.0	23
UE	916	1.28	5	4824	4.0	23
MBFS	986	1.51	9	5572	4.6	22

Table D.2: Cycle basis algorithms results for the ICIR97 instance.

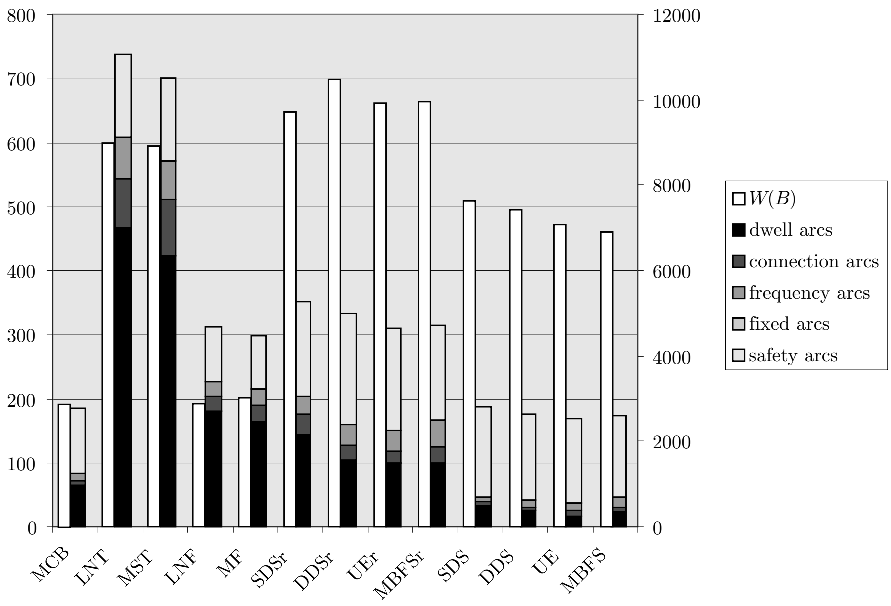


Figure D.1: Cycle bases widths and lengths for the NH97 instance.

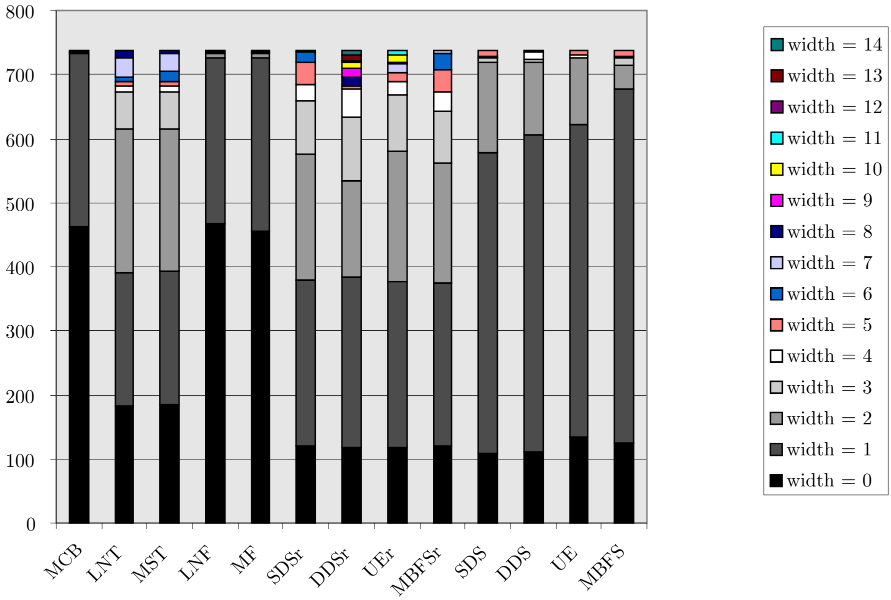


Figure D.2: Widths distribution of the basis cycles for the NH97 instance.

