Integrated capacity assessment and timetabling models for dense railway networks

Bešinović, Nikola

DOI
10.4233/uuid:9083a9cc-64a1-4676-9134-9f8652d629e0

Publication date
2017

Document Version
Accepted author manuscript

Citation (APA)
Bešinović, N. (2017). *Integrated capacity assessment and timetabling models for dense railway networks*. [Dissertation (TU Delft), Delft University of Technology]. TRAIL Research School. https://doi.org/10.4233/uuid:9083a9cc-64a1-4676-9134-9f8652d629e0

Integrated capacity assessment and timetabling models for dense railway networks

Nikola Bešinović

Delft University of Technology, 2017

Integrated capacity assessment and timetabling models for dense railway networks

Proefschrift

ter verkrijging van de graad van doctor

aan de Technische Universiteit Delft,

op gezag van de Rector Magnificus prof. ir. K.Ch.A.M. Luyben,

voorzitter van het College voor Promoties,

in het openbaar te verdedigen op dinsdag 4 juli 2017 om 10:00 uur

door

Nikola BEŠINOVIĆ

Master of Science in Operations Research in Transport

University of Belgrade, Servië

geboren te Zaječar, Servië

This dissertation has been approved by the
promotor: Prof. dr. ir. S.P. Hoogendoorn
copromotor: Dr. R.M.P. Goverde

Composition of the doctoral committee:

Rector Magnificus	Chairperson
Prof. dr. ir. S.P. Hoogendoorn	Promotor
Dr. R.M.P. Goverde	Copromotor

Independent members:

Prof. dr. L. Nie	Beijing Jiaotong University
Prof. dr. A. Schöbel	Georg-August-Universität Göttingen
Prof. dr. D. Huisman	Erasmus University Rotterdam
Prof. dr. C. Witteveen	Faculty of Engineering, Mathematics and Computer Science, TU Delft
Prof. dr. ir. R.P.B.J. Dollevoet	Faculty of Civil Engineering and Geosciences, TU Delft

This thesis is the result of a PhD research carried out from 2012 to 2016 at Delft University of Technology, Faculty of Civil Engineering and Geosciences, Department of Transport and Planning. This research was supported by the European FP7 project Optimal Networks for Train Integration Management across Europe (ON-TIME).

TRAIL Thesis Series no. T2017/9, the Netherlands TRAIL Research School

TRAIL
P.O. Box 5017
2600 GA Delft
The Netherlands
E-mail: info@rsTRAIL.nl

ISBN 978-90-5584-226-1

Printed in the Netherlands

Repetitio est mater studiorum.
Latin proverb

Preface

Improving railway services, we surely need that, many would say. And we could also agree, mathematical models are certainly worth-exploiting to do so. But, don't we already have a blast of available models and approaches to perform these tasks correctly and we only need to translate them into real applications? Do we really need another PhD thesis focusing on railway timetabling in the sea of already existing research? Nevertheless, only limited implementations of designing timetables do exist in practice, so something must be missing still. It was not always easy, particularly in the beginning of our quest, to discover these missing links and define new concepts to advance the state-of-the-art. Thanks to much previous research, coming from ETH Zürich and TU Berlin among others, we had to look further and explore the railway system in greater details in order to understand what is still needed before applying such mathematical models to timetabling becomes a common and widely accepted practice.

This thesis incorporates optimization, simulation and data analysis to create better, more effective and more reliable railway transport system. It advances the current practice in infrastructure capacity assessment and timetabling by integrating different mathematical models for designing high-quality railway timetables. This PhD research was part of the European FP7 project Optimal Networks for Train Integration Management across Europe (ON-TIME) that gathered infrastructure managers, IT companies and research groups to develop advanced models and algorithms for improving railway planning and operations that optimizes the use of existing infrastructure capacity and reduces overall delays in networks. I was engaged in manufacturing and writing this thesis between 2012 and 2016.

This book aims to be a teaching material and a sound support to students (e.g., in the fields of transport, logistics, operations research, computer science and econometrics) and practitioners in the field of railway traffic planning and management. Each chapter is considered to be standalone and can be read independently. The book is not only a set of new and improved optimization models for timetabling, but it represents a comprehensive package of advanced mathematical models and algorithms that serve to better and more effective overall timetable evaluation and design. Furthermore, it incorporates in-depth technical knowledge of railway systems and makes benefit of it in creating more realistic modelling representation of the system. It combines and balances operations research techniques and railway engineering knowledge. I

strongly believe, and this book shows, that advanced mathematical models coupled with a field expert knowledge represent the future of planning systems and greatly contribute to creating more attractive and sustainable railway services.

First and foremost thank goes to Rob Goverde for his wonderful support; his rigorous eye for even tiniest details kept the research at the high-end note and made me always strive for perfectionism. I also thank my great friend now Egidio, and four years back, a big brother in research and out of it. Rob, Egidio, thank you for your encouragement, patience and given freedom during my first steps as a PhD researcher. Thank to Ingo Hansen for critical and detailed feedback on our research and for the opportunity to co-lecture the course in Beijing in 2016. Thank to Serge for being a good promotor and for being always fit to discuss our progress.

I am indebted to co-authors and complete ON-TIME team for creating a great multi-disciplinary dynamic and challenging environment, for great discussions and valuable contribution to this book. I would also like to express my gratitude to all committee members, taking time to read this thesis and to provide useful comments. Thanks to ProRail and Netherlands Railways, we tested our approaches on real-life instances from which this thesis contributed greatly. Also, thank to Pavle for processing a mountain of TROTS data and allowing us to use clean and easy input for our work. Thank to Sander for being a great collaborator on developing the graphical interfaces for the timetable planning toolbox.

Utmost, special love and admiration to you Natalija and my family, for the everlasting and unconditional support during these long and colourful times. Without you, this journey would not have been as joyful and inspiring.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and motivation

Mainline railways in Europe are experiencing more and more intensive use of their train services, particularly in urban areas, as the worldwide demand for passenger and freight transport is increasing across all transport modes. At the same time, much of the existing mainline railway network has become susceptible to delays and disturbances. For example, Figure 1.1 compares punctuality and track occupation in twenty-four countries around the world (NS, 2015). It shows that the Netherlands, together with Switzerland and Japan, is one of the busiest networks internationally. Note that, in terms of performance, several countries have somewhat higher punctuality.

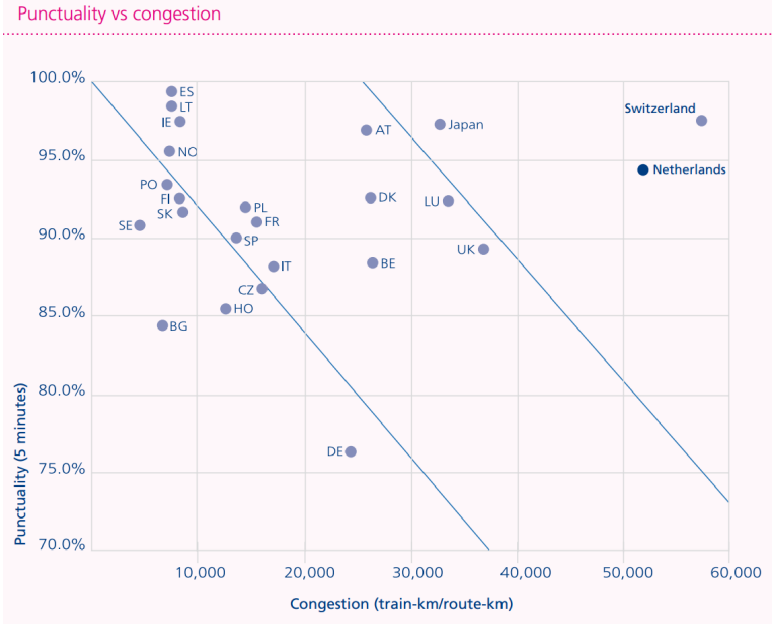

Figure 1.1: Punctuality vs congestion (NS, 2015)

In order to accommodate future demand, more train services need to be scheduled while maintaining or even improving the performance. In the Netherlands, the ongo-

ing project Better and More (*Beter en Meer* in Dutch) focuses first on better operating services, i.e., increased punctuality and customer' satisfaction, and then on more scheduled trains in the railway network (Ministerie van Infrastructuur en Milieu, 2013). In addition, a similar project exists that focuses on increasing number of train services – High-Frequency Rail Transport Programme (abbreviated PHS in Dutch). From the project report (Ministerie van Verkeer en Waterstaat, 2010), one of the principal goals (originally until 2020) was stated:

> "There will be [on average] 6 intercity trains and 6 Sprinters (all-station regional trains) every hour on the busiest rail routes in the country and there will be additional rail capacity for freight transport. This is the crux of the decision made by the Dutch Government on 4 June 2010 regarding the development of the High-Frequency Rail Transport Programme (PHS)."

This suggests that more train services will operate on a number of routes. However, current infrastructure capacity use with today's planning approaches is reaching its limits. After gradual increases on certain corridors in past years, some trains in 2017 timetable could not be scheduled due to insufficient infrastructure capacity. For example, from Utrecht to Amersfoort, six intercity trains per hour were requested, but only four were scheduled due to limiting platform capacity in station Amersfoort.

Meanwhile, designing railway timetables is still a largely manual process, which is extremely time-consuming and incorporates a substantial amount of constraints, particularly for busy networks such as the Netherlands and Switzerland (ProRail, 2016; SBB, 2016). In addition, such manual processes do not always include all design performance indicators, such as timetable feasibility (i.e., all trains operate undisturbed by other traffic), stability (do not have excessive infrastructure capacity occupation) or robustness (i.e., ability to mitigate certain everyday operational disturbances) (e.g., NS (2015); ProRail (2016)).

In the current planning process, planners often do not know if trains will be able to run without conflicts, so it is necessary to additionally evaluate timetables. Solutions are only tested afterwards and if any issues are observed, then those have to be updated and resolved by planners again. However, sometimes, detailed testing is performed only partially and only for certain performance indicators. For example, a microscopic simulation software, such as OpenTrack and RailSys, may be used to test a (part of) the network on timetable feasibility (Planting, 2016), while stability is hardly ever checked. Translating such partially evaluated timetables to everyday services means that trains may run late and have unexpected stops on the open tracks, somewhere in the fields, or just before the stations (De Goffau, 2013). This consequently affects the on-time performance causing delayed trains and reduces passenger satisfaction.

On one hand, a solution to the problem of saturated railway networks would be to build more railway capacity sufficient to run all trains on dedicated infrastructure; however,

constructing new railways is expensive, takes considerable time and faces a number of environmental constraints. On the other hand, mathematical models and algorithms for capacity estimation and timetabling could be used to produce better timetable solutions and to speed up the planning process. The latter is particularly useful as planners would have more time available for evaluating different timetable variants, schedule more trains and have more satisfied customers overall. In order to achieve goals of Better and More and PHS, more sophisticated automatic tools and algorithms are surely needed.

One of the successful implementations of mathematical models for timetabling is a tool called *Designer of Network Schedules* (DONS) (Kroon et al., 2009), developed for the main railway undertaking Netherlands Railways (NS) and currently used by both NS and the Dutch infrastructure manager ProRail. DONS consists of two models: CADANS and STATIONS. CADANS is a macroscopic design tool that focuses on normative feasibility, which is finding a timetable that satisfies so called macroscopic constraints. The macroscopic level considers stations as simple nodes and tracks in-between as arcs. STATIONS is a more detailed routing tool that finds a good routing plan for complex stations. However, DONS does not include efficiency (i.e., short travel times), stability, feasibility or robustness. To improve efficiency of solutions, DONS is supported by a post-optimization model. In 2008, only one year after new planning tools had been implemented, passenger numbers increased with 2.8% and annual profit with 10 million Euro, while the train punctuality improved from 84.8 to 87.0%. In recent years, DONS has barely been used for designing new timetables, as the timetabling instances became too complex and too difficult to solve by the existing algorithm. A similar timetabling application exists in Germany, where the tool TAKT has been developed (Opitz, 2009). TAKT also finds a timetable that satisfies macroscopic constraints in a first step, and is supported with a more sophisticated post-optimization to improve the constructed timetable according to a chosen objective function (Nachtigall & Opitz, 2008).

In the literature, various other mathematical models have been proposed for railway timetabling (Cacchiani & Toth, 2012). These models commonly use different objective variants of efficiency and robustness. However, most of the current models assume a macroscopic representation of infrastructure and do not include microscopic details. A microscopic level also considers detailed track infrastructure, signalling system, and train characteristics. This means that generated solutions are not always feasible, i.e., conflict-free, and thus would directly induce certain delays when operated in practice. Therefore, macroscopic timetabling models should be extended or integrated with more detailed models to ensure operational feasibility of the timetable. To this end, a few approaches have been proposed in the literature based on a hierarchical integration of timetabling models with different levels of detail. Schlechte, Borndörfer, Erol, Graffagnino, and Swarat (2011) presented a bottom-up approach which first aggregates microscopic running and headway times to be used by a macroscopic model that subsequently identifies an optimised timetable for a given objective function. Feasibility is checked by simulating the timetable at a microscopic level. Caimi (2009)

proposed a two-level framework for designing conflict-free timetables which presents a top-down approach.

The main shortcoming of existing integrated timetabling approaches is that they do not consider any iterative modification to the timetable when it has proven infeasible at the microscopic level (Caimi, 2009; Schlechte et al., 2011), which can occur quite often in dense railway networks. In other words, these approaches are one-directional and cannot guarantee timetable feasibility. In addition, they do not consider timetable stability and robustness. What is more, existing approaches for capacity estimation are limited to corridors, while they tend to be not applicable to stations (and complete networks) (Lindner, 2011).

1.2 Research questions

Considering the existing need for more sophisticated and in-depth approaches for creating more reliable and high quality railway timetables and more accurate capacity estimation, we formulate the main research question of this thesis as:

How to design efficient, feasible, stable and robust railway timetables that provide a high level of service to passengers and freight operators?

Timetable efficiency reflects the amount of time allowances in the scheduled travel times (running, dwell and transfer times) which must be as short as possible to provide short journey times and seamless connections. *Timetable feasibility* is the ability of all trains to adhere to their scheduled train paths[1]. A timetable is feasible if (i) the individual processes are realisable within their scheduled process times, and (ii) the scheduled train paths are conflict-free, i.e., all trains can proceed undisturbed by other traffic. *Timetable stability* is the ability of a timetable to absorb delays so that delayed trains return to their scheduled train paths without rescheduling. This is directly connected with the infrastructure occupation rate. The higher this rate, the lower are the time allowances and hence the less stable is the timetable. *Timetable robustness* is the ability of a timetable to withstand design errors, parameter variations, and changing operational conditions.

In order to answer the posed research question, we identify several open challenges that have to be tackled in advance to make generating feasible, stable and robust timetables possible.

1. How to evaluate infrastructure capacity occupation accurately? How to use capacity occupation as a stability measure? (Chapter 2)

2. Which performance measures and models have to be considered for high quality timetable planning? (Chapter 3)

[1] A train path is the infrastructure capacity needed to run a train between two places over a given time period (EC, 2001).

3. What is the added value of using microscopic models for timetable planning? (Chapter 4)

4. How to include efficiency, stability and robustness in macroscopic timetabling models and guarantee a good trade-off between timetable efficiency and robustness? (Chapter 5)

5. How to integrate microscopic and macroscopic models for efficient, feasible, stable and robust timetabling? (Chapter 6)

6. How to provide reliable running times[2] for timetable design using traffic realization data? (Chapter 7)

To guarantee designing efficient, feasible, stable and robust timetables, the timetable planning process should integrate multiple models considering different levels of detail to provide accurate input, detailed evaluation and fast computing optimizations. A conceptual timetabling framework is given in Figure 1.2. Given are a line plan (i.e., list of train lines with their stopping stations and frequencies), infrastructure including signalling system and rolling stock characteristics. The framework should include a macroscopic optimization model to solve timetabling problems for complex, large and dense railway networks. Since we also aim for robust solutions, timetables should be tested with stochastic simulation. What is more, they should provide a good trade-off between efficiency and robustness. Microscopic simulation of running times and capacity estimation are necessary to generate the input to the macroscopic models and to evaluate timetable feasibility and stability. Finally, we need data analysis to calibrate rolling stock characteristic parameters to provide reliable input to the overall planning framework.

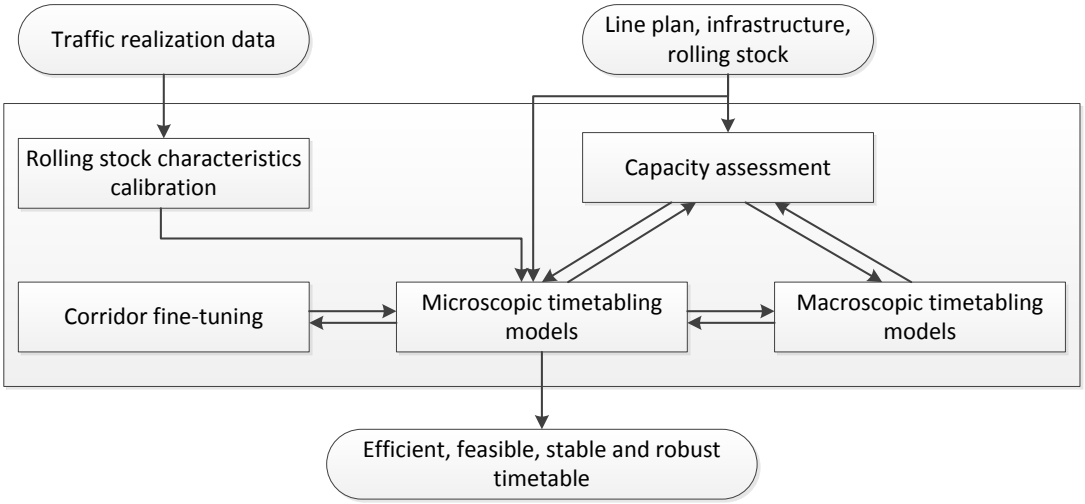

Figure 1.2: Modular multi-level performance-based timetabling framework

[2]The running times that are possible to realize in everyday operations.

1.3 Context

The research of this thesis was supported by the European FP7 project Optimal Networks for Train Integration Management across Europe (ON-TIME, 2016). The project involved infrastructure managers, academic institutions and software companies from France, Germany, Italy, Netherlands, Sweden, Switzerland and the UK. The project aimed at new models and methodologies for improving timetable planning and traffic management to provide better services by more efficient use of the existing infrastructure and reducing train delays.

1.4 Main contributions

The main contributions of this thesis are in the design, optimization, simulation and data analysis for an integrated railway timetabling approach of dense railway networks that incorporates the performance indicators: efficiency, feasibility, stability and robustness.

- *A modular performance-based railway timetabling approach to integrate timetable construction and evaluation into one consistent framework.* The advantage of this approach is that performance indicators are already taken into account during the timetable construction by which the resulting timetable is computed together with all performance measures which are either satisfied or optimized depending on the required criteria. This relieves the exhausting task of ex-ante simulation that some railways apply to test the constructed timetable such as conflicts, stability, robustness and energy consumption. Moreover, it is a notoriously difficult issue for timetable planners to adjust the timetable if the simulation output indicates timetable flaws like existing conflicts or unrealisable train running times. Each local change may have an impact elsewhere.

  In our approach, we replace the feedback from timetable evaluation to timetable adjustment by an integrated approach embedding the timetable evaluation in the construction process. The proposed framework and integrated models are suitable for developing both periodic and non-periodic timetables.

- *Microscopic models for evaluating the microscopic feasibility and stability and resolving conflicts of the macroscopic timetables*. Minimum running times are computed by integrating Newton's motion formula, while accurate headway computation is based on blocking time theory (Hansen & Pachl, 2014). In this way, train process times can be computed very fast, even for very dense railway traffic. Operational running times are calculated by means of an adjusted bisection model that introduces cruising phases at reduced speeds to cover the supplement times imposed by the timetable. The feasibility of the timetable is checked by an efficient conflict detection and resolution model that is based on blocking time theory, and in case of conflicts automatically computes new running and minimum headway times in order to adjust the macroscopic timetable.