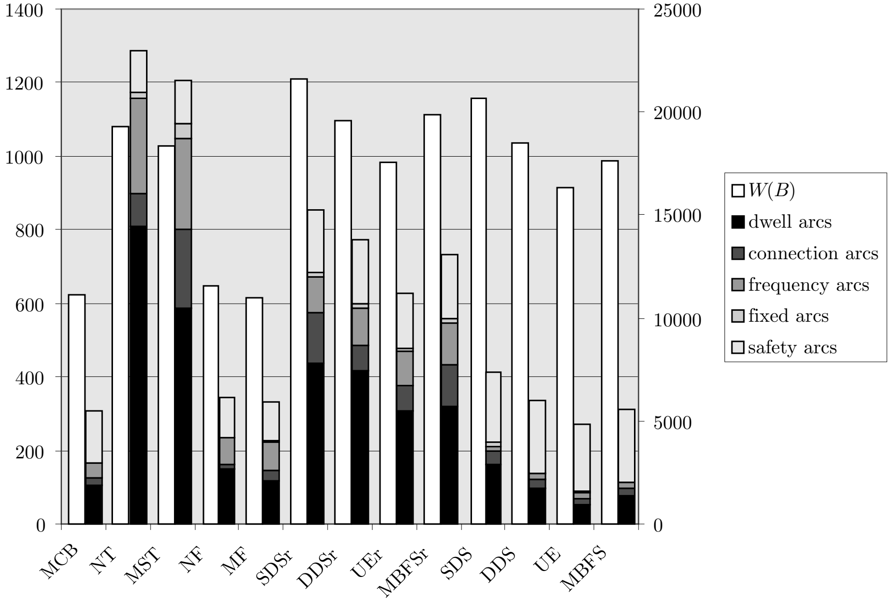


Figure D.3: Cycle bases widths and lengths for the ICIR97 instance.

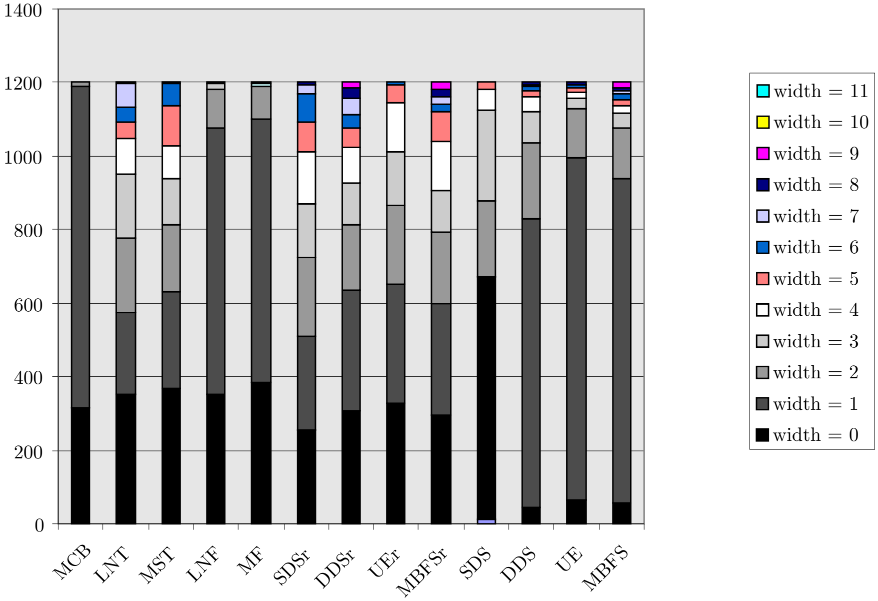


Figure D.4: Widths distribution of the basis cycles for the ICIR97 instance.

	Time (s)	Nodes	Root LP+cuts	Root LP
MCB	2893	*	1418	1418
LNT	38	7169	1418	1418
MST	13	1738	1418	1418
LNF	2868	961396	1418	1418
MF	53	23193	1418	1 418
SDSr	*	801319	1406	1406
DDSr	*	855910	1418	1418
UEr	*	896581	1414	1414
MBFSr	*	799836	1418	1418

Table D.3: Solving the NH97 instance for various cycle bases, with the linear passenger travel time objective F_t (*3600 s. time limit or 1 million nodes limit exceeded).

	Time (s)	Nodes	Root LP+cuts	Root LP
MCB	1006	262921	48	48
LNT	325	48984	48	48
MST	22	3356	48	48
LNF	*	981350	48	48
MF	102	28036	48	48
SDSr	*	752567	27	27
DDSr	*	692661	48	48
UEr	*	700485	40	40
MBFSr	*	698976	48	48

Table D.4: Solving the NH97 instance for various cycle bases, with the quadratic passenger travel time objective $F_{t,q}$ (*3600 s. time limit or 1 million nodes limit exceeded).

	Time (s)	Nodes	Root LP+cuts	Best integer
F_t (root LP= z_{opt} = 1418)				
LNT	12	3041	1418	
LNT+saf	9	937	idem	
LNT+Horton	22	1578	idem	
LNT+saf+Horton	16	1108	idem	
MST	6	1187	idem	
MST+saf	11	1205	idem	
MST+Horton	14	1004	idem	
MST+saf+Horton	135	14016	idem	
$F_{t,q}$ (root LP= z_{opt} = 48)				
LNT	14	1594	48	
LNT+saf	19	1191	idem	
LNT+Horton	19	828	idem	
LNT+saf+Horton	34	1540	idem	
MST	17	2431	idem	
MST+saf	15	885	idem	
MST+Horton	36	1854	idem	
MST+saf+Horton	25	1120	idem	
F_r (root LP = 0, z_{opt} = ?)				
LNT	*	198256	0	125
LNT+saf	*	176743	idem	137
LNT+Horton	*	156855	idem	153
LNT+saf+Horton	*	154873	idem	155
MST	*	268770	idem	149
MST+saf	*	202530	idem	199
MST+Horton	*	165797	idem	288
MST+saf+Horton	*	164992	idem	190

Table D.5: MST and LNT cycle basis with safety triangle cuts and Horton cuts, for the NH97 instance.

Samenvatting

Door de nog altijd toenemende mobiliteit, de privatisering van de Nederlandse Spoorwegen in 1995, en de vertragingen en punctualiteitsproblemen op het spoor, zijn de spoorwegen en hun dienstregelingen de laatste jaren een veelbesproken onderwerp geweest in Nederland. De verwachting is dat het spoor mede een antwoord kan bieden op voorspelde verdere groei van de mobiliteit. Tegen deze achtergrond bestudeert dit proefschrift wiskundige modellen en oplossingsmethoden voor het ontwikkelen van cyclische dienstregelingen van hoge kwaliteit. De Nederlandse dienstregeling, waar elk uur in principe identiek is, is een voorbeeld van zo'n cyclische dienstregeling.

Hoofdstuk 1 van het proefschrift betoogt dat het snel kunnen ontwerpen van dienstregelingen van hoge kwaliteit van groot belang is zowel voor infrastructuurbeheerders, zoals Railned, als voor vervoerders, zoals NS Reizigers. Eén van de taken van Railned is het adviseren van de Nederlandse overheid over investeringen in nieuwe spoorweg infrastructuur. Dergelijke grote infrastructuur investeringen moeten een extra vervoerscapaciteit over een lange tijdshorizon bieden. Railned evalueert daartoe een scala aan infrastructuur-scenario's, door voor elk scenario verschillende dienstregelingen te ontwerpen. Deze dienstregelingen worden vergeleken aan de hand van diverse criteria, zoals vervoersaanbod en robuustheid. Met de invoering van concurrentie op het Nederlandse spoor heeft Railned daarnaast de taak gekregen om uit de dienstregelingsvoorstellen van de diverse vervoerders één gezamenlijke dienstregeling samen te stellen. Deze gezamenlijk dienstregeling moet het maatschappelijke nut maximaliseren, door bijvoorbeeld korte reistijden te bieden, of door het aantal verwachte vertragingen zoveel mogelijk te beperken.

Wat betreft de vervoerder NS Reizigers speelt het snel kunnen ontwerpen van dienstregelingen van hoge kwaliteit een rol bij het uitvoeren van tactische en strategische studies. Tactische studies onderzoeken de haalbaarheid van wijzigingen in de huidige dienstregeling, en strategische studies richten zich op het verkennen van nieuwe dienstregelingsconcepten voor de toekomst. Hoe sneller dergelijke dienstregelingsstudies uitgevoerd kunnen worden, des te sneller en beter kan een vervoerder inspelen op veranderingen in de vervoersmarkt. Daarnaast biedt het grote voordelen als diverse dienstregelingscriteria, zoals kosten of de tevredenheid van passagiers, gekwantificeerd en tegen elkaar afgewogen kunnen worden. Wiskundige modellen en oplossingsmethoden om dienstregelingen te optimaliseren bieden daarom een waardevolle ondersteuning bij de dienstregelingsplanning van infrastructuurbeheerders en vervoerders.