In addition, the capacity occupation assessment is realized by a novel max-plus model following the compression method indicated by the UIC Code 406. With this new model, it is possible to compute the capacity occupation in stations as well as corridors. If the capacity occupation satisfies technical thresholds, the timetable is considered to be stable.

- *Macroscopic timetabling model for network optimization*. The developed stability-to-robustness approach is the first to introduce stability together with efficiency and robustness for the periodic timetabling problem. This two-stage approach integrates models for minimizing the cycle time and distribution of time allowances. The model also includes new multi-objective functions for improving timetable efficiency, stability and robustness. We provide a sensitivity analysis and demonstrate that a detailed analysis of weight factors must be considered to generate the best trade-off between efficiency and robustness. We also determine objective functions that allow more flexibility in generating different solutions.

- *The implementation of an iterative micro-macro approach*. This approach incorporates the strengths and advantages of microscopic and macroscopic algorithms to provide an overall effective and reliable solution. Network transformation algorithms are introduced to automatically convert data from the microscopic to macroscopic level and vice versa. A robust network timetable is designed by macroscopic optimization over large networks, including stochastic models for robustness evaluation. This is afterwards converted and analysed at the microscopic level. If track conflicts are detected and/or capacity norms are violated, necessary adjustments to train process times are undertaken by applying procedures of constraints tightening and relaxation. This iterative micro-macro process automatically terminates once the timetable is also microscopically feasible and stable.

- *A new simulation-based optimization method to calibrate the parameters of train running characteristics against observed track occupation data*. This approach derives train speed profiles from real distance-time trajectories collected at discrete points from track-free detection sections. A simulation-based optimization approach calibrates the parameters of the dynamic motion equations describing the tractive effort, the motion resistances, the braking effort, and the cruising phase. These parameters are fine-tuned for different classes of train compositions. A probability distribution is estimated for the input parameters of each class of composition. This also gives insight into different driving behaviour adopted during real operations. A practical application of the train parameter calibration method can be at the planning stage for generating distribution of parameters suitable for robust timetabling design, and in real-time operations for obtaining more reliable predictions of train speed profiles.

With these contributions, this thesis demonstrates the applicability of optimization, simulation and data analysis to efficiently solve relevant practical challenges of railway

traffic management.

1.5 Societal relevance

Mathematical models for automatic generation of timetables can provide reliable train services that use the given infrastructure optimally and can handle daily stochastic disturbances. In addition, the solutions are proven to be conflict-free and could be implemented in practice. The main importance of the proposed models and the modular framework for timetabling is to give planners means to perform their job better, which would lead to a higher level of service to customers. Railway planners can switch the focus from manual and time-consuming timetable design to detailed analyses of multiple automatically generated timetables. This will lead to choosing the best overall solutions that provide better service for passengers and freight operators by reduced delays and more trains running.

The modular framework developed in this thesis focuses on *tactical planning*. By tactical, we refer to planning undertaken well in advance of operations, when a line system is given and the available infrastructure is known. In particular, this thesis solves the problem of finding a basic hour pattern, which is generally performed up to one year ahead. Such basic hour pattern can be easily extended to a complete day timetable by copying the same train sequence.

The developed capacity assessment models can be used on both strategic and operational levels. The developed methods for capacity assessment can determine existing bottlenecks in networks, evaluate the benefits of infrastructure improvement projects and quantify gained additional transport capacity. The possible implications of a capacity assessment could be constructing new infrastructure, improving the existing one, or using the existing one more efficiently. Models for capacity assessment can help on deciding the most cost-effective projects. This would eventually save considerable amounts of money and direct it to the most profitable investments.

By better planned timetables, passengers could expect more trains running on time, short connection times and less delays. Such new timetables would generate better passenger punctuality and highly valued transport services. The timetabling framework can provide also more accurate running times for freight trains to be used in the planning processes. In addition, automatic support tools can be suited for more efficient ad-hoc planning of freight trains. These would lead to more flexibility to freight train operators and more punctual freight trains operations that minimally disturb passenger traffic.

1.6 Collaborations in the thesis

This thesis is a collection of five scientific articles and one book chapter and has been written together with co-authors. This section summarizes the contributions of people engaged with the research in this thesis. The most of the work in this thesis has been

done independently by the author. The author has been responsible for formulating research questions, studying related literature, performing the data analysis, formulating and implementing the models, analysing the results, and writing the chapters and corresponding articles. In the thesis, chapters are based on the following articles:

- Chapter 2: Bešinović, N., & Goverde, R. M. P. Capacity assessment in railway networks, In Borndörfer, R., Klug, T., Lamorgese, L., Mannino, C., Reuther, M., Schlechte, T. (Eds.), *Handbook on Operations Research in Railway Industry*, Springer, accepted.

- Chapter 3: Goverde, R. M. P., Bešinović, N., Binder, A., Cacchiani, V., Quaglietta, E., Roberti, R., & Toth, P. (2016). A three-level framework for performance-based railway timetabling. *Transportation Research Part C: Emerging Technologies*, 67, 62–83.

  Anne Binder contributed in Section 3.3.5 on methodology and writing regarding the model for energy-efficient train driving. Valentina Cacchiani, Roberto Roberti and Paolo Toth contributed in Section 3.3.4, on methodology and writing on the macroscopic model.

- Chapter 4: Bešinović, N., Goverde, R. M. P. & Quaglietta, E. (2017). Microscopic Models and Network Transformations for Automated Railway Traffic Planning. *Computer-Aided Civil and Infrastructure Engineering*, 32 (2), 89–106.

- Chapter 5: Bešinović, N. & Goverde, R. M. P.. A two-stage stability-to-robustness approach to robust periodic timetabling, submitted.

- Chapter 6: Bešinović, N., Goverde, R. M. P., Quaglietta, E., & Roberti, R. (2016). An integrated micro-macro approach to robust railway timetabling. *Transportation Research Part B: Methodological*, 87, 14–32.

  Roberto Roberti contributed in Section 6.5 and the Chapter's Appendices A and B, on implementing the macroscopic timetabling model of Cacchiani, Caprara, and Toth (2010).

- Chapter 7: Bešinović, N., Quaglietta, E., & Goverde, R. M. P., (2013). A simulation-based optimization approach for the calibration of dynamic train speed profiles. *Journal of Rail Transport Planning & Management*, 3(4), 126–136.

1.7 Thesis outline

The remainder of the thesis is structured as follows. Figure 1.3 gives the visual outline of the thesis. Chapter 2 introduces capacity assessment approaches for corridors, stations and networks that are applied in Chapters 3, 4, 5 and 6. It also includes a

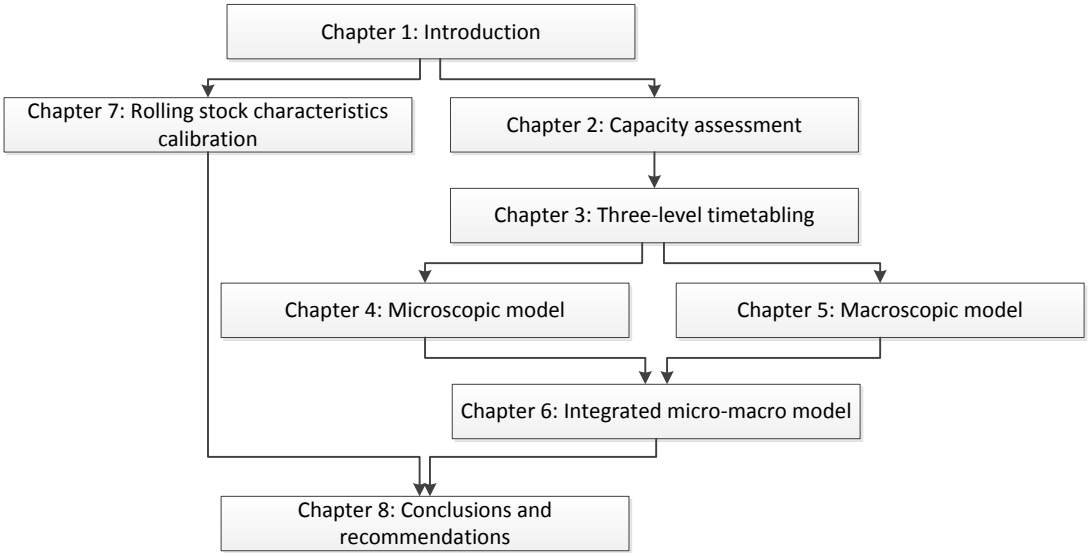

Figure 1.3: Visual outline of the thesis

new approach based on max-plus algebra for corridors and stations. Chapter 3 describes a conceptual framework for performance-based railway timetabling integrating timetable construction and evaluation on three levels: microscopic, macroscopic, and a corridor fine-tuning level. Chapter 4 defines microscopic models for timetable planning and in particular, models for computing accurate input for a macroscopic model, evaluating feasibility and stability of railway timetables and network transformations that allow seamless transitions from microscopic to macroscopic models and vice versa. Chapter 5 develops a new two-stage stability-to-robustness model for computing stable and robust timetables. Chapter 6 integrates the microscopic models from Chapter 4 with another macroscopic timetabling model for feasible, stable and robust timetabling. Chapter 7 presents a simulation-based optimization approach to calibrate the characteristic parameters of the train dynamics from realization data. Chapter 8 gathers the conclusions of this thesis and gives recommendations for future research on designing advanced decision support models for timetabling.

Appendix A demonstrates the graphical user interfaces for micro-macro timetabling and evaluating timetable robustness.

Chapter 2

Capacity assessment in railway networks

This chapter has been accepted for publication as:

Bešinović, N., & Goverde, R. M. P. Capacity assessment in railway networks, In Borndörfer, R., Klug, T., Lamorgese, L., Mannino, C., Reuther, M., & Schlechte, T. (Eds.), *Handbook on Operations Research in Railway Industry*, Springer.

2.1 Introduction

Passenger and freight railway traffic have increased considerably worldwide over the past two decades, and this trend is expected to continue (UNECE, 2015). Many railway networks are already exploited to their maximum capacity and extra measures are needed to satisfy the growing demand. The ON-TIME project has diagnosed multiple capacity issues in several European countries including France, Italy, the Netherlands, Sweden, and the UK (ON-TIME, 2012).

The possible implications of a capacity assessment could be constructing new infrastructure, improving the existing one, or using the existing one more efficiently. Upgrading the infrastructure may achieve these objectives, but is very costly and time-consuming. Therefore, more efficient planning of services may be more appropriate. Thus, understanding railway capacity is important to identify the most effective actions.

Various approaches for capacity assessment can be found in the literature and in practice. For example, RMCon (2012) and Jensen, Landex, Nielsen, Kroon, and Schmidt (2017) deployed simulations for this purpose. Schwanhäußer (1978, 1994) introduced queueing theory approach for evaluating the capacity. The extensions of this approach

are given in Büker and Seybold (2012); Huisman, Boucherie, and van Dijk (2002); Wendler (2007); Yuan and Hansen (2007) and Weik, Niebel, and Nießen (2016). Krueger (1999) and Lai and Barkan (2009) proposed parametric modelling. Analytic approaches based on optimization models for capacity assessment are presented in Burdett and Kozan (2006); Mussone and Calvo (2013) and Burdett (2015). However, none of these models consider a timetable with its scheduled arrival and departure times as an input for the capacity assessment.

Based on extensive practical experience, it has been concluded that timetable structures are required to understand the interactions in a dense and complex railway network. Therefore, timetable structures should be used to determine the required infrastructure in terms of numbers of platforms or tracks (Odijk, Romeijn, & van Maaren, 2006). Mackie and Preston (1998) and Eliasson and Börjesson (2014) also stressed the necessity of timetables for estimating the social benefit of railway investment appraisals. In particular, explicit timetable decisions (e.g., train orders and connections) are required assumptions for the analysis. Otherwise, the results will be arbitrary and scenarios will not be comparable.

This chapter describes the main (timetable-based) methods for capacity assessment that are based on *timetable compression*. Particularly, we focus on timetable-based models that consider infrastructure and rolling stock as given and fixed. In addition, the chapter is oriented towards deterministic models for assessing the level of capacity occupation, rather than the maximum theoretical capacity. For the latter, we refer to Delorme, Gandibleux, and Rodriguez (2009). Section 2.2 introduces the relevant terminology and aspects of railway capacity research. Section 2.3 presents the *compression methods*, the basics of *blocking time theory*, and states the limitations of existing applications. These form the basis for the description of advanced tools for capacity assessment on the different infrastructure levels of corridors (Section 2.4), nodes (Section 2.5) and networks (Section 2.6). Finally, Section 2.7 discusses approaches for improving capacity and gives directions for further development.

2.2 Railway capacity and blocking times

In order to discuss railway capacity, it is important to first give some definitions. Railway capacity is highly complex and depends on multiple factors. The *theoretical capacity* of railway lines and station layouts is defined as the maximum number of train paths (time-distance infrastructure slots) on the infrastructure in a given time window and represents an upper limit for infrastructure capacity. It usually assumes a homogeneous traffic where all trains are identical and optimally spaced throughout the time period (UIC, 2004).

The *practical capacity* of railway infrastructure is defined as the maximum number of train paths on the infrastructure in a given time window given the traffic pattern, operational characteristics or timetable structure. Practical capacity thus depends on the mix of train services with different characteristics.

Table 2.1: Used terminology in railway capacity research

Term	Synonyms
Theoretical capacity	Design capacity (TRB, 2013), absolute capacity (Burdett & Kozan, 2006), capacity throughput (Čičak, Mlinarić, & Abramović, 2012; Sogin, Lai, Dick, & Barkan, 2013)
Practical capacity	Achievable capacity (TRB, 2013), effective capacity (Goverde & Hansen, 2013)
Capacity occupation	Infrastructure occupation (UIC, 2004), occupancy time (UIC, 2013), consumed capacity (Hansen & Pachl, 2014), capacity utilization (Goverde, 2007), carrying capacity (Hu, Li, Meng, & Xu, 2013), used capacity (Abril et al., 2005)
Capacity occupation rate	Utilization rate (Landex, 2009)

Capacity occupation is defined as the amount of time that the train paths from a given timetable structure in a given time window occupy the infrastructure. Commonly, capacity occupation is expressed in minutes. Moreover, the *capacity occupation rate* (expressed in %) is defined as the ratio of capacity occupation to the given time window. It provides an indication of how a timetable may perform. Other measures for quantifying railway capacity found in the literature like the number of passengers over a given time window and amount of goods over a given time window. Table 6.1 gives an overview of the terminology commonly found in railway capacity research.

Railway capacity depends on various aspects that can be categorized in three groups: infrastructure, rolling stock, and traffic. *Infrastructure* is defined by the railway layout (single-track, double-track, number and length of platform tracks), distance between stations, track speed limits (depending on curves, grades and switches), and the signalling system (block lengths, number of signalling aspects, train protection). For example, Goverde, Corman, and D'Ariano (2013) showed the influence of various signalling systems on the capacity occupation. *Rolling stock* characteristics are, among others, train composition (multiple unit or locomotive hauled wagons), length, maximum speed, and traction and braking characteristics. Capacity also depends on *traffic management and operational rules* like dominant train type (passenger, freight or mixed), use of tracks (unidirectional/bidirectional), mix of train services with different characteristics (speed, stopping pattern, frequency), train sequences, dwell times and connections in stations (Strategic Rail Authority, 2014). UIC (2004) explained that capacity depends on the way the infrastructure is utilized which is represented in the capacity balance of the number of trains, the average speed, the traffic heterogeneity, and stability. A detailed analysis of different aspects affecting capacity can be found in Abril et al. (2008); Harrod (2009); M. J. Schmidt (2014); Shih, Dick, Sogin, and Barkan (2014) and Lindfeldt (2015), while an empirical comparison of different ca-

pacity assessment methods can be found in Rotoli, Navajas Cawood, and Soria (2016).

Due to the high complexity of capacity assessment, railway infrastructure is often decomposed and assessed independently (Pachl, 2014). We distinguish different infrastructure segments such as nodes, line sections (corridors) and networks. A *node* is a track layout with switches and multiple route possibilities. A node may be a small station with only a few platform tracks and limited interlocking areas, but also a big station with higher number of tracks and more complex interlockings, and may serve as a terminal for train lines. In addition, a junction can be considered as a node, which includes only interlocking but does not provide train stopping possibilities. A *line section* is a railway line between two nodes with a fixed number of parallel tracks and no switches. A line section can have one or more parallel tracks and the sequence of trains cannot change. Trains on a line section are usually separated by a block system, where each block can be allocated to at most one train. A *corridor* represents a longer railway line that consists of multiple line sections. Finally, a *network* is an area of various interconnected corridors which are considered at once during the capacity assessment.

2.2.1 Blocking times

The concept of blocking times (Pachl, 2014) is closely related to capacity assessment and the basis for the remainder of this chapter. A *resource* represents a subset of infrastructure elements that is exclusively allocated to a single train at a given time. In practice, this is a block section or an interlocking route section including one or more switches or crossings. A *train route* defines a set of consecutive resources that can be used by a train to traverse from one point to another (e.g., between two stations). A (time-distance) *train path* extends the train route with the time the route is used.

The *blocking time* of a resource is the time during which the resource is solely dedicated to a single train and cannot be used by any other. The blocking time consists of an approach, running and clearing time, corresponding to the train running time from the approach signal to the point located train length away the signal at the end of the block. In addition, the blocking time includes setup and release times of the route and signals, as well as the driver sight and reaction time before the approach signal. Figure 2.1 illustrates a blocking time computation for a single resource (i.e., block section) of a running train.

The successive blocking times over a train route form a *blocking time stairway*, which can be computed for all train paths of a given timetable. Generally, a timetable consists of arrival and departure times at nodes, defining *scheduled running time*, which includes running time supplement. For computing blocking times, we need running times over each resource, which are obtained by computing an exact train time-distance speed profile corresponding to a feasible dynamic speed profile for a given scheduled running time. Figure 2.2 illustrates the conversion from timetable departure/arrival times to a train dynamic speed profile and blocking time stairway. The modelling details of the macroscopic to microscopic conversion are presented in Bešinović, Goverde, and Quaglietta (2017).

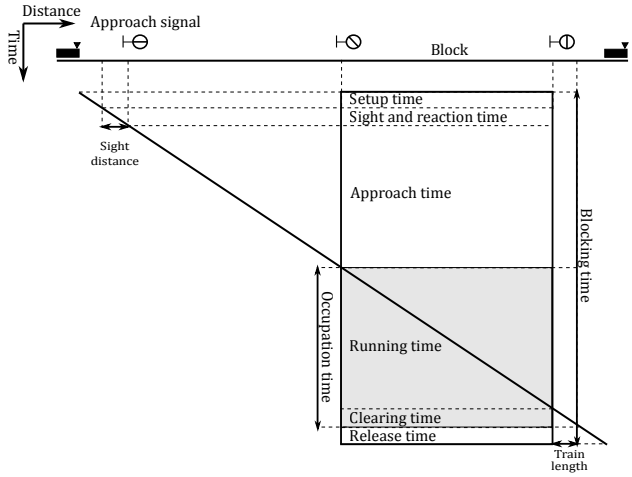

Figure 2.1: Blocking time for a running train over a block section defined by two signals and the corresponding approach signal

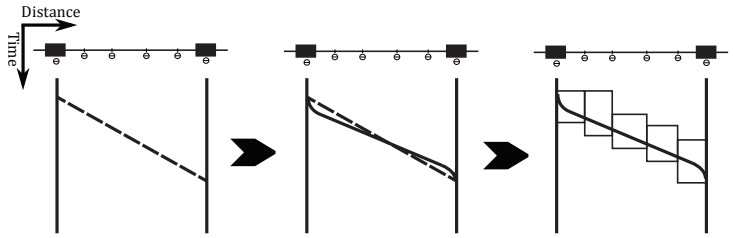

Figure 2.2: Macro to micro conversion: from time-distance line to blocking time stairway between two stations on a single track with five block sections

Blocking time stairways are applied to compute minimum headways. The minimum headway time h_{ijs} between trains i and j on a corridor or node z is computed as

$$h_{ijz} = \max_{k \in R_{ijz}} (f_{ik} - s_{jk}), \tag{2.1}$$

where R_{ijz} are the resources used by both i and j in corridor or node z, and s_{jk} and f_{ik} are the associated start time and end time of the blocking time for resource k, respectively. We assume that i precedes j and both stairways have the same reference, namely, time 0. If z is a corridor, then we obtain the *minimum line headway time* between the two trains; and if it is a node, then it is a *minimum station headway time*. The resource that defines a minimum headway time is called a critical resource, such as, the critical block between two compressed blocking time stairways is the block where the stairways touch each other.