Hoofdstuk 2 van het proefschrift gaat dieper in op de planning van dienstregelingen. Bij het ontwerpen van een dienstregeling moeten de diverse afhankelijkheden met andere planningen, zoals de lijnvoering en de personeelsplanning, in acht worden genomen, en vice versa. Verder beschrijft het hoofdstuk hoe de dienstregeling door Railned gebruikt wordt als een instrument om de capaciteit van toekomstige infrastructuur uitbreidingen te evalueren.

Nadat de sociale en organisatorische achtergrond van het dienstregelingsontwerp uitvoerig behandeld zijn, beschrijft hoofdstuk 3 een geheeltallig programmeringsmodel voor het optimaliseren van dienstregelingen, genaamd het Cyclic Railway Timetabling Problem (CRTP). Het CRTP modelleert een dienstregeling aan de hand van aankomsttijden en vertrektijden van treinen op de diverse knooppunten in het spoorwegnet. Door uitvoerige voorbeeld-restricties wordt duidelijk gemaakt hoe de diverse voorwaarden waaraan een dienstregeling moet voldoen, zoals aansluitingen en veiligheidseisen, kunnen worden gemodelleerd. Het hoofdstuk beschrijft verder lineaire doelstellingsfuncties die gericht zijn op het minimaliseren van de reistijden, het maximaliseren van de robuustheid van de dienstregeling, en het minimaliseren van de mate waarin de opgelegde eisen moeten worden geschonden, indien deze onhaalbaar blijken te zijn. Ook wordt het benaderen van een kwadratische variant van deze doelstellingsfuncties behandeld. Het CRTP heeft een compacte graaf representatie, en de speciale structuur van deze graaf wordt ook uitvoerig besproken.

Hoofdstukken 4 en 5 gaan vervolgens in op de wiskundige aspecten van het CRTP. Hoofdstuk 4 beschrijft het Periodic Event Scheduling Problem (PESP), een bestaand model uit de literatuur dat het CRTP veralgemeniseerd. De wiskundige formulering van het PESP is gebaseerd op de tijdstippen waarop gebeurtenissen plaatsvinden, en op periodieke restricties die het tijdsverschil tussen paren van gebeurtenissen tot een gegeven tijdvenster beperken. De periodiciteit van de restricties wordt gemodelleerd door geheeltallige variabelen. Ook het PESP heeft een graaf representatie, en aan de hand van deze representatie wordt het verband tussen het PESP en het klassieke kortste pad probleem verduidelijkt. Verder beschrijft hoofdstuk 4 enkele eigenschappen van het PESP, waaruit een nuttig resultaat voor het cyclisch ordenen van gebeurtenissen wordt afgeleid. Tenslotte wordt een kort overzicht van aan het PESP gerelateerde literatuur gepresenteerd.

Hoofdstuk 5 generaliseert het bekende concept van tensions in een graaf voor de periodieke situatie. Gebaseerd op deze periodieke tensions wordt het PESP getransformeerd naar de zogenaamde Cycle Periodicity Formulation (CPF). De CPF maakt gebruik van procestijden, en vereist dat de procestijden binnen elke cycle in de graaf periodiek zijn. Deze periodiciteit binnen een cycle wordt gemodelleerd door een geheeltallige variabele. In principe moet de periodiciteit binnen elke cycle in de graaf afgedwongen worden, maar het proefschrift toont aan dat het volstaat om de cycles te beschouwen in een geheeltallige basis van de cycle ruimte van de graaf. Verder toont een voorbeeld aan dat een niet-geheeltallige cycle basis er toe kan leiden dat het model een foutieve dienstregeling levert. Aan de hand van het cyclisch ordeningsresultaat uit hoofdstuk 4 wordt daarna de relatie aangetoond tussen bepaalde cycles in de graaf, en de volgorde waarin treinen over het spoor rijden. Deze trein-volgordes leiden vervolgens tot een klasse van sneden voor de CPF, die tot de grotere klasse van reeds bekende cycle sneden blijken te behoren.

Het resterende deel van hoofdstuk 5 is gewijd aan verschillende aspecten van cycle bases. Zo wordt aangetoond dat de bekende klassen van strikt fundamentele en algemeen fundamentele cycle basen geheeltallig zijn, hetgeen betekent dat ze gebruikt kunnen worden om de CPF te formuleren. Vervolgens betoogt het proefschrift dat het voordelig is om de CPF met een smalle cycle basis te formuleren, en dat algoritmen voor het zogenaamde minimum cycle basis probleem een benadering van zulke smalle cycle bases berekenen. Uitgaande van de speciale structuur van de graaf representatie van het CRTP, worden verder twee gewichtsfuncties voor het bepalen van opspannende bomen voorgesteld, die strikt fundamentele cycle bases met een smalle breedte leveren. Tenslotte beschrijft het hoofdstuk hoe de breedte van een strikt fundamentele cycle basis verder verkleind kan worden door de bijbehorende algemeen fundamentele cycle basis te berekenen.

Hoofdstuk 6 is gewijd aan diverse uitbreidingen van het CRTP. Als eerste wordt uitgelegd hoe variabele rijtijden van treinen in het model kunnen worden opgenomen. Vervolgens wordt de modellering van flexibele aansluitingen beschreven, hetgeen inhoudt dat twee verzamelingen van treinen één of meerdere aansluitingen op elkaar moeten bieden. De derde uitbreiding toont hoe de capaciteiten van stations opgenomen kunnen worden in het CRTP, en de vierde uitbreiding beschrijft een alternatieve doelstellingsfunctie die gericht is op het minimaliseren van het aantal trein-composities dat nodig is om de dienstregeling uit te voeren. Tenslotte wordt in hoofdstuk 6 een cycle fixatie heuristiek beschreven. Deze heuristiek is geïnspireerd door de werkwijze van praktische planners, die vaak eerst de belangrijkste treinen plannen, de dienstregeling voor deze treinen vastprikken. Vervolgens worden iteratief andere klassen van treinen aan de dienstregeling toegevoegd.

In hoofdstuk 7 worden de ideeën uit de voorgaande hoofdstukken getest op een drietal testproblemen. Deze tests tonen aan dat het van essentieel belang is om een preprocessing stap uit te voeren teneinde de grootte van de instanties te reduceren. Ook blijkt het te lonen om in een tweede preprocessing stap de breedte van de tijdvensters in de periodieke restricties te verkleinen. Vervolgens wordt duidelijk dat de PESP variant van het CRTP model vrij lange rekentijden vergt, en dat de CPF duidelijk beter presteert bij het optimaliseren van cyclische spoorwegdienstregelingen. Om deze reden is het overige deel van hoofdstuk 7 gewijd aan verdere tests van de CPF.

Een tweede factor die van belang is om de CPF snel op te kunnen lossen, blijkt het soort cycle basis te zijn waarmee het model geformuleerd wordt. De tests tonen aan dat de voorgestelde gewichtsfuncties voor opspannende bomen over het algemeen de beste resultaten opleveren voor de reistijd-doelstellingsfunctie. Het blijkt veel meer tijd te kosten om een robuuste dienstregeling te berekenen, terwijl een dienstregeling met een minimaal benodigd aantal trein-composities vrij snel berekend kan worden. Voor instanties met een onhaalbaar eisenpakket blijkt de voorgestelde doelstellingsfunctie een effectief middel te zijn om een indicatie te verkrijgen van de benodigde aanpassingen. Het toevoegen van sneden aan de model formulering reduceert de rekentijd in de meeste gevallen. Toch voldoen deze technieken niet om binnen een redelijke tijd goede dienstregelingen te berekenen voor het grootste testprobleem. Echter, met de cycle fixatie heuristiek kunnen ook voor dit grootste probleem binnen een redelijke rekentijd goede dienstregelingen worden berekend. Tenslotte concludeert het hoofdstuk dat de kwaliteit van de berekende dienstregelingen zodanig is, dat de extra rekentijd die een optimalisatie vergt gerechtvaardigd is.

Hoofdstuk 8 vat tenslotte de belangrijkste resultaten van het proefschrift samen, en beantwoordt de onderzoeksvragen die in hoofdstuk 1 zijn geformuleerd. Daarnaast worden kort de beperkingen van het proefschrift beschreven, en worden enkele suggesties voor vervolgonderzoek gedaan.

Curriculum Vitae

Leon Peeters studied Econometrics at the University of Maastricht, with a major in Operations Research and Computer Science. After a six month internship at KPN Research in 1996, he graduated in 1997 with an M.Sc. thesis on the optimization of the backbone of a GSM network.

In 1998, he started as a Ph.D. student at the Rotterdam School of Management of the Erasmus University Rotterdam. His Ph.D. research considered the optimization of cyclic railway timetables. Part of this research was carried out in cooperation with the innovation department of Railned, the Dutch railway infrastructure capacity manager, and with the logistics department of NSR, the largest Dutch passenger railway operator.