2.3 Existing methods in practice

In Europe, the two most common analytic approaches for capacity assessment are based on the *timetable compression method*. Timetable compression is the process of shifting train paths to each other as much as possible, bringing them to the time

distance of minimum headway times. The total time needed for operating such a compressed timetable is the capacity occupation. Here, the *minimum headway time* is the minimum time separation between two train paths that provides conflict-free train runs. The first method has been proposed by the International Railway Association UIC – the UIC 406 capacity method (UIC, 2004). The second method is the British Capacity Utilization Index (CUI) method (Gibson, Cooper, & Ball, 2002). Meanwhile, in the US, a timetable compression method has not been applied yet (Pouryousef, Lautala, & White, 2015).

2.3.1 UIC 406 capacity method

The UIC 406 capacity method is based on the blocking time theory. Originally, UIC (2004) described a method for evaluating capacity of line sections. In the 2nd edition, UIC (2013) expanded the approach to the capacity assessment of nodes. The method requires a timetable and a division of the network into line sections and nodes. The original purpose of the UIC 406 capacity method was to measure the capacity occupation of a given timetable, which is achieved by compressing the train blocking time stairways. In addition, the method has been used for assessing practical capacity. This has been done by adding extra trains in the timetable, called *timetable enrichment*.

The UIC 406 capacity method intends to standardize evaluations for obtaining comparable examination results by defining recommended values for the capacity occupation rate of a line section (UIC, 2004). The recommended capacity occupation rates have been proposed only for double tracks and are distinguished between a) dedicated suburban passenger traffic, dedicated high-speed lines, and mixed traffic lines and b) peak period and daily period. Suggested capacity occupation rates are 85% and 70% for dedicated suburban traffic (peak and daily period), while they are 75% and 60% for dedicated high-speed lines and mixed traffic lines. UIC (2013) proposed some preliminary ranges for nodes, but these still have to be confirmed. It is assumed that these occupation rates would guarantee stable services with respect to small disturbances. These recommendations were based on the practices among European infrastructure managers (IMs) at the time, but highly depend on the infrastructure layout, the way it is utilised, and the typical size of delays. Recommended capacity occupation rates are referred to as *saturation rates* (Abril et al., 2008), while a corridor that reaches these rates is called a *saturated corridor*.

If a corridor is not saturated yet, additional trains may be added. This is done through an iterative process. First, the capacity occupation is computed by timetable compression. If the rate is smaller than the saturation rate, the timetable is enriched by one or more trains. Then, the capacity occupation rate is reassessed. These iterations are repeated until the corridor has been saturated. In addition, enriching can be used to determine a corridors' theoretical capacity. For further details on the enrichment process, see Delorme et al. (2009) or Jensen et al. (2017).

2.3.2 CUI method

The CUI is the measure based exclusively upon the headway norms in nodes, given as Timetable Planning Rules (Network Rail, 2015). Similar to UIC 406, the CUI method builds on a network decomposition into line sections that are evaluated separately by compressing the timetable for each infrastructure segment. A line section for CUI is always determined by two neighbouring nodes, while it may be longer for the UIC 406 method. The method does not consider an exact infrastructure occupation based on blocking times, which makes it less accurate than the UIC 406 method. Thus, we refer to CUI method as to a *normative capacity assessment*. A further comparison between UIC 406 and CUI may be found in Melody (2012) and ON-TIME (2012).

2.3.3 Open challenges

Recently, Lindner (2011) evaluated the UIC 406 capacity method. The 2nd edition (UIC, 2013) improved on his observations partially. One of the main remaining limitations of the UIC 406 method is the capacity assessment in nodes. It proposes to decompose a node in switch areas and (platform) track areas, and evaluate each segment independently. More recently, Rotoli et al. (2016) gave a descriptive simplified approach for evaluating nodes by using this decomposition and assuming a general node layout. Such a node decomposition may not consider all route dependencies and leads to underestimated capacity occupation. Section 2.5 introduces an analytic model that overcomes this issue.

A second limitation is due to the network decomposition to line sections which causes certain train dependencies to be neglected and result again in an underestimated capacity occupation. Third, the lengths of the decomposed line sections affect the resulting capacity occupation significantly. To overcome these challenges, we propose a network model for capacity assessment that preserves microscopic details of the infrastructure and all train dependencies (Section 2.6). Fourth, the given saturation rates represent a rough guideline rather than an exact values to follow. These rates are highly dependent on the infrastructure layout, train characteristics and level of service; and they may vary significantly for different national networks. However, additional research is necessary to achieve better insight.

Armstrong, Preston, and Hood (2015) proposed a solution for the limitation of the CUI method, which is mainly applicable on line sections, an extension for assessing the capacity in nodes. However, due to the coarser level of detail, CUI is a less accurate and rather cumbersome method that is difficult to apply to complex nodes. Following the timetable planning requirements defined by European IMs (ON-TIME, 2014), we encourage using the UIC 406 capacity method for further capacity analyses.

2.4 Capacity assessment of corridors

The compression method is quite easy to apply and should allow a natural deployment. However, only the capacity assessment of corridors is straightforward. To that purpose, various analytical and simulation models have been developed. Landex (2009) extended the UIC 406 method to single tracks, while Abril et al. (2005) applied it on double-track corridors. Čičak et al. (2012) proposed an approach for theoretical capacity of single track lines using a normative compression method. Abril et al. (2008) and Pouryousef et al. (2015) are suggested for further reading on implementations of capacity assessment for corridors in Europe and the USA.

However, only a few of them incorporate a compression method explicitly to evaluate the capacity use of a given timetable, such as RailSys (RMCon, 2012) and EGTRAIN (Quaglietta, 2014).

2.5 Capacity assessment of nodes

In this section, we describe the *max-plus automata* model for capacity assessment in nodes and give a numerical example (Section 2.5.1). Max-plus automata combine properties of the heaps-of-pieces theory and max-plus algebra, and were introduced by Gaubert and Mairesse (1999). The max-plus algebra is a mathematical technique to model and analyse discrete event dynamic systems (DEDS) such as railway systems. We refer to Goverde (2007) and Heidergott, Olsder, and van der Woude (2014) for more details on max-plus algebra applied to railways.

One of the main advantages of max-plus automata is that it explicitly model the infrastructure resources and the blocking times of these resources corresponding to blocking time stairways. This is exactly what is required to compute the capacity occupation of a set of resources by a given set of train paths. Differently from the general max-plus algebra, in the max-plus automata, both the start and end time of each resource by each train is taken into account.

We assume a given timetable with assigned train routes (i.e., a *route plan*) and corresponding blocking time stairways for the trains. In this section, we view a blocking time stairway of a single train as a *piece*. Note that a piece may represent a complete or partial train route through a node. For example, a train route may consist of multiple pieces. Graphically, we may picture a compressed timetable as a *heap* of all blocking time stairways stacked on each other, a *heap-of-pieces*.

2.5.1 Max-plus automata model

A max-plus algebra is a semiring over $\mathbb{R}_{\max} = \mathbb{R} \cup \{\varepsilon = -\infty\}$, equipped with the two binary operations maximum ($\oplus$) and addition ($\otimes$). For $a, b \in \mathbb{R}_{\max}$ the max-plus operations are defined as

$$a \oplus b = \max(a, b) \quad \text{and} \quad a \otimes b = a + b. \tag{2.2}$$

The element $\varepsilon = -\infty$ is the neutral element for $\oplus$ and absorbing for $\otimes$. The element $e = 0$ is the neutral element for $\otimes$. Many properties of max-plus algebra are similar to conventional algebra. The scalar max-plus operations are extended to matrices in a standard way. Let $\mathbb{R}_{\max}^{n\times n}$ be the set of $n \times n$ matrices with elements in $\mathbb{R}_{\max}$. Then, for any matrices $A = (a_{ij})$, $B = (b_{ij}) \in \mathbb{R}_{\max}^{n\times n}$ matrix addition $\oplus$ and matrix multiplication $\otimes$ are defined as

$$[A \oplus B]_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij}), \tag{2.3}$$

$$[A \otimes B]_{ij} = \bigoplus_{k=1}^{n} a_{ik} \otimes b_{kj} = \max_{k=1,..,n} (a_{ik} + b_{kj}). \tag{2.4}$$

A max-plus automaton is a tuple $H = (T, R, M, s, f)$. Here, T is a finite set of tasks that represent all train routes $l \in T$, while R is a finite set of resources that can be block sections or track detection sections (as defined in Section 2.2.1). Also, M is a function that maps a task to the resources it uses. Formally, M is a morphism $T \to \mathbb{R}_{\max}^{R\times R}$ defined uniquely by a finite family of matrices $M(l)$, $l \in T$. We define $s_i(l)$ and $f_i(l)$ as the start and end time of resource i used by task l, respectively. Further, these construct the corresponding R-dimensional row vectors $s(l)$ and $f(l)$. In other words, the task l represents a (partial) train route, while $s(l)$ and $f(l)$ depict the upper and lower contour of the corresponding blocking time stairway. We also assume that each stairway starts at time 0.

The matrix $M(l)$ represents the blocking time stairway, which also equals the capacity occupation, of a task l and is defined as

$$M_{ij}(l) = \begin{cases} e, & \text{for} \quad i = j, i \notin R(l), \\ f_j(l) - s_i(l), & \text{for} \quad i, j \in R(l), \\ \varepsilon, & \text{otherwise.} \end{cases} \tag{2.5}$$

A matrix element $M_{ij}(l)$ gives the time difference between the end time of the resource j and start time of the resource i. In addition, if a resource is not used, we assign e, if $i = j$, and ε elsewhere.

We define a route plan w as an ordered sequence of tasks by successive trains $w = l_1 \cdots l_n$, where $l_1, \ldots, l_n \in T$. Then, tasks from the route plan are added one by one to the heap of pieces by which the occupation of the resources is computed sequentially as

$$M(w) = M(l_1 \cdots l_n) = M(l_1) \otimes \cdots \otimes M(l_n). \tag{2.6}$$

Thus, matrix $M(w)$ defines the capacity occupation used by all train routes in w compressed together. Moreover, we define $x(e)$ as an empty schedule of length $|R|$. Then an upper contour $x(w)$ of schedule w is given as

$$x(w) = M(w) \otimes x(e).$$

In general, schedule w represents a given train mix (number and types of trains with corresponding routes). For practical reasons, the first train may be added as an additional train at the end of the sequence. The start time of this final train is the end point of the capacity occupation. In case of a periodic timetable, adding this first train from the next period is required, as it determines the earliest possible time to schedule the next period, which completes a full cycle. This will guarantee a necessary separation between the last train of the current period and the first of the next one. To do so, let a be the first task in a schedule of tasks w. Then the capacity occupation $\mu(w)$ of a schedule w is computed as

$$\mu(w) = \min_{i \in R(a)} \left(x_i(wa) - (f_i(a) - s_i(a)\right), \tag{2.7}$$

where wa is the schedule for one period w with an additional train route a that belongs to the next period. We use an added train route a to determine the earliest possible start of the next period. Here, $x(wa)$ represents the capacity occupation including repeated train a. However, as mentioned, the actual occupation is defined until the start time of a, so we subtract the occupation time of a from $x(wa)$, that is, the difference $f(a) - s(a)$. Next, the capacity occupation $\mu(w)$ is defined between the start time of each element of the first train in w and a. Accepting that w starts at 0, then μ can take the minimum value of the vector $(x(wa) - (f(a) - s(a))$. So, (2.7) computes the occupation μ of a node for a given route plan w that specifies an ordered sequence of blocking time stairways $l \in T$. Note that the model complexity depends on the route choices and not on the station layout complexity, so the set R can be limited to the set of used resources in the given route plan.

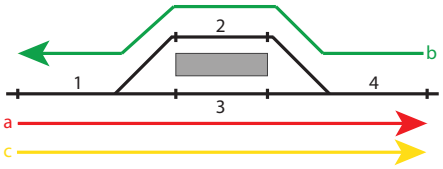

Figure 2.3: Example 1: Simple node infrastructure with trains a, b and c

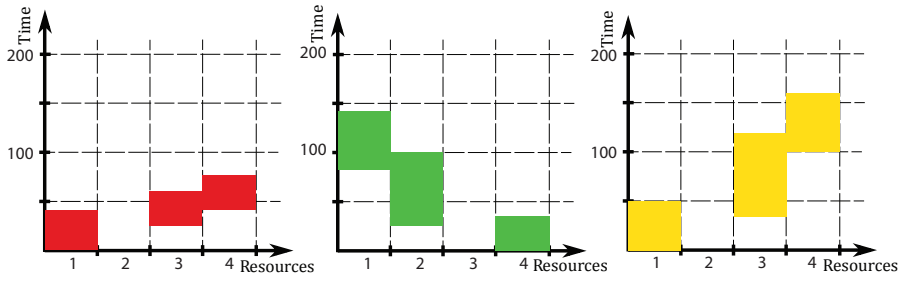

Figure 2.4: Train routes: a – red, b – green and c –blue

Consider the following example for computing the capacity occupation of the node presented in Figure 2.3. Consider three trains a, b, and c, timetable $w_1 = abc$ and

resources $r = 1,..,4$. Trains a and c use resources $\{1,3,4\}$, while b uses $\{4,2,1\}$. Note that the order of resources defines the direction of each train. The train blocking times are given in the blocking stairways (in seconds) as follows:

Route r	$s(r)$	$f(r)$
a	$[0,\varepsilon,25,40]$	$[40,\varepsilon,60,75]$
b	$[80,25,\varepsilon,0]$	$[140,100,\varepsilon,35]$
c	$[0,\varepsilon,35,100]$	$[50,\varepsilon,120,160]$

Figure 2.4 shows individual train routes, i.e., pieces, of a, b and c. Each piece is physically connected in reality. However, it does not have to be connected in a two-dimensional plot since the horizontal axis reports all resources in the node (Figure 2.3) and more, these resources can be ordered randomly. So, if a train route does not use a resource, a 'gap' in a piece may be observed. For example, resource 2 is not used by train route a.

The corresponding matrices M for train routes a, b and c are defined by applying (2.5) as follows

$$M(a) = \begin{bmatrix} 40 & \varepsilon & 60 & 75 \\ \varepsilon & e & \varepsilon & \varepsilon \\ 15 & \varepsilon & 35 & 50 \\ 0 & \varepsilon & 20 & 35 \end{bmatrix}, \quad M(b) = \begin{bmatrix} 60 & 20 & \varepsilon & -45 \\ 115 & 75 & \varepsilon & 10 \\ \varepsilon & \varepsilon & e & \varepsilon \\ 140 & 100 & \varepsilon & 35 \end{bmatrix},$$

$$M(c) = \begin{bmatrix} 50 & \varepsilon & 120 & 160 \\ \varepsilon & e & \varepsilon & \varepsilon \\ 15 & \varepsilon & 85 & 125 \\ -50 & \varepsilon & 20 & 60 \end{bmatrix}.$$

The matrix M for a partial route plan ab is computed as

$$M(ab) = M(a) \otimes M(b) = \begin{bmatrix} 215 & 175 & 60 & 110 \\ 115 & 75 & \varepsilon & 10 \\ 190 & 150 & 35 & 85 \\ 175 & 135 & 20 & 70 \end{bmatrix}.$$

Matrix $M(ab)$ defines the capacity occupation of ab, representing that a is immediately followed by b. Similarly, train route c is added to the route plan as $M(abc) = M(ab) \otimes M(c)$. The upper contour of the route plan $abca$ is then computed as $x(abca) = M(abca) \otimes x(e) = (375,175,395,410)^T$. And the capacity occupation for the route plan abc is then computed using (2.7) as

$$\mu(abc) = \min(x(abca) - (f(a) - s(a))) = \min \left(\begin{bmatrix} 375 \\ 175 \\ 395 \\ 410 \end{bmatrix} - \begin{bmatrix} 40 \\ \varepsilon \\ 35 \\ 35 \end{bmatrix} \right) = 335,$$

where the minimum is taken over the vector entries. Figure 2.5 shows the final result.

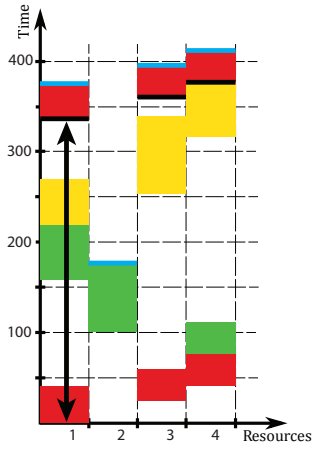

Figure 2.5: Capacity occupation for a route plan $w_1 = abc$. The upper contour $x(abca)$ is showed by the blue line. The capacity occupation $\mu(w)$ is presented with a double arrow.

2.5.2 Satisfying additional timetable constraints

Since the max-plus automata model takes train routes in temporal order and compresses them one by one, some additional modeling is necessary to properly represent certain train interactions. In particular, we propose procedures needed for modelling train overtaking and connections due to passenger transfers or train coupling/decoupling. Meanwhile, constraints for a train turning in a terminal station do not request any extra modelling.

Overtaking of a slower train and/or a lower priority one is applied as a common measure for reducing capacity occupation in the planning phase, and alleviating train delays during operations. If a train is overtaken in a node, then the train route is partitioned in an inbound route and an outbound route. The inbound route is a train route from the node entry point to the platform track, while the outbound route runs from the platform track to the exit from the node. Coupling or decoupling of trains can be treated similarly.

A timetable often includes constraints that represent traffic requirements such as passenger transfers, which are not necessarily related to the infrastructure limitations. In other words, connecting trains often use a dedicated infrastructure. In order to maintain the timetable dependency in the max-plus automata model, additional modelling is necessary to keep the two trains together. To do so, train routes of these trains are modelled as a single task.

2.6 Capacity assessment in networks

Capacity assessment of railway networks is not a general practice yet. CAPRES (Lucchini, Rivier, & Emery, 2000) is a railway network capacity assessment tool based on saturation of a periodic timetable with extra train paths. PETER (Goverde, 2007, 2010) is an analytical tool for evaluating the capacity occupation rate and stability of a periodic timetable on the network level based on max-plus algebra. These models are based on a macroscopic network description and were originally developed for normative headway times, like the CUI. On the other hand, KABAN (Ekman, 2011) is a microscopic capacity assessment tool built on a detailed modelling of infrastructure, periodic timetable and train routes, which also applies max-plus algebra to compute the capacity occupation. However, due to the high level of details, KABAN is limited only to small-sized networks. This section focuses on the general max-plus algebra modelling, such as used in PETER. For similar approaches, see also Heidergott and de Vries (2001) and Heidergott et al. (2014). Note that instead of using normative headway times, we also explain how to deploy the results of capacity assessments of corridors and nodes as input to the network capacity assessment. This provides improved accuracy similar to UIC 406 over the CUI method, and allows evaluating large-scale national networks.

When considering large-scale networks, the microscopic detail of the capacity assessment of corridors and nodes is aggregated into a macroscopic model that connects all the corridors together at the nodes. For the capacity assessment at corridor level, the train paths were split in parts and tackled separately over the successive line sections. At the network level, the successive train paths must again be considered as a whole. Likewise, existing interactions between various train paths at nodes, over successive or crossing corridors, must be regarded at the network level as well. Since the resources were already taken into account at the corridor and node level, the network model can be formulated using only time constraints. On the other hand, for a normative capacity assessment, the events can be any arrival, departure or passing-through events in the network which are connected by minimum running and dwell times or normative minimum headway times. Moreover, on the network level other operational constraints can be taken into account, such as passenger transfers and rolling stock connections.