From January 2002 until May 2003, he visited the Algorithms & Data Structures group at the University of Konstanz through the European Commission's AMORE network.

He has presented his work at various international conferences and workshops, and in 1999 he received the TRAIL IT&L best paper award. Several of his papers have been published in journals and conference proceedings.

Erasmus Research Institute of Management

ERIM Ph.D. Series
Research in Management

Title:	**Operational Control of Internal Transport**
Author:	J. Robert van der Meer
Promotor(es):	Prof.dr.ir. M.B.M. de Koster, Prof.dr.ir. R. Dekker
Defended:	September 28, 2000
Series number:	1
Published:	ERIM Ph.D. Series Research in Management
ISBN:	90–5892–004–6

Title:	**Quantitative Models for Reverse Logistics**
Author:	Moritz Fleischmann
Promotor(es):	Prof.dr.ir. J.A.E.E. van Nunen, dr. R. Kuik
Defended:	October 5, 2000
Series number:	2
Published:	Lecture Notes in Economics and Mathematical Systems, Volume 501, 2001, Springer Verlag, Berlin
ISBN:	3540–417–117

Title:	**Optimization Problems in Supply Chain Management**
Author:	Dolores Romero Morales
Promotor(es):	Prof.dr.ir. J.A.E.E. van Nunen, dr. H.E. Romeijn
Defended:	October 12, 2000
Series number:	3
Published:	ERIM Ph.D. Series Research in Management
ISBN:	90–9014078–6

Title:	**Layout and Routing Methods for Warehouses**
Author:	Kees Jan Roodbergen
Promotor(es):	Prof.dr.ir. M.B.M. de Koster, Prof.dr.ir. J.A.E.E. van Nunen
Defended:	May 10, 2001
Series number:	4
Published:	ERIM Ph.D. Series Research in Management
ISBN:	90–5892–005–4

Title:	**Rethinking Risk in International Financial Markets**
Author:	Rachel Campbell
Promotor(es):	Prof.dr. C.G. Koedijk
Defended:	September 7, 2001
Series number:	5
Published:	ERIM Ph.D. Series Research in Management
ISBN:	90–5892–008–9

Title:	**Labour flexibility in China's companies: an empirical study**
Author:	Yongping Chen
Promotor(es):	Prof.dr. A. Buitendam, Prof.dr. B. Krug
Defended:	October 4, 2001
Series number:	6
Published:	ERIM Ph.D. Series Research in Management
ISBN:	90–5892–012–7

Title:	**Strategic Issues Management: Implications for Corporate Performance**
Author:	Pursey P. M. A. R. Heugens
Promotor(es):	Prof.dr.ing. F.A.J. van den Bosch, Prof.dr. C.B.M. van Riel
Defended:	October 19, 2001
Series number:	7
Published:	ERIM Ph.D. Series Research in Management
ISBN:	90–5892–009–7

Title:	**Beyond Generics; A closer look at Hybrid and Hierarchical Governance**
Author:	Roland F. Speklé
Promotor(es):	Prof.dr. M.A. van Hoepen RA
Defended:	October 25, 2001
Series number:	8
Published:	ERIM Ph.D. Series Research in Management
ISBN:	90–5892–011–9

Title: **Interorganizational Trust in Business to Business E-Commerce**
Author: Pauline Puvanasvari Ratnasingam
Promotor(es): Prof.dr. K. Kumar, Prof.dr. H.G. van Dissel
Defended: November 22, 2001
Series number: 9
Published: ERIM Ph.D. Series Research in Management
ISBN: 90–5892–017–8

Title: **Outsourcing, Supplier-relations and Internationalisation: Global Source Strategy as a Chinese puzzle**
Author: Michael M. Mol
Promotor(es): Prof.dr. R.J.M. van Tulder,
Defended: December 13, 2001
Series number: 10
Published: ERIM Ph.D. Series Research in Management
ISBN: 90–5892–014–3

Title: **The Business of Modularity and the Modularity of Business**
Author: Matthijs J.J. Wolters
Promotor(es): Prof.mr.dr. P.H.M. Vervest, Prof.dr.ir. H.W.G.M. van Heck
Defended: February 8, 2002
Series number: 11
Published: ERIM Ph.D. Series Research in Management
ISBN: 90–5892–020–8