In general, the *network model* consists of event times at nodes and precedence constraints between them, which represent the interconnection structure of the various trains. Before an event time may occur, it must satisfy all precedence constraints which take the form

$$x_i \geq a_{ij} + x_j, \tag{2.8}$$

where x_i and x_j are two event times and $a_{ij} \geq 0$ is the minimum time duration from event time x_j to event time x_i. This precedence constraint is very general and can be used to define a directed acyclic graph (DAG) with the event times as the nodes and the minimum time durations as the arc weights between the nodes. The minimum time durations may correspond to minimum line or station headway times, or to scheduled

activities between events such as the aggregated running time between nodes or a minimum dwell or transfer time in a node (see Figure 2.6). A critical path between two nodes in a graph represents the longest path between the two nodes, that is, the path with the highest sum of weights. A critical path algorithm over the DAG then finds the earliest occurrence times of all the events in the network which correspond to a compressed timetable with all precedence constraints respected. Finding a critical path in DAG can be done by any shortest path algorithm after negating the weights.

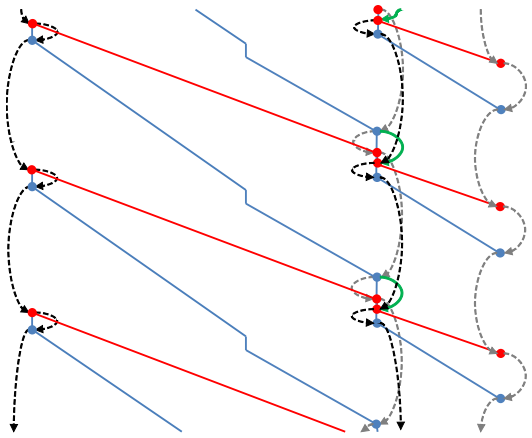

Figure 2.6: Modelling timetable constraints in a network including event times (dots), runs, stops and transfers (solid arcs), and minimum headway times (dashed arcs)

For periodic timetables, it is more convenient to consider periodic event times and assess the network capacity occupation in a basic timetable period. For this, an event i represents a triple $i = (E_i, L_i, S_i)$, where E_i is the event type (arrival or departure), L_i is the associated train line and S_i the station. Denote by $x_i(k)$ the event time of a periodic event i in timetable period k. So, the event time of an event i in the first period is $x_i(1)$, in the second period it is $x_i(2)$, and so on. If the events occur on time with a scheduled cycle time T, then $x_i(k+1) = x_i(k) + T$. Now the precedence constraints can be written as

$$x_i(k) \geq a_{ij} + x_j(k - m_{ij}), \tag{2.9}$$

for all predecessor events j of i, where a_{ij} is the same for each period and m_{ij} is a non-negative integer indicating the period shift between the two events. For example, event j is scheduled m_{ij} periods before event i. Mostly, $m_{ij} \in \{0,1\}$, corresponding to two events that are scheduled in the same period ($m_{ij} = 0$) or in successive periods ($m_{ij} \geq 1$), so that the time separation crosses a period boundary. Any scheduled activity that covers more than one period can be split in parts with dummy events, so in the sequel we assume $m_{ij} \in \{0,1\}$.

The earliest occurrence of an event time is now obtained by

$$x_i(k) = \max_j \big(a_{ij} + x_j(k - m_{ij})\big), \tag{2.10}$$

where j ranges over the predecessors of i. This can be formulated conveniently in

max-plus algebra. Let $x(k) = \big(x_1(k), \ldots, x_n(k)\big)'$, and collect the minimum activity and headway times in two matrices A_0 and A_1, with $[A_{m_{ij}}]_{ij} = a_{ij}$ and fill the empty entries by $\varepsilon = -\infty$ indicating that there is no direct precedence relation from event j to i. If there are parallel arcs between the same events with the same period shift, then only the maximum arc weight has to be added to the matrix. The recursive equation (2.10) can now be written for all events together as $x(k) = A_0 \otimes x(k) \oplus A_1 \otimes x(k-1)$, where k ranges over the successive timetable periods. It is a straightforward result from max-plus algebra theory that any max-plus system can be reformulated as a purely first-order system of the form

$$x(k) = A \otimes x(k-1), \tag{2.11}$$

where $A = A_0^* \otimes A_1$, with the Kleene star operator $A^* = A^0 \oplus A^1 \oplus \ldots \oplus A^{n-1}$ and the powers are understood in the max-plus algebra, e.g., $A^2 = A \otimes A$ (Heidergott et al., 2014). For simplicity, we assume that A is irreducible, meaning that it corresponds to a strongly-connected precedence graph defined by n nodes and an arc (j,i) with arc weight a_{ij} for all entries $a_{ij} \neq -\infty$. For the general results, see Goverde (2007).

The main result from the max-plus algebra approach is that the network capacity occupation equals the eigenvalue λ of the system matrix A. The eigenvalue problem is defined as

$$A \otimes v = \lambda \otimes v, \tag{2.12}$$

where v is an eigenvector corresponding to the eigenvalue λ. The eigenvector v represents a compressed timetable allowing the railway system to operate with cycle time λ. To see this, we write (2.12) in conventional form as

$$\max_j \big(a_{ij} + v_j\big) = \lambda + v_i. \tag{2.13}$$

Considering v as a timetable vector in some period, then the left-hand side gives the earliest occurrence time for event i in the next period and the right-hand side says that this occurrence time is exactly λ after the previous event time v_i. If $G(A)$ is strongly connected, then the eigenvalue λ is unique (Goverde, 2005; Heidergott et al., 2014), and so (2.13) holds for each v_i with the same λ. Since the a_{ij} are the minimum activity and headway times, v is the compressed timetable, and λ is the network capacity occupation.

A *critical circuit* is a circuit in the precedence graph with the maximum ratio of total arc weight to the number of arcs in the circuit, which equals λ. To obtain a stable timetable that can cope with delays, the timetable must be operated with a period length $T > \lambda$. The events on the critical circuit also identify the critical activities and headways in the network, similar to the critical blocks in the capacity assessment of corridors. Figure 2.7 shows a large network where the critical circuit is the traffic over a partial single-track line. Efficient algorithms are available for solving the max-plus eigenvalue problem, and in particular graph algorithms based on the critical circuit

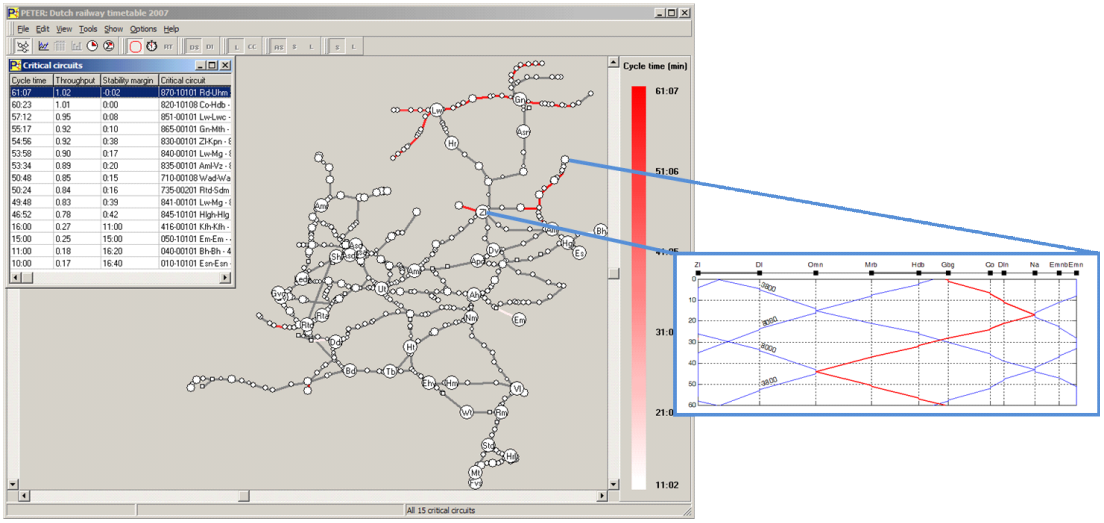

Figure 2.7: Critical circuit in a large network (PETER)

(Goverde, 2005; Heidergott et al., 2014). For example, the policy iteration algorithm runs in $IO(m)$ time where I is the number of iterations of the main loop and m is the number number of arcs (Cochet-Terrasson, Cohen, Gaubert, McGettrick, & Quadrat, 1998).

2.7 Conclusions and future developments

Railway capacity research plays an important role in railway planning and operations. In this chapter, we gave an overview of methods for railway capacity assessment, with the focus on deterministic timetable-based models. We first presented common methods based on timetable compression, UIC 406 and CUI. The CUI is a normative method while the UIC 406 model considers a higher level of detail that allows more accurate estimation of capacity occupation. We also described the existing and advanced models for assessing different infrastructure segments independently like corridors and nodes, but also whole networks.

The benefit of capacity assessment is manifold. First, evaluating existing or new timetables to determine capacity occupation will provide insight into the expected level of service. Second, the infrastructure bottlenecks can be determined in a network. Third, capacity assessment may suggest possible improvements in traffic organization like using alternative train routes that are more efficient. Fourth, proposing the most attractive and beneficial infrastructure projects based on capacity assessment is particularly valuable to infrastructure managers and governments in using available funds most efficiently. Fifth, the impact of scheduled construction and maintenance works on traffic can be estimated.

The future development of capacity assessment models should stay in line with the existing compression method, the UIC 406. To make it a standard evaluation tool and

apply it internationally, additional research on capacity saturation rates and required levels of service for punctuality and regularity is essential. The network models should gain more attention, as only these are able to incorporate all interactions occurring in a timetable. In addition, it is necessary to maintain the high level of accuracy by transforming data from microscopic to macroscopic models.

Chapter 3

A three-level framework for performance-based railway timetabling

---

Apart from minor updates, this chapter has been published as:

Goverde, R. M. P., Bešinović, N., Binder, A., Cacchiani, V., Quaglietta, E., Roberti, R., & Toth, P. (2016). A three-level framework for performance-based railway timetabling. *Transportation Research Part C: Emerging Technologies*, 67, 62-83.

3.1 Introduction

The performance of railway operations depends highly on the quality of the timetable. In the last decade, timetabling software has become more and more common, from running time computations via mathematical timetable optimization to railway operations simulation. Nevertheless, these tools and their focus vary widely from country to country and often lack consistency since they are used independently for different purposes and do not lead to an integrated set of tools geared towards a well-defined timetable design process. Also many papers on railway timetabling have been published in the scientific literature, but they hardly found their way into practice. The same could be said for train rescheduling models. This raises the question: what would be needed to get the scientific state-of-the-art implemented and applicable to practical (re)scheduling? This question was the basis for the EU FP7 research project ON-TIME (Optimal Networks for Train Integration Management across Europe) (ON-TIME, 2016). Although wider in scope, one of the aims was to develop improved methods for the construction of timetables that are capable of coping with normal statistical variations and minor perturbations in operations. The result is a three-level framework for performance-based timetabling which is explained in this paper.

A state-of-the-art review of literature and practice revealed a lot of research in mathematical models for macroscopic timetable optimization (ON-TIME, 2013), see also the review papers by Bussieck, Winter, and Zimmermann (1997), Cordeau, Toth, and Vigo (1998), Caprara, Kroon, Monaci, Peeters, and Toth (2007), and Lusby, Larsen, Ehrgott, and Ryan (2011). These macroscopic models rely implicitly on reliable input data which may not always be available. This might explain why these models and algorithms did not yet find their way into daily timetabling practice, except at the strategic level. A recent trend in the scientific literature consists of robust timetabling models that incorporate stochasticity or uncertainty in the input (Cacchiani & Toth, 2012). Microscopic timetabling models that use a higher level of detail are limited in the literature and mainly focus on single track railways (e.g., Brännlund, Lindberg, Nou, & Nilsson, 1998). Also models based on blocking time theory (Hansen & Pachl, 2014) fall within this category. Most of these blocking time models are employed for computing capacity consumption using the timetable compression method or within microscopic simulation tools. Moreover, optimization models based on blocking times have been developed for real-time rescheduling (e.g., Corman & Meng, 2015; Corman & Quaglietta, 2015; D'Ariano, Pranzo, & Hansen, 2007; Meng & Zhou, 2014). Recent papers apply two-level microscopic-macroscopic models to generate conflict-free timetables (Caimi, Chudak, Fuchsberger, Laumanns, & Zenklusen, 2011; Gille, Klemenz, & Siefer, 2008; Schlechte et al., 2011). In these papers, the transformation from microscopic to macroscopic models is straightforward but the reverse is more complicated.

The timetabling practice shows a similar separation, with either macroscopic models to compute network timetables using normative input, or microscopic blocking-time based tools for detailed planning on corridors and stations but without support for network optimization. Timetable evaluation on feasibility, stability or robustness is typically applied –if at all– after timetable construction using simulation tools with unclear procedures how the results are used to improve the timetable design. Timetabling tools are mostly concerned with routine work such as running time calculations, mostly discarding energy-efficiency, and making visualizations such as time-distance diagrams and platform occupation diagrams. Some railways (SE, UK) are starting to apply microscopic simulation tools for conflict detection as a complementary step to their macroscopic timetable planning tools. If a significant change of the timetable is foreseen either for lines or for complicated areas, robustness simulation studies are made also to ensure the feasibility of the timetable and give a rough idea of its robustness (ON-TIME, 2013).

Based on the state-of-art review essential performance measures were derived that should be taken into account to achieve a good timetable (Goverde & Hansen, 2013). These performance indicators include infrastructure occupation, stability, feasibility, robustness, resilience, journey time efficiency and energy efficiency. Depending on the degree that these indicators are taken into account in the timetable design process, a higher timetabling level can be obtained that lead to better timetables but at the cost

of increased data requirements (Goverde & Hansen, 2013).

In this paper, we propose to integrate timetable construction and timetable evaluation with the aim to incorporate all timetable performance indicators in the timetable design process and thus achieve the highest timetabling level. Using a microscopic model for large-scale networks including stochastic elements to evaluate and optimize a timetable is practically impossible due to abundant level of details and the size of real-life problems. But this is also not necessary, since the timetable performance indicators apply to different levels of timetabling detail and therefore we may zoom in and out at different levels to optimize or evaluate the various performance indicators. We therefore propose an integrated timetable design process at three levels:

- A microscopic level based on accurate running time and blocking time calculations using train dynamics, infrastructure characteristics, and signalling logic. This level is required for evaluating feasibility, infrastructure occupation, and stability.
- A macroscopic level based on an aggregated network structure of main timetable nodes only. This level is required for optimizing and evaluating journey time efficiency and robustness over large-scale networks.
- A mesoscopic level for fine-tuning the train speed profiles on corridors between the main nodes. This level is required for optimizing energy-efficiency and robustness on corridors between the main nodes.

The mesoscopic level gets input from both the microscopic and macroscopic levels and may use a mixture of microscopic and macroscopic models itself. In particular, at this level the train speed profiles over the corridor are optimized taking into account the available time allowances computed in the other levels. Finally, a consistent datastructure is important to switch between the three levels. In particular, the microscopic models compute reliable input to the macroscopic and mesoscopic models, and reversely the macroscopic timetables need to be translated back to the microscopic level for microscopic evaluation and further fine-tuning in the corridors.

This paper presents an innovative three-level modular timetabling framework to integrate the three levels of timetable optimization and evaluation into a consistent design process. We propose an iterative process on the three levels, where each performance indicator is optimized or evaluated at the appropriate level. As a proof of concept this framework has been implemented with a set of algorithms that are described in this paper from a functional perspective as examples of a possible implementation. The implemented models and algorithms are state of the art but it is their interaction in the framework that is the main contribution of this paper. The railML standard was selected as the data exchange format between the developed modules and external data sources. The modularity of the framework allows any algorithm to be replaced by any other algorithm of choice, such as existing software within railway companies or other

models for specific needs. The approach is applied to a case study from the Dutch railways showing an overall improved timetable performance and thus demonstrating the feasibility of this approach. However, we emphasize that the framework and implemented models are generic and can be applied to any railway from any country. For instance, we apply blocking time theory for conflict detection and infrastructure occupation, which is also supported by the International Union of Railways (UIC, 2013). The blocking time modelling represents a generic building block while the exact computation of its components (mainly the approach time) depends on the specific signalling logic (Hansen & Pachl, 2014).

The main original contributions of this paper can be summarized as

1. A proposal for performance-based railway timetabling with integrated timetable construction and evaluation,
2. Description of an integrated three-level modular framework incorporating six main performance indicators in the timetabling process,
3. A proof-of-concept by an implementation of algorithms in a consistent architecture with standardized data exchange formats, and
4. Demonstration of the approach on a real-life non-trivial dense railway network.

Section 3.2 presents the timetable performance indicators that should be taken into account to reach a high timetabling design level. Section 3.3 then presents the performance-based timetabling framework with successively the functionalities of microscopic timetabling, macroscopic timetabling, corridor fine-tuning, and their interactions. Section 3.4 illustrates the approach to a case study of the Dutch railway network, and finally, Section 3.5 ends with conclusions and recommendations.

3.2 Timetable performance

The quality of a railway timetable can be measured by several Key Performance Indicators (KPIs). Traditional KPIs are the operational speeds or scheduled running times on train lines, and more general scheduled journey times in networks including transfer times where train lines meet. On the other hand, the main KPIs of railway operations are punctuality and reliability. Short journey times in the timetable do not necessarily imply good punctuality or transfer reliability, but on the contrary they may lead to large waiting and realized travel times when connections are missed or trains cancelled. Therefore, the timetable must also be robust to normal variations of running and dwell times so that punctual and reliable operations can be realized.

Furthermore, structural route conflicts between trains due to too tight scheduling must be avoided to prevent unnecessary braking and waiting of trains with negative consequences for safety, punctuality and energy consumption. The latter point is typical

for railways which are characterized by trains competing for the same infrastructure. Track capacity allocation is therefore an integrated part of railway timetable design. At this level the timetable is also known as the traffic plan, which contains the exact routes of all trains and the orders of trains over conflicting routes. Also the safety and signalling constraints must be incorporated to prove that the traffic plan is conflict free and the infrastructure capacity consumption allows normal deviations from train paths. The above concepts are captured in several performance indicators as follows (Goverde & Hansen, 2013):

- **Journey time efficiency**: The time scheduled between any origin and destination including running times, dwell times and transfer times.
- **Infrastructure occupation**: The share of time required to operate trains on a given railway infrastructure according to a given timetable pattern.
- **Timetable feasibility**: The ability of all trains to adhere to their scheduled train paths. A timetable is feasible if (i) the individual processes are realizable within their scheduled process times, and (ii) the scheduled train paths are conflict free, i.e., all trains can proceed undisturbed by other traffic.
- **Timetable stability**: The ability of a timetable to absorb initial and primary delays so that delayed trains return to their scheduled train paths without rescheduling.
- **Timetable robustness**: The ability of a timetable to withstand design errors, parameter variations, and changing operational conditions.
- **Energy consumption**: The amount of energy consumed by the train traffic.

Some of these performance indicators are based on typical macroscopic quantities such as journey time efficiency, while others require a microscopic level of detail such as infrastructure occupation, timetable feasibility and energy consumption. Timetable stability refers to a minimum amount of time allowances that must be available throughout the timetable and in particular at bottlenecks, while robustness refers to how these allowances are distributed between the train paths to maintain performance when trains deviate slightly from their scheduled paths. Stability is closely related to infrastructure occupation and can be incorporated using the UIC guidelines on acceptable infrastructure occupation (UIC, 2013) at the microscopic level, while robustness represents a trade-off with short journey times and is therefore best considered at the macroscopic level together with journey time. Energy consumption is typically a secondary objective, particularly in dense railway networks, and can therefore be considered as a fine-tuning step after the time allowances have been set based on feasibility and robustness. The contribution of the present paper is to apply these indicators in an integrated timetabling framework to actually compute timetables that satisfy all the indicators and thus reach the highest timetabling level with an additional sustainability dimension of energy efficiency.

In this paper we measured the performance indicators as follows. Journey time efficiency is measured as the maximum and mean journey time increase over the minimum journey times[1] for selected origin-destination journeys over the network. Timetable feasibility is a hard constraint in the timetable design framework, which means that all resulting process times are realizable and all conflicts between train paths have been solved. For feasibility we accept tight headways between train paths (i.e., zero buffer time), while this is penalized by robustness. For evaluation we count the number of conflicts and unrealizable running times. Infrastructure occupation is measured using the UIC compression method in percentage for each (partial) corridor and station in the network. Stability is based on the measured infrastructure occupation and the UIC guidelines for stable infrastructure occupation, and is measured as satisfying the UIC guidelines for each corridor and station. It is also a hard constraint in the timetable design framework. Robustness is measured by the average settling time of the delay propagation over the entire network over a set of delay scenarios. Finally, energy consumption is measured as the percentage energy saving over all trains in the network with respect to the minimal running times.

3.3 Performance-based timetabling

3.3.1 Framework

The proposed timetabling approach tries to schedule all train path requests with sufficient time allowances for a stable and robust conflict-free timetable and satisfying the UIC infrastructure occupation norms (UIC, 2013). This is in accordance to the Network Statements issued by the Infrastructure Managers from all EU countries to allocate the infrastructure capacity to the Railway Undertakings. This might require extending critical running times on corridors with an unacceptable capacity consumption to decrease running time differences. Moreover, we compute timetables at a precision of 5 s instead of a minute, to avoid capacity waste and unrealizable process times by rounding to minutes.

The timetabling framework is performance-based in the sense that all six timetabling KPIs from Section 3.2 are explicitly taken into account to guide the timetable construction process. To make this possible an integrated approach is proposed on three levels:

- A microscopic level for highly detailed local computations;
- A macroscopic level for aggregated network optimization; and
- A fine-tuning level for corridor optimization.

Figure 3.1 illustrates this three-level timetabling approach. The input data are standardized railML files (Bosschaart, Quaglietta, Janssen, & Goverde, 2015). The microscopic

[1] A minimum journey time is the minimum technical time based on a given line plan.

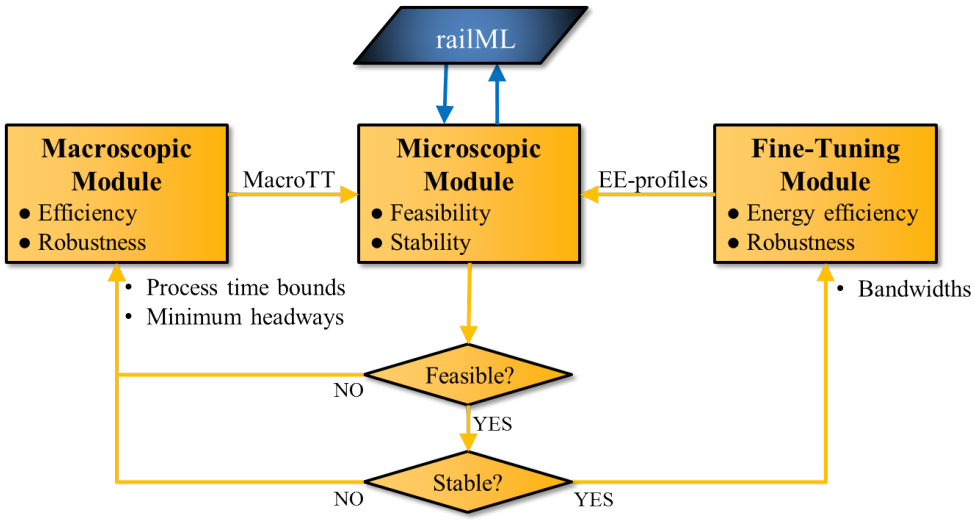

Figure 3.1: Three-level performance-based timetabling framework

model computes detailed running and blocking times, and aggregates the results into a macroscopic model that contains only the main macroscopic stations characterized by train interactions such as overtaking, connections, and merging or crossing railway lines that need decisions at the macroscopic level such as synchronization and train sequence orders. The macroscopic model then computes a network timetable taking into account network constraints and trying to avoid cancelled train path requests. The macroscopic timetable is transformed back to the microscopic model that fills in the details on microscopic level. These two models work iteratively where the microscopic model is used for conflict detection, infrastructure occupation and stability given the (completed) macroscopic timetable, while the macroscopic model optimizes a trade-off between journey time efficiency and robustness given the constraints set by the microscopic model. Infrastructure occupation is based on the UIC timetable compression method (UIC, 2013) which also provides norms for acceptable stability. The macroscopic model is an Integer Linear Programming (ILP) model and includes a simulation model to find the most robust timetable out of several hundred feasible solutions. The overall cost function contains several terms including a robustness cost derived from the simulations. These micro-macro iterations converge to a timetable that is conflict-free, stable and robust (Bešinović, Goverde, Quaglietta, & Roberti, 2016).

The final allowance times (over a required minimum allowance time) and buffer times in the timetable are the result of the interaction between the microscopic and macroscopic models. The microscopic level takes care that sufficient buffer time is available on the corridors and stations by computing the infrastructure occupation using the timetable compression method, while the macroscopic model optimally allocates it over the timetable. If the train density is very high then the optimization will try to find the optimal train sequence orders and overtaking locations, and homogenize the trains by e.g. extending the running time of the intercity trains until the capacity consumption is satisfied. Of course, the number of trains must be realistic otherwise

the macroscopic algorithm will have to cancel train path requests to build a feasible timetable. This is clearly penalized in the objective function, but it can be applied as a last operation. Since our aim is to design robust conflict-free timetables, we assume that the capacity balance of the number of trains, heterogeneity, average speed, and stability can be satisfied. If this is not the case then the timetable planner must adjust the input data based on the feedback given by the optimization (e.g., cancelled train path requests, homogenized trains, and high infrastructure occupation at bottlenecks). Hence, our framework also works in highly congested networks.

Along with the developed microscopic, macroscopic and fine-tuning models it is highly important to utilize the data consistency between the models. In particular, the iterative data flow has to be maintained between the micro and macro models on one side and between the micro and fine-tunning models on the other. To this purpose, we developed two functions to transform data to a desired model. First, we perform a micro-to-macro network aggregation, which determines stops and other timetable points that will be the nodes (macroscopic stations) in the macro network connected by arcs with the corresponding aggregated running times. A similar network aggregation was deployed in Schlechte et al. (2011). Second, a new reverse macro-to-micro transformation computes operational running times for computed macroscopic arrival and departure times from the macroscopic timetable (MacroTT), where the allocated time supplements are exploited in a feasible speed profile for each train run. By doing this, we enable further microscopic calculations such as conflict detection and capacity assessment.

The third level optimizes the speed profiles of all trains on each corridor between main stations while maintaining the scheduled event times at the corridor ends. The micro and fine-tuning models use the same level of infrastructure detail, so no network transformations are needed here. The microscopic model mainly provides the scheduled event times to the corridor fine-tuning model which have to be respected. The fine-tuning module first computes energy-efficient speed profiles (EE speed profiles) for the given scheduled event times, after which the blocking times are updated and bandwidths are determined around the speed profiles for the local trains (micro-to-corridor transformation) for corridor improvements. The IC trains are now fixed by the computed energy-efficient train speed profiles, while the time-distance speed profiles for the local trains over the successive corridors will be optimized within the available bandwidths that maintain feasibility. The corridor fine-tuning optimization optimizes the arrival and departure times at the intermediate stops in the corridor with respect to expected delays and energy savings considering stochastic dwell times at these stops. The train bandwidths are initially determined by the earliest start and latest end of the blocking times over the given corridor. The optimization of the published departure times within the given bandwidths between the important timetable points and bottlenecks does not influence the robustness of the network which was optimized in the previous level. This is the reason that only the local trains are fine-tuned within the given freedom left. If a given bandwidth results in a fine-tuned EE speed profile that is not conflict-free, a more conservative bandwidth is determined and the corridor

EE speed profile is recomputed. This process is repeated until a feasible timetable is obtained. In this way, we preserve timetable feasibility. These feasibility checks are internal to the fine-tuning module to avoid reiterations over the micro-macro levels. The final result is exported in railML timetable format extended with scheduled speed profile information that can be used by the trains for running punctual and energy efficient.

In timetable construction passenger demand can be considered explicitly or implicitly. Recently, Niu and Zhou (2013) proposed an approach for optimizing a train timetable in a highly congested urban subway line with the goal of minimizing the passenger waiting times at stations, while taking into account passengers that cannot board a train due to the limited capacity. They proposed a local search algorithm to optimize a timetable for a single station and a genetic algorithm to optimize a timetable for a subway line. Barrena, Canca, Coelho, and Laporte (2014b) proposed two mathematical formulations for a rail rapid transit single-line timetabling problem with the goal of minimizing passenger waiting times at stations, while considering a dynamic demand context. They solved the problem by an adaptive large neighborhood search meta-heuristic. The framework we propose considers passenger demand in several ways implicitly. As mentioned above, we try to schedule all train path requests, which are obtained beforehand based on the passenger demand, i.e., one of the goals is to maximize the transport volume. Furthermore, the passenger demand is taken into account by minimizing the trip times, which correspond to the sum of running and dwell times. In addition, passenger connections are optimized: we consider both the number of available connections (which is maximized) and the connection times (connection times that are "too short" or "too long" are penalized). The latter objective is similar to those considered in the literature, i.e., we try to reduce passenger waiting times at stations. All these goals related to passenger demand are considered in the macroscopic timetabling model, while the microscopic timetabling model is fundamental to check the timetable feasibility. Algorithm 1 shows a complete list of the successive steps of the performance-based timetabling approach. Each of these steps is performed by a separate exchangeable module and as such the approach is general. In the remainder of the paper we will focus on the implementations carried out within the ON-TIME project from a functional point of view.

3.3.2 Microscopic timetabling

The microscopic module considers multiple functions for computing and providing necessary input to other modules as well as evaluating a timetable at the microscopic level. These functions incorporate three KPIs: infrastructure occupation, stability and feasibility. As already mentioned in Section 3.3.1, the module first computes the speed profiles and running times. Next, the blocking times are determined which are the necessary input for conflict detection and infrastructure occupation, as well as for deriving minimum local headway times for the macroscopic module.

The microscopic network used within the microscopic timetabling allows high de-

Algorithm 1 Performance-based railway timetabling

```
Input: railML infrastructure, rolling stock, interlocking, timetable
Result: railML timetable with traffic plan at track section level
Build microscopic network topology
Compute time-optimal speed profile and minimum running times
Build macroscopic network topology
Compute nominal running times by adding minimum running time supplements
Compute operational speed profiles based on nominal running times
Compute blocking times
Conflicts ← 1; Stable ← 0
repeat until Stable
    while Conflicts do
        Compute minimum local headways
        Compute macroscopic network by aggregating running times
        and local headways
        Compute macroscopic timetable using network timetable optimization
        Recompute operational speed profiles based on the macroscopic timetable
        Compute microscopic running and blocking times
        Conflict detection
    end while
    Compute capacity consumption
    if an unstable corridor exists then
        for each unstable corridor do
            Relax nominal and maximum running times
            Conflicts ← 1
        else Stable ← 1
    end if
end repeat
Compute energy-efficient speed profiles
Compute bandwidths for local trains
Corridor timetable optimization of local trains
Return Timetable railML
```

tailed computations with accurate output. Arcs represent homogeneous behavioural sections defined by a constant characteristic of speed limit, gradient, and curvature, while the nodes present various infrastructure elements like signals, stopping points, and borders of track sections and switches. Additionally, procedures were developed for network and data transformations from the microscopic to macroscopic level, and vice versa. For details of the building blocks of the microscopic module, see Bešinović et al. (2017); Bešinović et al. (2016); Bešinović, Quaglietta, and Goverde (2014). In the remainder of this section we consider successively the main microscopic functionalities: speed and running time calculations, conflict detection, and infrastructure occupation and stability.

Speed and running time calculations

At the basis of a good timetable are well-defined running times. In particular, the scheduled running time consists of a minimum running time and an additional running time supplement. A good understanding of these two components is essential for the design of conflict-free, robust and energy-efficient timetables.

The minimum running time is the time required for driving a train from one point to another assuming conflict-free driving as fast as possible. Additionally, the corresponding speed profile represents a detailed train speed profile. The computation algorithms for speed profiles and running times have to be as detailed as possible in order to provide the high accuracy requirements. Running times are computed from microscopic train dynamics that require detailed rolling stock and infrastructure data, including route-specific static speed and height profiles. The corresponding Newtons motion equations are solved by numerical ordinary differential equation solvers (Hansen & Pachl, 2014).

In regular daily operations, trains are affected by stochastic variations of running and dwell times due to e.g., varying train compositions, driver behaviour, passenger volumes and weather conditions. Therefore, allowance times are added to the minimum process times so that they are robust to normal variations of the process times. These allowances must satisfy certain timetable design norms, consisting of a mix of relative and absolute values for the nominal process times (minimum process time plus minimum allowance). Running time supplements are given in percentage of minimum running time, in some countries depending on train category, while nominal dwell times are specified depending on rolling stock type and station, and nominal transfer times are provided depending on station and platform distances. The resulting nominal process times are input to the macroscopic timetable optimization as lower bounds to the scheduled process times. In the optimization the nominal times can be increased further depending on the network constraints and objective functions, resulting in the scheduled running times. The objective function of the macroscopic optimization must prevent excessive journey times by stretches of all running and dwell times. In addition an overall upper bound can be provided to the roundtrip time of trains.

Hence, in the first iteration, the minimum running times are enriched with the minimum time supplements and as such represent the nominal running times that are used in the macroscopic model. Additional to the running time, the operational speed profile defines the associated train speed profile. The operational speed profile can be obtained by exploiting the available time supplements in two ways: a) cruising at speeds below the speed limits, or b) computing energy-efficient speed profiles with optimal cruising speeds and coasting. During the timetable construction with several micro-macro iterations the reduced speeds are applied as these are much faster to compute than the optimal speed profiles. In the fine-tuning these speed profiles are replaced by the energy-efficient ones.

In current practice mostly macroscopic timetabling models are used that, in a nutshell,

try to assign time allowances in order to satisfy a given objective function. In this way, the running time supplements are allocated without actually testing that the resulting distribution of time supplements result in acceptable train speed profiles. A big variation between two (or more) successive allocated time supplements may be problematic to reproduce a valid speed profile. Even if a speed profile is possible satisfying the given time supplements, the constructed running behaviour may be unacceptable from a practical point of view when very low cruising speeds result. For example, the German practice requires that cruising speeds may not be under 40 km/h. This may be violated in the case of a relative large running time supplement over a short section. Furthermore, it is undesirable to continuously change driver behaviour such as alternating between accelerating and decelerating with different cruising speeds. Hence, even a macroscopically feasible timetable cannot always be reconstructed at the microscopic level and consequently implemented in practice. Therefore, the operational speed profiles must be computed to test feasibility of the distribution of the time allowances.

We implemented these guidelines in the computation of operational speed profiles. The scheduled running times and corresponding operational speed profiles are computed after each macroscopic timetable computation, resulting in feasible train speed profiles, which are also essential for an accurate calculation of blocking times.

The successive blocking times per train over a corridor represent a so-called blocking time stairway. Blocking times are computed using blocking time theory (Hansen & Pachl, 2014). The blocking time of a block section depends on the block length, the train speed, and the signalling system. It consists of a setup time, sight and reaction time, the approach time to the block section over at least the braking distance, the running time in the block, the clearing time in which the train clears the block over its entire length, and the release time of the route, see Figure 3.2 for an example in three-aspect two-block signalling (Goverde et al., 2013). The blocking times are the essential input to the main microscopic algorithms such as conflict detection, capacity assessment and minimum headway computation. Recall that the blocking times are based on the nominal running times in the initial micro-macro iteration, and on the scheduled running times in all following iterations.

Conflict detection and realizability

Timetable feasibility is a key performance measure. It is important to have a feasible timetable in order to provide uninterrupted train runs, i.e., without unnecessary braking and re-acceleration. This timetable KPI is beneficial from several perspectives: 1) it improves safety by preventing unnecessary red signal approaches; 2) it gives less workload to drivers; 3) it provides a more comfortable ride to passengers; and 4) it saves energy. Therefore, each time a macroscopic timetable has been computed, the microscopic module automatically checks the timetable on microscopic feasibility.

The feasibility of the timetable is tested twofold: a) a realizability check of scheduled event times; and b) conflict detection. The former is simply tested by checking

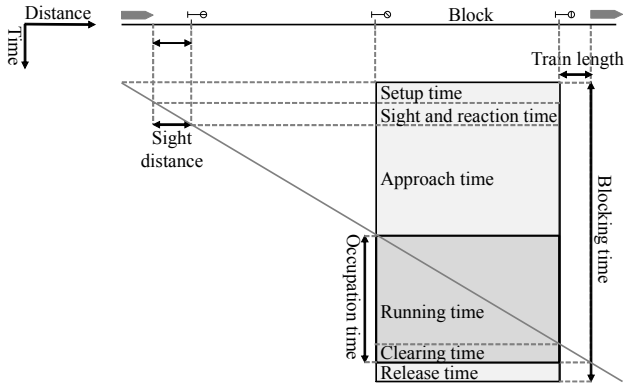

Figure 3.2: Blocking time of a running train

whether the scheduled running and dwell times exceed the minimum values. Note that the macroscopic timetabling model always provides realizable aggregated scheduled process times, so this realizability check is mainly focused on the event times at the smaller stations and other microscopic timetable points after transforming the macroscopic timetable onto the microscopic network. Unrealizable process times are mainly caused by rounding down, which becomes problematic specifically when scheduled event times must be given in minutes. In our approach, the macroscopic model computes timetables with a precision of 5 s, while we allow a precision of 1 s in the microscopic model so that rounding is not an issue anymore.

The conflict detection model determines if the scheduled trains can run undisturbed. For this blocking times are used on the basis of the operational speed profiles. Conflicts are indicated by an overlap of the blocking times of two successive trains. The second train then approaches the block section that is still blocked by the preceding train and therefore must brake in response to the signalling logic. These track conflicts are solved by shifting trains in time until their blocking times do not overlap anymore. After all track conflicts have been detected, the corresponding minimum headways are recomputed. These new headways are given back to the macroscopic timetabling model to iteratively adjust the macroscopic timetable until all track conflicts are resolved.

Capacity consumption and stability

Capacity consumption is defined as the time share needed to operate trains on a given infrastructure according to a given timetable pattern taking into account scheduled running and dwell times. As such, it directly determines the stability of the timetable. The same as for conflict detection, we use the computed blocking times to evaluate the capacity consumption.

A timetable is called *stable* if a certain initial and primary train delay can be absorbed by the time allowances in the timetable without active dispatching (Goverde, 2010). Therefore, the larger the time supplements and buffer times the better is the ability of the timetable to prevent propagation of delays, i.e., the timetable is more stable. If the total amount of buffer time in a corridor is higher than the amount recommended

by the UIC code 406, the timetable is considered sufficiently stable. Otherwise it is defined unstable and the macroscopic timetable has to be recomputed to reduce the infrastructure occupation on the critical corridors or stations and thereby releasing buffer times.

The recommended UIC stability norms are given in Table 3.1. The values presented here are for a given corridor for the peak period or the whole day. Norms for station areas still require more research as elaborated in UIC (2013).

Table 3.1: Recommended UIC infrastructure occupation for corridors

Type of line	Peak period	Daily period
Dedicated suburban passenger traffic	85%	70%
Dedicated high-speed	75%	60%
Mixed traffic	75%	60%

The capacity assessment model developed is based on max-plus automata (Gaubert & Mairesse, 1999) and explained in Bešinović et al. (2017). The model is applicable to both corridors and stations. For now, we assume the given UIC norms from Table 3.1 for both corridors and stations. If the computed infrastructure occupation is not satisfactory for a corridor then we relax the running time supplements of the trains in the corridor to allow more homogenized traffic through the considered corridor by reducing running time differences. For instance, a fast train line may be allowed a higher time supplements over such a corridor. This relaxation is explained in Bešinović et al. (2016). The relaxed constraints are input to a new iteration of the macroscopic timetable optimization.