Title: **The Quest for Legitimacy; On Authority and Responsibility in Governance**
Author: J. van Oosterhout
Promotor(es): Prof.dr. T. van Willigenburg, Prof.mr. H.R. van Gunsteren
Defended: May 2, 2002
Series number: 12
Published: ERIM Ph.D. Series Research in Management
ISBN: 90–5892–022–4

Title: **Information Architecture and Electronic Market Performance**
Author: Otto R. Koppius
Promotor(es): Prof.dr. P.H.M. Vervest, Prof.dr.ir. H.W.G.M. van Heck
Defended: May 16, 2002
Series number: 13
Published: ERIM Ph.D. Series Research in Management
ISBN: 90-5892-023 - 2

Title: **Planning and Control Concepts for Material Handling Systems**
Author: Iris F.A. Vis
Promotor(es): Prof.dr. M.B.M. de Koster, Prof.dr.ir. R. Dekker
Defended: May 17, 2002
Series number: 14
Published: ERIM Ph.D. Series Research in Management
ISBN: 90–5892–021–6

Title: **Essays on Agricultural Co-operatives; Governance Structure in Fruit and Vegetable Chains**
Author: Jos Bijman
Promotor(es): Prof.dr. G.W.J. Hendrikse
Defended: June 13, 2002
Series number: 15
Published: ERIM Ph.D. Series Research in Management
ISBN: 90–5892–024–0

Title: **Analysis of Sales Promotion Effects on Household Purchase Behavior**
Author: Linda H. Teunter
Promotor(es): Prof.dr.ir. B. Wierenga, Prof.dr. T. Kloek
Defended: September 19, 2002
Series number: 16
Published: ERIM Ph.D. Series Research in Management
ISBN: 90–5892–029–1

Title: **Incongruity between Ads and Consumer Expectations of Advertising**
Author: Joost Loef
Promotor(es): Prof.dr. W.F. van Raaij, Prof.dr. G. Antonides
Defended: September 26, 2002
Series number: 17
Published: ERIM Ph.D. Series Research in Management
ISBN: 90–5892–028–3

Title: **Creating Trust between Local and Global Systems**
Author: Andrea Ganzaroli
Promotor(es): Prof.dr. K. Kumar, Prof.dr. R.M. Lee
Defended: October 10, 2002
Series number: 18
Published: ERIM Ph.D. Series Research in Management
ISBN: 90–5892–031–3

Title: **Coordination and Control of Globally Distributed Software Projects**
Author: Paul C. van Fenema
Promotor(es): Prof.dr. K. Kumar
Defended: October 10, 2002
Series number: 19
Published: ERIM Ph.D. Series Research in Management
ISBN: 90–5892–030–5

Cyclic Railway Timetable Optimization

Cyclic Railway Timetable Optimization describes mathematical models and solution methods for constructing high quality cyclic railway timetables. In a cyclic timetable, a train for a certain destination leaves a certain station at the same time every cycle time, say every half an hour, every hour, or every two hours. Cyclic timetables are widely used in European railways. They offer a clear and transparant product to the railway customers, who only need to memorize the minutes of the hour at which their regular trains depart. Because of the important role of timetable planning for railway operators and railway infrastructure managers, models and methods for optimizing cyclic railway timetables provide a valuable tool for these organizations. The thesis presents a mathematical model for optimizing cyclic railway timetables, and studies the theoretical aspects behind the model. The investigated aspects include cyclic sequencing, periodic tensions, cycles in graphs, cycle bases of graphs, algorithms for constructing cycle bases, and cutting planes for the model. The developed theoretical ideas are tested on some real-life cyclic railway timetabling instances. The thesis further develops several extensions of the basic model.

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 and the Rotterdam School of Economics. 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 focussed on the management of the firm in its environment, its intra- and inter-firm 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 graduate program in Research in Management. Within ERIM, over two hundred senior researchers and Ph.D. candidates are active in the different research programs. 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.

The ERIM PhD Series contains Dissertations in the field of Research in Management defended at Erasmus University Rotterdam. The Dissertations in the Series are available in two ways, printed and electronical. ERIM Electronic Series Portal: http://hdl.handle.net/1765/1.

ERIM

ERASMUS RESEARCH
INSTITUTE OF MANAGEMENT

www.research-in-management.nl/

ISBN 90-5892-042-9