3.3.3 Macroscopic timetabling

The macroscopic timetabling considers the railway network at an abstract level, neglecting many details of the real-world network. In particular, only network points like stations and junctions, where trains overtake, merge, cross or connect, as well as the lines connecting them are represented at the macroscopic level. The motivation of applying this network reduction is that it is computationally faster to work with a simplified network, and therefore several potential timetables can be evaluated according to the different key performance indicators, including robustness. Clearly, once the 'best' macroscopic timetable has been determined, its feasibility at a microscopic level is checked, as described in Section 3.3.1. Macroscopic timetabling is used to determine the best feasible schedule of trains in the macroscopic network by considering a trade-off between timetable *efficiency* (e.g., scheduling as many trains as possible on the network and obtaining the shortest journey time for the passengers between origins and destinations, by scheduling efficient passengers connections) and *robustness*.

Several macroscopic timetabling approaches have been presented in the literature. In the proposed three-level framework, we focus on macroscopic optimization models, as they are suitable to achieve the performance measures described in Section 3.2.

Indeed, these indicators can be seen as the objectives to be reached through the optimization models. The main objectives traditionally considered in the optimization models can be summarized as providing the maximum *efficiency* of the railway system and avoiding delay propagation as much as possible. These goals lead to a common classification of the optimization models in *nominal timetabling models* and *robust timetabling models* (Cacchiani & Toth, 2012).

A way for modelling nominal timetabling is introduced in Serafini and Ukovich (1989) for periodic timetabling, in which one needs to determine a schedule of trains that is repeated every given time period (e.g. every hour). This problem is called the Periodic Event Scheduling Problem (PESP). In PESP, an event represents a periodic arrival or departure of a train at a station. Such events are scheduled for one basic period (say, 1 hour), which is then repeated throughout the day. Let N be the set of all events to be scheduled and T the considered cycle time. Then, the model uses integer variables $v_i \in \{0,\dots,T-1\}$ representing the time instant at which event $i \in N$ takes place. In order to satisfy safety and track capacity requirements, a set of periodic constraints is needed. In particular, for any pair of trains using the same track, a minimum headway time must be respected, and overtaking along a track linking two consecutive stations, as well as crossing of trains traveling in opposite directions along the same track, must be forbidden. In addition, for each train, minimum running times and dwell times must be respected. Each periodic constraint deals with a pair of events $i, j \in N$ (i.e., arrivals or departures of trains) and a periodic time window that imposes a lower bound l_{ij} and an upper bound u_{ij} on the time interval between the two events modulo the cycle time, $l_{ij} \leq v_j - v_i \pmod{T} \leq u_{ij}$. These constraints can be linearized by introducing a binary variable p_{ij} for each pair of events $i, j \in N$ so that the periodic constraint can be formulated as

$$l_{ij} \leq v_j - v_i + p_{ij}T \leq u_{ij}, \tag{3.1}$$

where $p_{ij} = 1$ if $v_j < v_i$, and 0 otherwise. These constraints can also be used to model connections between trains.

A different way for modelling nominal timetabling is to formulate it as a *job-shop scheduling problem*. A train trip is described by a set of stations which a train must serve or pass through. Therefore, a trip can be viewed as a job, i.e., a set of tasks to be performed (Oliveira & Smith, 2000; Szpigel, 1973). These jobs are scheduled on tracks regarded as resources (or machines) in such a way that only one train can occupy a track segment at a time, while several trains can be at a station at a time as long as its capacity is respected. These types of models turn out to be very effective for real-time rescheduling (D'Ariano, Pacciarelli, & Pranzo, 2007).

An alternative effective representation, frequently used for the nominal non-periodic timetabling, is to expand the graph representing the railway network, through the entire time horizon, usually of one day, discretized in time units (Cacchiani et al., 2010; Caprara, Fischetti, & Toth, 2002). In this case, the problem is modelled by means of a *time-space graph* $G = (V,A)$. The node set V is defined by the union of the sets of nodes, called, respectively, *departure* and *arrival nodes*, representing the time instants

at which some train can arrive at and depart from a station. The arc set A is partitioned into arc subsets $A^1,\ldots,A^{|Tr|}$ for each train $t \in Tr$, where Tr represents the set of trains to be scheduled. In particular, the arc subset A^t for a train t contains a set of *starting arcs* corresponding to the feasible departures of train t from its first station; a set of *station arcs* representing the feasible dwellings of train t at each visited station; a set of *segment arcs* representing the feasible runs of train t from each visited station to the following; and a set of *ending arcs* corresponding to the feasible arrivals of train t at its last station. With the described time-space graph representation, a *time-distance path* in the graph corresponds to a timetable for a train. Given this graph representation of the timetabling problem, two modelling options exist. The first one is to use binary arc variables $x_a, a \in A^t, t \in Tr$, where x_a is equal to 1 if, and only if, arc a is selected in an optimal solution (Borndörfer & Schlechte, 2007; Cacchiani et al., 2010; Caprara et al., 2002). This formulation is a *multi-commodity flow* formulation, in which the commodity, i.e., the train, index is hidden in the multi-graph definition. The second model is the *path formulation* (Borndörfer & Schlechte, 2007; Cacchiani, Caprara, & Toth, 2008). It considers the sets $P^t, t \in Tr$ of feasible paths for each train $t \in Tr$, i.e., paths that respect the requirements on running and dwell times. The path formulation has binary path variables $x_p, p \in P^t, t \in Tr$, indicating if path p is chosen ($x_p = 1$) or not ($x_p = 0$) in the solution. The model contains exponentially many variables and can be solved by a branch-and-price approach or by using heuristic algorithms. The multi-commodity flow and the path formulations are characterized by constraints that forbid the simultaneous selection of incompatible arcs or paths, respectively, due to e.g., minimum headway time violation, overtaking or crossing of trains. Indeed, in these models, the nominal running and dwell times for the trains are imposed directly in the definition of the time-space graph. The incompatibility constraints are formulated in the form of packing or clique constraints:

$$\sum_{a \in C} x_a \leq 1, \quad C \in \mathcal{C}, \tag{3.2}$$

where $\mathcal{C}$ represents the (exponentially large) family of maximal subsets C of pairwise incompatible arcs. Similar constraints are imposed in the path formulation. Additional constraints are used to impose that at most one arc, associated with a given train $t \in Tr$, is selected among all the arcs in the set $\delta_t^+(\sigma)$, representing the arcs starting from source node σ, so that at most one timetable is selected for each train:

$$\sum_{a \in \delta_t^+(\sigma)} x_a \leq 1, \quad t \in Tr. \tag{3.3}$$

Similar constraints are imposed in the path formulation to ensure that at most one path is selected for each train. Additional constraints can also be imposed to model the connections between trains, taking into account the nominal connection times.

Each one of the described optimization models for the nominal timetabling problem can be embedded in the proposed three-level framework. These models aim at de-

termining an efficient train schedule, i.e., they take into account the nominal running times, dwell times and connection times computed in the microscopic model, as described in Section 3.3.2. However, as explained in Section 3.2, the quality of a timetable is evaluated also according to its robustness against delays. Robust timetables can be achieved through different methods, such as Stochastic Programming (Kroon, Maróti, Retel Helmrich, Vromans, & Dekker, 2008), Light Robustness (Fischetti & Monaci, 2009), Recoverable Robustness (Liebchen, Lübbecke, Möhring, & Stiller, 2009) or Lagrangian Robustness (Cacchiani, Caprara, & Fischetti, 2012). These methods are based on modifications of the optimization models for nominal timetabling, so as to include *empty time slots* between trains that help to reduce delay propagation. Even though these methods can effectively compute robust timetables, they generally require longer computing times than those for solving the nominal problem.

To limit the computing time, instead of embedding an optimization model for robust timetabling in the three-level framework, we split the macroscopic timetable computation into two components. The first component represents *macroscopic timetable optimization*, which is implemented by an ILP model for nominal timetabling. In particular, we use the path formulation with a time horizon of one hour and a time discretization of 5 seconds. The ILP model is solved iteratively in a heuristic way, and its output provides a set of efficient macroscopic timetables (see Section 3.3.3). The second component aims at evaluating the robustness of these efficient timetables by means of a fast local search algorithm (see Section 3.3.3). The output of this component is a robust and efficient macroscopic timetable that will then be checked at a microscopic level by the microscopic module. Providing a set of timetables to the second component is a key point of our framework, as it increases the chances of deriving a good quality robust timetable. This also motivates our choice of solving the nominal timetabling through a heuristic algorithm.

In the following, we outline the objectives and constraints that are included in the macroscopic ILP model, and present a *delay propagation algorithm* that is used to achieve timetable robustness. We refer the reader to Bešinović et al. (2016) for further details on these methods.

Macroscopic timetable optimization

The macroscopic timetable optimization adopts a time expanded graph, built upon the macroscopic network, and is based on the path formulation described above. For a given train and its *route* (i.e., the sequence of macroscopic stations that the train serves or passes), its macroscopic feasible timetable corresponds to a *feasible time-distance path* in the time expanded graph that visits all the stations on the route while respecting the given maximum *journey time* from its origin station to its destination station. We associate a cost to each time-distance path, which represents the quality of the corresponding timetable for the train, without taking into account the interaction with other trains. In particular, the path cost takes into account the running and dwell times exceeding the nominal ones. The ILP model contains a binary variable for each feasible time-distance path of any train, which specifies whether the path is selected

as the timetable of the train in the solution or not. Therefore, this model contains exponentially many variables. Instead of solving it to optimality by a branch-and-price approach, which would require long computing times, we apply a *randomized multi-start greedy heuristic* (Bešinović et al., 2016). The macroscopic timetable optimization takes into account several performance measures, included as a *weighted multi-objective function*, in which different penalties are associated with the different objectives. Depending on the penalty values, one objective can have priority over another one, or the goal can be to find a trade-off between the different objectives. The multi-objective function contains the following terms, each one weighted by a penalty which are parameters of the optimization model:

- Minimization of path costs, i.e., minimization of running and dwell times,
- Minimization of missed connections,
- Minimization of time exceeding the nominal connection times,
- Minimization of cancelled train path requests.

The first three terms clearly relate to the performance measure of journey time efficiency described in Section 3.2, while the latter maximizes transport volume. Connection times cannot be included directly in the path cost, since they refer to pairs of trains and not to single trains. However, *timetable connectivity*, i.e., the connection between pairs of trains for passenger transfers or rolling stock connections, is also taken into account as one of the objectives of the macroscopic model. In particular, we minimize the number of *missed connections*, as well as the time exceeding the *nominal connection time*. We consider a connection as missed if at least one of the two connecting trains is cancelled. If both trains are scheduled we compute the difference between the actual connection time and the nominal one.

Another main goal of the macroscopic optimization component is to maximize the *transport volume*, i.e., the passenger or cargo-tonne delivered: this is achieved in our model as the minimization of cancelled train path requests. Timetable *feasibility* at a macroscopic level is achieved by defining, for each train, feasible time-distance paths in the graph, and by imposing that at most one path, among a set of conflicting paths, can be part of the solution.

The proposed model can deal both with cyclic and non-cyclic timetabling. In the former, we are given routes for *train lines* rather than individual trains, as all trains belonging to the same line must visit the same sequence of stations. Similarly, we are given the journey time of each line and in addition the *periodicity* of the trains of the line. In order to satisfy the periodicity constraint, we impose that either all trains of the line are scheduled or all of them are cancelled. Clearly, the penalty for train cancellation is very high and therefore it is very unlikely that a complete train line will be cancelled. Different planning time horizons are to be considered for cyclic or non-cyclic timetabling. In our case study we focus on the cyclic case (see Section 3.4).

The ILP model is solved in a heuristic way by iteratively executing a randomized multi-start greedy heuristic. At each iteration, the trains are scheduled one at a time according to a given order. Scheduling a train corresponds to selecting one of the feasible time-distance paths for the train. A dynamic programming procedure, which takes into account all the trains already scheduled in the current iteration, computes a feasible timetable for the current train, and takes into account all the described objectives by assigning penalties to the unpromising nodes of the graph associated with the train, so that the best path will visit the nodes with the smallest possible penalties. In the case of periodic timetabling, at each iteration of the algorithm we select a feasible time-distance path for the entire line, i.e., we select simultaneously one path for each train of the line ans thus ensuring that the periodicity constraint is respected.

Once the algorithm has been executed for a given number of iterations, several macroscopic feasible timetables are available and are then evaluated to assess their robustness quality. As explained in Section 3.3.2, robustness is incorporated at a microscopic level by inserting time allowances. In the next paragraph, we explain how we consider robustness also at a macroscopic level.

Robustness evaluation

A delay propagation algorithm is used to take into account the stochasticity of the events, such as train delays, that can occur during operations. The goal of this algorithm is to evaluate the robustness quality of each feasible timetable determined by the randomized multi-start greedy heuristic and to select as best timetable the one having the smallest *robust cost*. The latter is given by the cost of the timetable according to the multi-objective function plus the cost of the timetable according to the delay propagation algorithm.

A set of delay scenarios (1000 in our computational experiments) is randomly generated. For each delay scenario the effect on each timetable is evaluated by applying a local search algorithm that tries to resolve the potential conflicts caused by the generated delays by retiming the trains (Bešinović et al., 2016). The algorithm computes the overall delay propagation or establishes that some conflicts cannot be resolved. Accordingly, a cost is assigned to each timetable: it takes into account the effect of all the delay scenarios on the timetable, and is defined as the average settling time (i.e., the time required until all delays have been absorbed by the time allowances in the timetable) over all the delay scenarios. The timetable with the smallest robust cost is then selected as the best macroscopic timetable and this is the outcome of the macroscopic timetabling.

3.3.4 Corridor fine-tuning

Energy efficiency becomes more and more important within the railway system. Currently several approaches exist for energy-efficient driving and energy-optimal conflict resolution within real-time traffic management and optimization (Hansen & Pachl, 2014). Although energy efficiency is an important concern to railway infrastructure

managers and railway undertakings, only little literature focuses on energy-efficient timetabling. However, the timetable is the static basis for real-time operation. On the one hand, the static timetable has to enable real-time operational control measures, which means that allowance times are available to provide flexibility for traffic management. On the other hand, when real-time optimization methods are applied such as energy-efficient driving, the possible real-time speed profiles have to be considered already within the timetabling process in order to avoid conflicts due to the real driving behaviour. Therefore, within the ON-TIME timetabling approach energy-efficient speed profiles are already considered in the timetabling process.

Energy-efficiency within the entire planning process

Most of the scientific literature focuses on energy efficient driving strategies, where the real-time train speed profile is optimized. However, the potential of energy-efficient driving is directly connected to the given timetable (Scheepmaker & Goverde, 2015). Still, energy consumption is typically considered as a secondary goal in the existing timetabling models, with feasibility, efficiency and robustness being the primary goals. We also consider this hierarchical approach within our tree-level timetabling framework. Before presenting our corridor fine-tuning approach, we first describe different timetabling approaches that consider energy consumption and classify them according to the timetabling levels.

Line planning was not considered within the ON-TIME project. Although line planning affects the energy consumption and the operational costs, the energy consumption is only scarcely regarded in methods for line planning. The number and position of the stops, the train lines and the train frequency are planning decisions for the railway undertakings and the public authorities. Oettich, Albrecht, and Scholz (2004) presents an approach for the optimal train frequency planning in a suburban network considering feedback of the quality of the offer on demand. It turns out that vehicle size, headway and demand are closely coupled and frequent small vehicles lead to less energy consumption per passenger. Gassel and Albrecht (2009) present the impact of request stops on railway operation including the energy consumption.

The process of creating a conflict-free timetable deals with the optimal allocation of the infrastructure capacity. Within the ON-TIME project this is considered at the macroscopic and microscopic optimization level. However, approaches can be found which consider energy as relevant criteria within this timetabling level. Kraay, Harker, and Chen (1991) already dealt with the optimal pacing of freight trains on a single-track line under the consideration of energy consumption.

Adjustment and optimization of running times is regarded in several publications, where energy consumption is part of the operating costs. Ghoseiri, Szidarovszky, and Asgharpour (2004) consider energy consumption as a measure of railway undertaking satisfaction and total passenger journey time as a measure of passenger satisfaction.

In the timetabling process a trade-off between these conflicting objectives has to be made. T. Albrecht (2005) discusses a similar dependency between planned and given

running time and energy consumption. He minimized the traction energy consumption of a train during the whole journey by finding the optimal allocation of running time allowance for a suburban railway line among different sections. This algorithm is based on a predefined total amount of running time allowance between fixed target points and energy-efficient driving between two target points according to a given running time. Sicre, Cucala, and Fernandez (2010) presented a similar approach to calculate the optimal allocation of running time allowances among the different sections in order to generate an optimal schedule. The total available running time supplement for the whole service is an input parameter from macroscopic timetabling, as well. Cucala, Fernandez, and Sicre (2012) also consider finding the optimal allocation of the running time allowance. Hence, in recent approaches the allocation of running time under consideration of energy efficiency is based on a predefined macroscopic timetable with fixed required passing, arrival and departure times at important timetable points as energy efficiency is always a conflicting criteria against minimizing the journey times and enlarging capacity consumption. Within the ON-TIME project infrastructure occupation is one of the major optimization criterion and is therefore considered on a higher level. Energy consumption is consequently a secondary optimization criteria in order to ensure the required capacity, robustness and stability needs. This approach ensures that in dense networks minimizing the energy consumption does not reduce timetable stability.

On the other hand, although detailed speed profile planning is typically not part of the timetabling process the consequences of different driving speed profiles should be considered. Therefore, the energy-efficient driving strategies of a train journey should be regarded for the calculated scheduled running times to guarantee feasibility and stability of the final timetable.

The optimized timetable must be communicated to both the driver and the passenger. The timetable is published to the passengers with a precision of one minute in Europe. T. Albrecht (2005) already discussed the effects of rounding the arrival and departure times in a timetable. Different published departure times can increase or decrease the energy consumption along a line. Especially when the dwell time is shorter than the planned dwell time, additional energy savings are possible by using the actual allowance times.

On the one hand, the published times are important for passenger arrivals and delay calculations in case of long dwell times. On the other hand, these published times are restrictive because early departures are not allowed in case of small dwell times. During timetabling the dwell time allowances could be exchanged with running time allowances where they could be applied for e.g. energy-efficient driving in case of short dwell times.

Energy-efficient speed profiles

The energy-efficient speed profiles are computed with respect to the microscopic infrastructure and rolling stock characteristics for the scheduled running times (includ-

ing running time supplements) obtained from the micro-macro timetabling iterations. The optimal driving speed profiles are determined according to the theory of energy-efficient driving (Howlett & Pudney, 1995). This speed profile is typically characterized by different optimal driving regimes and the switching points between the regimes: acceleration with maximum acceleration power, cruising at an optimal cruising speed, coasting without any tractive effort, and braking with maximum (service) braking effort. Figure 3.3 shows a simplified illustration of the application of different driving regimes between two stops on a simple section with constant gradient and speed limit (T. Albrecht, 2014). The determination of the optimal driving regimes and the switching points is a well-discussed optimization problem and can be done using different optimization methods.

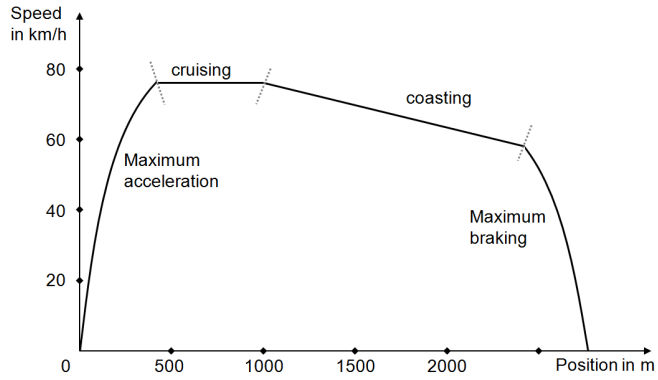

Figure 3.3: Energy-optimal driving regimes (T. Albrecht, 2014)

T. Albrecht (2005) already developed and applied an approach for finding the optimal regimes and regime durations. He used the theory of energy-efficient train control to implement the algorithm on a driver advisory system for energy-efficient train operation in real-time. The algorithms can be used off-line within the timetable planning process as well in order to simulate energy-efficient train movements when drivers are familiar with the theory of energy-efficient train operation or are supported by a driver advisory system. Therefore, the energy-efficient speed profiles should be used in the planning process, as well.

The calculated speed profiles are used to recompute the blocking times within the three-level timetable design framework so that at the microscopic level the speed profiles can be checked on conflicts within the timetable. In addition, the information on the switching points and regimes can be used to guide optimal energy-efficient driving in case of punctual train operation even if no dynamic driver advisory systems are used. If driver advisory systems are used they essentially give dynamic speed advice with respect to delays and follow the scheduled energy-efficient speed profile otherwise. The output of the timetabling framework is a Timetable railML file with scheduled train paths at (track-free detection) section level, extended with scheduled energy-efficient speed profiles, consistent with the ON-TIME real-time railway traffic management framework (Quaglietta et al., 2016).

Corridor optimization

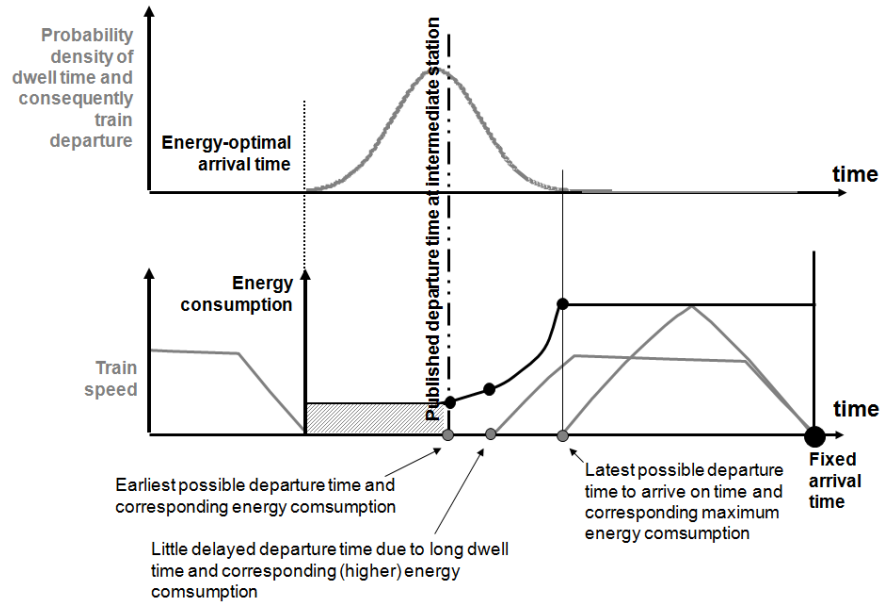

Figure 3.4: Dependency between the dwell time distribution, departure of the train, and corresponding energy consumption

Figure 3.4 explains the dependency between the dwell time distribution, the departure of the train, and the corresponding energy consumption. The figure shows that a short planned dwell time leads to a possible punctual departure of the train and less energy consumption because the allowance time could be used for additional running time. If the dwell time is slightly higher this leads to a little delayed departure and a higher energy consumption on the following section. In contrast to this, a pessimistic published departure time (for a higher scheduled dwell time) might use the little delayed departure time as published departure time. In this case, the probability of a delayed departure is less, but the probability of waiting for the departure time is higher, because there is a high probability that the dwell time might be shorter. Therefore, the minimal achievable energy consumption is higher than when publishing an earlier departure time. This means that dwell times should not be considered as deterministic in the timetabling process but as dwell time distributions within the process of finding the published departure times. The dwell time distributions must correspond to the realized dwell times and must be obtained using operational data. Because the local trains at intermediate stations are influenced mostly by the stochastic dwell time process rather than other conflicts or restrictions, these trains should be considered to enlarge the robustness of the timetable.

The target of the corridor optimization is to determine the published arrival and departure times at intermediate stops within given bandwidths, under consideration of the stochastic dwell times and energy-efficient driving in case of short dwell times. The mathematical approach is another two-stage optimization process (Binder & Albrecht, 2013).

This final step in the timetabling process is the corridor fine-tuning for local trains between the defined macroscopic timetable points. Note that the event times at these macroscopic timetable points were optimized in the macroscopic timetable optimization. For intercity trains all served stations are important points and the energy-efficient speed profiles are already determined in the previous step. For local trains however the arrival and departure times at intermediate stops on the corridors were not yet optimized and they offer flexibility for optimization within given time windows, see Figure 3.5. Therefore, the departure and arrival times at the beginning or end of the corridors are fixed at the major stations, and this in fact defines the corridors in which the timetables of the local trains are optimized.

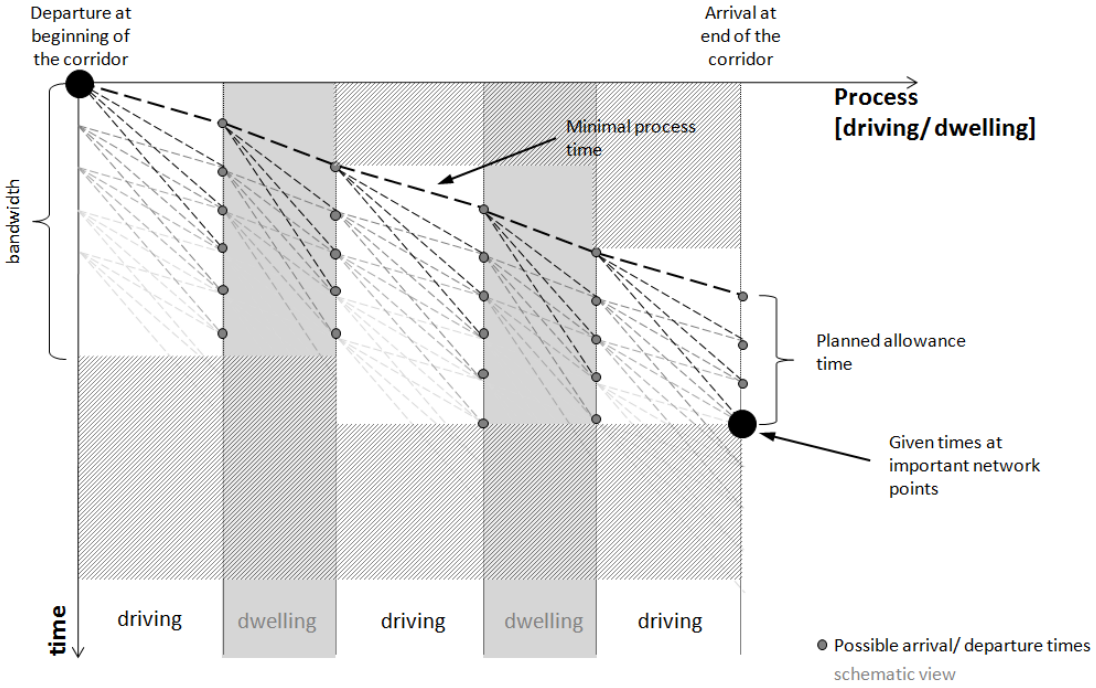

Figure 3.5: Flexibility of the corridor optimization

The bandwidths at the intermediate stations are determined from the blocking times of the trains preceding and following the local train that has to be optimized (gray shaded in Figure 3.5). Hence, the speed profiles of the adjacent trains are important in order to maintain a conflict-free timetable. The total amount of running time supplement over the corridor and the bandwidth between the macroscopic timetable points are provided by the macroscopic and microscopic timetable levels, respectively.

The optimization algorithm is based on the running time optimization along a line as presented in T. Albrecht (2005). However, T. Albrecht (2005) considered the dwell times as deterministic and only the departure events at the intermediate stations were considered as process stage at which the decision about the running time supplement on the following sections are made. In our new approach the dwell process is divided into two different states, an arrival and a departure stage. This is illustrated in Fig-

ure 3.5 by marked dots around the dwelling process. The departure and arrival time pairs between successive intermediate stops define the time supplement that can be used for energy-efficient driving between the two stops. The stage transition from the departure stage to an arrival stage is determined by a running time including running time supplement and the corresponding energy consumption under the assumption of energy-efficient driving. The probability of the stage transition can be regard as one, which means that the optimized running time must be realized. Furthermore, the dwelling processes need time supplements. The amount of supplement is based on the probability of a given dwell time according to the dwell time distribution function. Consequently, the dwelling process is characterized by a minimal process time and possible time allowance. The different dwell times at one station are determined by a specific probability of these dwell times. The process transition from the arrival stage to the departure stage does not consume energy. The convolution of all possible arrival times and the dwell time distribution leads to all possible departure time stages and their corresponding probability. Each stage (of arrival and departure time) is consequently characterized by a cost and a probability of reaching this stage.

In our approach, in addition to the total energy consumption the probability of delays at the target station and at the intermediate stops are considered as relevant. These have to be determined according to the possible corridor timetables since different published times at the intermediate stops may influence the departure stage (no departure before the published departure time) and of course the departure delay. In order to quantify the quality of a timetable, these three criteria are calculated at the arrival and departure stage. Hence, the first optimization target is the optimal distribution of the running time supplement according to possible timetables. This results in a multi-stage, multi-criteria decision problem which can be solved using dynamic programming (Bellman, 1957). Table 3.2 summarizes the relevant variables of the corridor optimization.

The quality criteria are calculated backwards starting at the last stage I of the optimization space which has fixed quality criteria. As the stage transition of the dwelling process is stochastic, expected values of the quality criteria are calculated for $i = I - 1$ back to $i = 1$ as follows, for the expected energy consumption

$$Q^a_{1,I}(t^a_I) = 0, \quad \forall t^a_I \tag{3.4}$$

$$Q^d_{1,i}(t^d_i) = \begin{cases} Q^d_{1,i}(t^d_{plan,i}) & t^d_i < t^d_{plan,i} \\ E_j(u_j) + Q^a_{1,i+1}(t^a_{i+1} = t^d_i + u_j + T^R_{min,j}) & t^d_i \geq t^d_{plan,i} \end{cases} \tag{3.5}$$

$$Q^a_{1,i}(t^a_i) = \int_{T^D_{min,i}}^{T^D_{max,i}} f_i(\tau) \cdot Q^d_{1,i}(t^a_i + \tau) d\tau, \tag{3.6}$$

Table 3.2: Notation of variables

Notation	Description
$T^R_{min,j}$	minimal running time on section j
$T^R_{max,j}$	maximal running time on section j
u_j	running time supplement on section j
$E_j(u_j)$	energy consumption on section j by using u_j
$T^D_{min,i}$	minimal dwell time at station i
$T^D_{max,i}$	maximal dwell time at station i
t^d_i	possible departure time at station i
t^a_i	possible arrival time at station i
Φ	possible corridor timetable (planned arrival and departure times)
Φ^*	optimal corridor timetable (planned arrival and departure times)
$t^d_{plan,i}$	planned departure time at station i
$t^a_{plan,i}$	planned arrival time at station i
$Q^a_q, q \in [1,2,3]$	quality criteria at arrival stage
	1 - expected energy consumption
	2 - expected delay at target station
	3 - expected delay at intermediate stations
$Q^d_q, q \in [1,2,3]$	quality criteria at departure stage
w_q	weighting factor of the running time optimization
v_q	weighting factor of the corridor timetable optimization

the expected delay at the target station

$$Q^a_{2,I}(t^a_i) = \max(0, t^a_I - t^a_{plan,I}), \quad \forall t^a_I \tag{3.7}$$

$$Q^d_{2,i}(t^d_i) = \begin{cases} Q^d_{2,i}(t^d_{plan,i}) & t^d_i < t^d_{plan,i} \\ Q^a_{2,i+1}(t^a_{i+1} = t^d_i + u_j + T^R_{min,j}) & t^d_i \geq t^d_{plan,i} \end{cases} \tag{3.8}$$

$$Q^a_{2,i}(t^a_i) = \int_{T^D_{min,i}}^{T^D_{max,i}} f_i(\tau) \cdot Q^d_{2,i}(t^a_i + \tau) d\tau, \tag{3.9}$$

and the expected delay at intermediate stations

$$Q^a_{3,I}(t^a_I) = 0, \quad \forall t^a_I \tag{3.10}$$

$$Q^d_{3,i}(t^d_i) = \begin{cases} Q^d_{3,i}(t^d_{plan,i}) & t^d_i < t^d_{plan,i} \\ Q^a_{3,i+1}(t^a_{i+1} = t^d_i + u_j + T^R_{min,j}) & t^d_i \geq t^d_{plan,i} \end{cases} \tag{3.11}$$

$$Q^a_{3,i}(t^a_i) = \max(0, t^a_i - t^a_{plan,i}) + \int_{T^D_{min,i}}^{T^D_{max,i}} f_i(\tau) \cdot Q^d_{3,i}(t^a_i + \tau) d\tau. \tag{3.12}$$

The decision within the dynamic programming approach is made based on the expected values of the criteria within a multi-criteria approach at each stage. Therefore the

optimal running time supplement u_j^* at the departure stage is given by

$$u_j^*(t_i^d) := \arg\min_{u_j} \tilde{Q}(u_j, t_i^d) \tag{3.13}$$

with

$$\tilde{Q}(u_j, t_i^d) = \sum_q w_q Q_{q,i}^d(u_j, t_i^d). \tag{3.14}$$

The output of the allowance allocation process gives for each possible departure and arrival time at each station the expected values of the relevant criteria for the remaining train run until the target station. Hence, the values obtained for the given departure time at the first station $Q_{q,1}^d$ give an indication about the timetable itself, because the criteria are significantly influenced by $t_{plan,i}^a$ and $t_{plan,i}^d$. In order to find the optimal corridor timetable all possible timetables Φ_k have to be analyzed which result in specific timetable quality criteria $Q_{1,1}^d(\Phi_k)$, $Q_{2,1}^d(\Phi_k)$ and $Q_{3,1}^d(\Phi_k)$. Consequently, another multi-criteria optimization problem has to be defined, in order to find the optimal corridor timetable Φ^*. Again, the weighted sum method is used to find the solution that minimizes all three criteria,

$$\Phi^* := \arg\min_{\Phi_k} \sum_q v_q Q_{q,1,}^d(\Phi_k). \tag{3.15}$$

A more detailed problem formulation and results for a German regional train line can be found in (Binder & Albrecht, 2013).

3.4 Case study

The performance-based timetabling approach has been applied on a case study of a central part of the railway network in the Netherlands, consisting of the railway network bounded by the four main stations Utrecht (Ut), Eindhoven (Ehv), Tilburg (Tb) and Nijmegen (Nm), with a fifth main station s-Hertogenbosch (Ht) in the middle and 20 additional smaller stations and stops. Four corridors connect Ht to the other main stations. The train line plan in this part of the network is taken from the 2011 timetable and consists of four intercity lines and six local train lines with a frequency of two trains per hour each, see Figure 3.6. The intercity lines 800 and 3500 offer a regular 15 min service between Ut and Ehv but have different origin/destinations outside this area. The regional line 13600 from Tb to Ht continues as the line 16000 from Ht to Ut, and vice versa. The line 9600 from Ehv couples in Ht to the line 4400 to Nm, and vice versa. In addition, an hourly freight path with maximum speed of 120 km/h is scheduled from Ut-Ehv. So overall, 41 trains are running per hour in this network. For the computation of the nominal running times we used 5% running time supplement for each train type. So the scheduled running times include at least 5% running supplements. As an illustration of our results we focus on the corridor Utrecht-Eindhoven in this paper, although we report the computation times for the entire network.

Table 3.3: Computation times (entire network)

	Iterations	Mean time [s]	Total time [s]
Initial microscopic computations	1	35	35
Micro-macro iterations			1080
Macro (1000 macro iterations)	9	80	
Micro computations	9	40	
Finetuning*			215
Micro computations	1	5	
Energy-efficient speed profiles	1	210	
Total			1330

*Excluding stochastic optimization of local trains

Table 3.3 shows the breakdown of the computation time of the optimized timetable for the entire network. The total computation time was 22 minutes. The initial microscopic calculations required 35 s. The micro-macro iterations converged in 9 iterations with a mean computation time of 2 minutes per iteration, with 80 s for the macroscopic calculations and 40 s for the microscopic calculations. The time to set up the input for the fine-tuning model was 5 s and the computation of the energy-efficient speed profiles for all train runs between stops for the fixed scheduled running times took another 210 s. The stochastic optimization of the local trains over all the corridors took some additional hours, but this can be seen as a final fine-tuning step which only changes the timing of short stops but does not change the timetable at the main nodes.

Figure 3.7 shows a time-distance diagram of the computed hourly timetable for the corridor Ut-Ehv. The vertical axis shows time in minutes downwards. The horizontal axis shows distance with the station positions indicated. The blue lines are IC trains, the magenta lines are local trains, and the green line is the freight train. Note that the sections Btl-Ehv and Htn-Htnc have four tracks. Figure 3.8 shows the corresponding blocking time diagram for the route of intercity train line 3500. Note that only the blocking times are shown for all trains running on the same tracks as train line 3500. The gaps in the blocking time stairways for some trains correspond to running on parallel tracks in stations or the four-track lines between Htn-Htnc and Btl-Ehv. Around Ht also some blocking times are visible corresponding to crossing trains from/to Tilburg or Nijmegen.

The optimized timetable shows periodic passenger trains with regular 15 min services of both IC and local trains where two similar train lines follow the same route. Hence, effectively 15 min train services are realized instead of two separate 30 min train lines. The ICs overtake the local trains at Geldermalsen (Gdm) in the southbound direction, but not in the return direction. The fast freight train departs after the local train from Utrecht Centraal (Ut) and overtakes this local train at the four-track line around Houten (Htn).

The blocking time diagram of Figure 3.8 shows no overlapping blocking times and hence illustrates that the timetable is conflict-free. Moreover, the timetable is robust

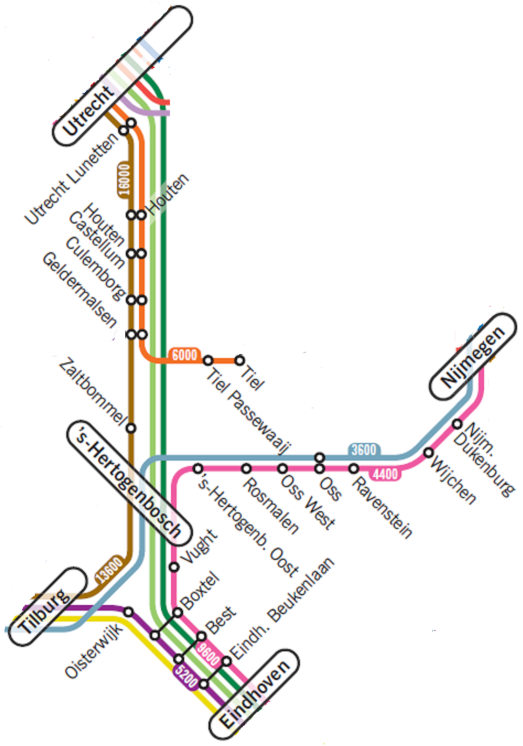

Figure 3.6: Passenger line plan of the Dutch case study

illustrated by the buffer times (white space) between the train paths. Only between Houten Castellum (the station just after Htn) and Culemborg (Cl) the freight path and the next local train are tight so that a slight delay of the freight train might propagate to the local train but the buffer time between this local train and the next IC prevents further knock-on delays. In Gdm, the local train also has a longer dwell time that can be used to recover from an arrival delay. In the absence of the freight train the situation is robust, which is the usual case currently with on average one freight path per two hours on this corridor.

Table 3.4: Infrastructure occupation

Corridors			Stations		
Corridor	Time [min]	Ratio [%]	Station	Time [min]	Ratio [%]
Ut-Ht	34.7	57.8	Btl	15.7	26.2
Ht-Ut	32.1	53.4	Ehv	15.7	26.1
Ehv-Ht	22.0	36.7	Gdm	15.7	29.5
Ht-Ehv	24.2	40.3	Ht	35.0	58.3
			Htn	15.0	25.0
			Ut	20.9	34.8
			Vga	17.2	28.7

Table 3.4 gives the infrastructure occupation of the main corridors and stations, respectively. All the infrastructure occupation percentages are below the recommended stability value of 60% defined by the UIC for mixed traffic corridors in daily periods,

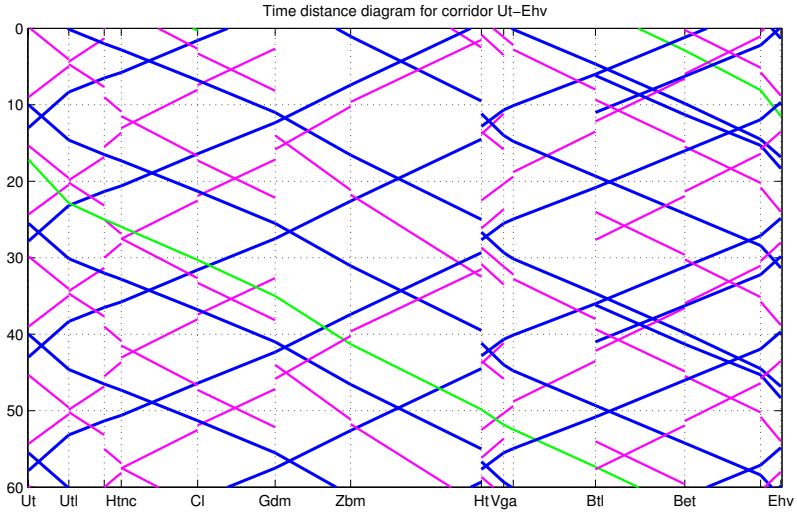

Figure 3.7: Time-distance diagram corridor Utrecht-Eindhoven

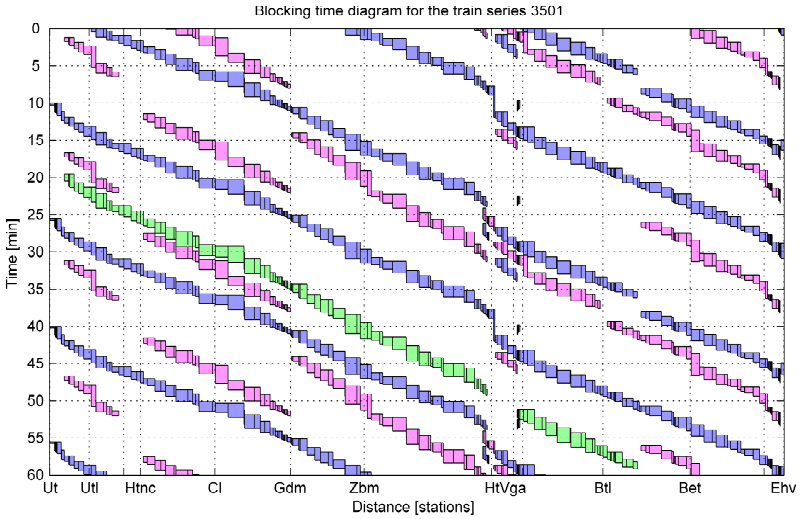

Figure 3.8: Blocking time diagram corridor Utrecht-Eindhoven

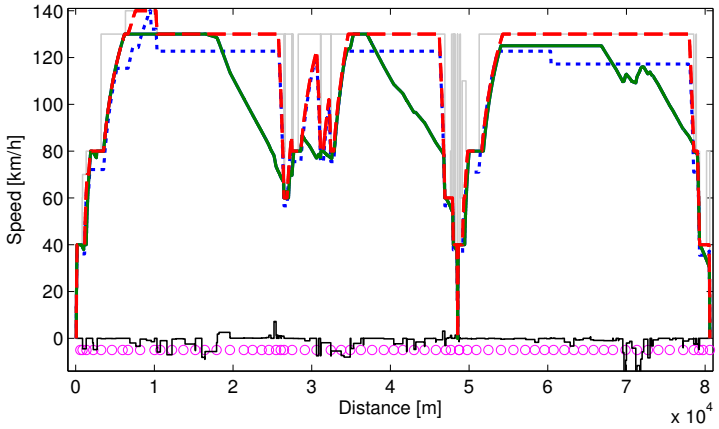

Figure 3.9: Speed profiles: static speed limit (solid grey), time-optimal (dashed red), reduced cruising speed (dotted blue), and energy-optimal (solid green)

which was one of the constraints of the timetabling algorithms. Corridor Ut-Ht is the heaviest used one with infrastructure occupation 57.8%. Ht has the highest infrastructure occupation of 58.3%, which includes also the crossing routes from/to Tilburg and Nijmegen. The relative low infrastructure occupation of corridors Ht-Ehv and back is due to the four tracks between Btl and Ehv.

Table 3.5: Journey times

O-D	Minimum [min]	Scheduled [min]	Increase [%]
Ut-Ehv	44.9	48.2	7.3
Ehv-Ut	47.6	51.3	7.8

Table 3.5 gives the average journey times over all trains running over the complete corridor from Ut to Ehv or backwards in a basic hour, i.e., eight IC trains and one non-stop freight train of 120 km/h speed limit. The minimum journey time refers to the minimum running and dwell times while the scheduled journey time includes the time supplements. On average, the time allowances over the complete corridor are 7.3% and 7.7% for the southbound and northbound directions, respectively, which can be exploited for energy-efficient driving.

Table 3.6: Energy consumption all trains

Speed profile	Energy consumption [kWh]	Energy saving [%]
Minimal-Time	64 395	-
Reduced cruising speed	58 800	8.7
Energy-optimal	41 667	35.3

Figure 3.9 illustrates the various speed profiles for the intercity line 3500 Ut-Ehv with intermediate stop in Ht. The bottom of the figure indicates the gradients (solid black line) and the signals (magenta circles) over the line. The dashed red line is the time-optimal speed profile corresponding to the minimum running times, while the dotted blue line is the operational speed profile with the running time supplements distributed over the line using reduced cruising speeds. The solid green line is the energy-optimal speed profile with clear coasting regimes before the areas with speed restrictions. Table 3.6 gives the total energy consumption of all trains running in the network of the case study, so 21 trains with all passenger trains counted once (corresponding to a basic half hour timetable including the freight train). With respect to the minimum running times the running time supplement saves 8.7% energy consumption when cruising at a reduced speed and even 35.3% using the energy-optimal speed profile with coasting. As was illustrated in Figure 3.9 for the IC 3500, the time supplements of the trains are distributed well over the corridor so that coasting could be applied very effectively. While concentrating on this corridor, the various local train lines might be optimized within the corridor-fine tuning. Therefore, the important network points have to be defined, e.g., Ut and Ht, but also important overtaking points such as Geldermalsen (Gdm) or smaller stations with operational importance such as Boxtel (Btl). Consequently, the

number of important network points within a very dense network such as the Netherlands, is very high. This leads to a very restrictive optimization potential within the corridor fine-tuning. Furthermore, the dwell times at the minor station, such as Best (Bet) or Eindhoven Beukenlaan (Ehb) are high compared to the planned running times between Ehv and Btl. Because of these restrictions the corridor fine-tuning is not able to offer high additional energy saving in the Netherlands, but would be suitable in networks with less traffic demand.

Table 3.7: Comparison between optimized and original timetable

KPI	Measure	Original timetable	Optimized timetable
Journey time efficiency	Max journey time increase [%]	33.3	14.2
	Mean journey time increase [%]	18.3	9.1
Timetable feasibility	Unrealizable minimum running times [#]	9	0
	Unrealizable nominal running times [#]	16	0
	Conflicts [#]	1	0
Infrastructure occupation	Max infra occupation corridor [%]	61.4 (Ut-Ht)	57.8 (Ut-Ht)
	Max infra occupation stations [%]	57.4 (Ht)	58.3 (Ht)
Stability	Infrastructure occupation $\leq 60\%$	no	yes
Robustness	Average settling time [s]	-	14340
Energy consumption	Energy saving [%]	14.7	8.7

Table 3.7 compares the KPIs from Section 3.2 computed for the optimized timetable and the original basic hour timetable from 2011 over the network corresponding to Figure 3.6. Journey time efficiency is measured with both the maximum and mean journey time increase over the minimum journey time. The maximum and mean journey times are considerably less for the optimized timetable as a result of the optimization. Zooming in on the scheduled running times, we see that the original timetable has 9 train runs with scheduled running times smaller then the minimum running time (with 291 s total running time shortage), and an additional 7 train runs for which the nominal running time with 5% supplement is not realized. In practice, these 16 train runs will result in delays and possibly conflicts. In general, the original timetable has much more time supplements but they are unevenly distributed with more supplements allocated before the main stations. Moreover, the original timetable has 1 scheduled conflict (with 46 s overlap). The optimized timetable is realizable and conflict-free. For the original network the infrastructure occupation has been computed with the unrealizable running times replaced by the minimum running times (and corresponding speed profile and blocking times). The maximum corridor infrastructure occupation is for both timetables on the corridor Ut-Ht, with the original timetable exceeding the UIC stability norm of 60% for daily periods. The other corridors for the original timetable are stable. Station Ht has in both timetables the highest infrastructure occupation, with the optimized timetable slightly higher but within the stability norm of 60%. For the robustness evaluation we used the stochastic delay propagation algorithm as implemented in the framework and computed the average settling time. The optimized timetable settled on average within 4 hours (14340 s) from initial delay scenarios with a mean delay of 523 s and a maximum delay of 1365 s. The implemented delay

propagation requires a feasible timetable and therefore could not be used to evaluate robustness of the original timetable, which had 9 unrealizable running times and an additional conflict that generated structural delays and thus never settled. A different delay propagation algorithm could be used that is able to deal with structural delays, such as described in Goverde (2010). The used initial delay scenarios in the optimization assumed quite large delays leading to a big delay propagation, which in practice would lead to timetable adjustments by traffic control. A benchmark for a settling time norm does not yet exist. Note that in the macroscopic optimization we did not consider a hard constraint on the setting time but just took the timetable with the best robust cost incorporating settling time performance. The reported energy savings are relative to the minimal-time running times, and computed with respect to cruising at a reduced speed determined by the available running time supplements. Again, for the original timetable we used the minimum running times for the unrealizable running times. The energy consumption of the original timetable was computed as 54935 kWh, which is less than the optimized timetable despite some unrealizable running times. This can be explained by the excessive running time supplements in the original timetable. In conclusion, the optimized timetable outperforms the original timetable, which is slower, unstable, contains structural delays, and a scheduled conflict. This demonstrates the validity of the performance-based timetabling approach and its implementation in the three-level framework.

3.5 Conclusions

This paper proposed to integrate timetable construction and evaluation into one consistent framework. The advantage of this approach is that performance indicators are already taken into account during the timetable construction by which the resulting timetable is computed together with all performance measures which are either satisfied or optimized depending on the required criteria. This relieves the tedious task of ex-ante simulation that some railways apply to test the constructed timetable on e.g. conflicts, stability, and robustness. Moreover, it is a notorious difficult issue for timetable planners to adjust the timetable if the simulation output indicates timetable flaws. Each local change may have an impact elsewhere. In our approach, we replace the feedback from timetable evaluation to timetable adjustment by an integrated approach embedding the timetable evaluation in the construction process. This has been made possible by the advances in both microscopic and macroscopic timetable models, but also by efficient and consistent data transformations between various levels. This enables an effective framework where microscopic details can be combined with macroscopic optimization over large networks, including stochastic models for robustness evaluation.

This paper presented a three-level modular performance-based timetabling framework to integrate timetable construction and evaluation, and illustrated it to a case study from the Netherlands showing excellent results on all performance indicators. In particular, the approach highlighted eight recommendations that need to be considered explicitly

in the design of a stable robust conflict-free timetable with optimal journey times:

- Microscopic calculations of running and blocking times taking into account all running route details at section level (gradients, speed restrictions, signalling),
- Microscopic conflict detection guaranteeing a conflict-free timetable,
- Timetable precision of 5 s (or at most 10 s) to minimize capacity waste,
- Incorporation of (UIC) infrastructure occupation and stability norms,
- Macroscopic network optimization with respect to trip times, transfer times, cancelled train path requests and associated cancelled connections,
- Macroscopic robustness analysis using stochastic simulation to obtain a robust network timetable,
- Stochastic optimization of timetables for local trains on corridors taking into account stochastic dwell times at intermediate stops,
- Energy-efficient speed profiles computed and incorporated for all trains.

Moreover, standardized data exchange files such as railML for infrastructure, rolling stock, and the timetable is recommended, where the presented timetabling approach generates an output Timetable railML with scheduled train paths at (track-free detection) section level, extended with scheduled energy-efficient speed profiles. The modularity of the framework allows any algorithm to be replaced by any other algorithm which is further supported by the use of standardized data formats making the framework very flexible. The implemented algorithms use generic models that can be configured for any specific railway characteristics making the implemented framework internationally applicable.

Chapter 4

Microscopic models and network transformations for automated railway traffic planning

---

Apart from minor updates, this chapter has been published as:

Bešinović, N., Goverde, R. M. P. & Quaglietta, E. (2017). Microscopic Models and Network Transformations for Automated Railway Traffic Planning. *Computer-Aided Civil and Infrastructure Engineering*, 32 (2), 89-106.

4.1 Introduction

Timetabling is one of the major planning tasks in railway traffic and becomes increasingly complicated with the increasing demand for more services. Planners are constantly under pressure to fit additional trains into busy schedules while at the same time maintaining and improving the level of service such as seamless connections and punctuality. Timetables need to provide accurate time-distance infrastructure slots, or train paths, that secure conflict-free train runs. Moreover, the plan must adhere daily stochastic variations in the train services, i.e., be robust.

Integrated automatic timetabling models provide fast solutions that allow analyses of multiple timetable scenarios and tweaking different planning criteria. This will eventually lead to a better understanding of the capacity use and overall high-quality timetables. Tsiflakos and Owen (1993) already stressed the importance of automated decision support and presented a structural representation of railway data necessary for any further application of optimization techniques. Indeed, there is an evident need for modelling approaches that allow an efficient use of optimization algorithms and other supporting models in timetabling.

We make a distinction regarding the level of detail considered in timetabling. Two approaches can be recognized – microscopic and macroscopic. The latter considers the railway network at a higher level, in which station is represented as a node and tracks by linking arcs. In a microscopic approach, detailed infrastructure aspects like speed limits, gradients, curves are signaling system are considered. In this paper, we introduce microscopic models that can accurately evaluate timetables and support macroscopic models to construct operationally acceptable timetables which are feasible and stable. The railway research on both microscopic and macroscopic models attracted significant research (Castillo et al., 2015; Peng et al., 2011; Sels, Dewilde, Cattrysse, & Vansteenwegen, 2016; Xie, Ouyang, & Somani, 2016).

An extensive review of timetabling models is given in Cacchiani and Toth (2012). Kroon et al. (2009) presented the practical implementation of a set of optimization models for Netherlands Railways. These optimization models assumed a macroscopic infrastructure model using default norms for safe separation times of following, crossing and meeting trains. This normative approach cannot guarantee to solve all route conflicts in the computed (macro) timetable, or on the other hand may lead to inefficient large buffer times. Moreover, scheduling train paths over the given infrastructure and the capacity assessment of the resulting timetable are separated processes. Therefore, macroscopic approaches should be integrated with more detailed models that ensure the operational feasibility of the timetable.

Timetable feasibility is the ability of all trains to adhere to their scheduled train paths. A timetable is feasible if (i) the individual processes are realisable within their scheduled process times, and (ii) the scheduled train paths are conflict free, i.e., all trains can proceed undisturbed by other traffic. A *conflict* is defined as an overlap (in time and space) between two trains on the same route which represents that one train cannot use the railway infrastructure without interfering the other train. A few approaches have been proposed in literature based on a hierarchical integration of timetabling models with different level of details (Caimi, Fuchsberger, Laumanns, & Schüpbach, 2011; De Fabris, Longo, Medeossi, & Pesenti, 2013; Gille et al., 2008; Schlechte et al., 2011). The current integrated models using microscopic details for timetabling, do not perform any feasibility check of the timetable produced, except for Schlechte et al. (2011); while none of them considers any iterative modification to the timetable if it is proved to be infeasible at the microscopic level. In other word, Schlechte et al. (2011) used a microsimulation for conflict detection, while none of the approaches consider any conflict resolution methods. Hence, these models solve the timetabling problem in one direction only and thus represent an open-loop strategy.

D'Ariano et al. (2007) proposed a model for real-time train rescheduling that includes a feasibility check and recomputing speed profiles with some simplifying assumptions, such as trains running at maximum speeds with possible braking at conflicts, a simplified interlocking model at station layouts, and fixed speed-independent clearing times. In our model, we explicitly compute operational running times, sight, setup and clearing times, and consider track sections instead of block sections which in particular

matters in station areas with switches. This provides a more accurate conflict detection. A review of other real-time rescheduling models can be found in Cacchiani et al. (2014).

Timetable stability is defined as the capability of absorbing train delays (UIC, 2013). As a stability measure, we adopted the UIC recommendation that a timetable is stable if capacity occupation rates are under certain norms depending on the traffic structure. *Capacity occupation* is defined as the time share needed to operate trains according to a given timetable pattern taking into account scheduled running and dwell times. Thus, we first compute the capacity occupation for stations and corridors and then compare obtained values with the UIC norms. The current practice of a posteriori capacity assessment of the final timetable is not efficient: a lot of time may be invested in producing a timetable that afterwards may not satisfy the stability norms.

Within tactical railway planning, capacity assessment is generally based on the microscopic UIC compression method (Landex & Jensen, 2013; UIC, 2013), while stability on the network level can be assessed by the stability analysis tool PETER (Goverde, 2007). The UIC method have been developed for assessing lines and corridors. However, the main limitation of the UIC method is that it computes the capacity in station areas by considering the platform tracks separately from the interlocking areas in between the home signals and the platform tracks (Armstrong et al., 2015; Lindner, 2011) This independence assumption results in an underestimated station capacity.

Nash and Huerlimann (2004); Siefer and Radtke (2006) and Quaglietta (2014) presented advanced microscopic simulation tools, which are able to accurately simulate railway operations based on a detailed modelling of infrastructure, signalling and train dynamics that could be used to detect conflicts in a timetable. However, these multi-purpose microscopic simulation models need long computation times to evaluate conflict-freeness of timetables on large and heavily utilized railway networks. Therefore, they are not suitable for fast analyses during the design of a timetable.

Train running time computations are one of the most common models in railway applications, and have been used for computing minimum running times in timetable planning or for energy efficient driving in real-time applications. These models are commonly including principles of optimal control theory. A detailed review can be found in A. R. Albrecht, Howlett, Pudney, and Vu (2013), or Scheepmaker and Goverde (2015). *An operational speed profile* is the one that exploits existing time supplements between departure and arrival times to allow the train arriving on time, instead of being ahead of scheduled. The operational speed profile is used to assert that an acceptable speed profile exists for allocated time supplements. For example, it may occur that a macroscopic timetable assigned an excessive running time supplement that would require a train to run very slow, below a certain practical minimum speed. Such a speed profile should be avoided. Second, we need the operational speed profiles for detecting timetable conflicts and assessing the infrastructure capacity.

Communication between microscopic and macroscopic models is essential for efficient

and consistent bidirectional transformations. These transformations would allow generating accurate input to a macroscopic model on one side, and evaluating a timetable on the detailed microscopic level on the other. Schlechte et al. (2011) introduced a micro to macro transformation, but the reverse transformation from macro to micro has not been described in the literature yet.

The state-of-the-practice suggests that improvements in the timetable planning process are necessary in various directions (ON-TIME, 2016). Most notably, a timetable is expected to be realisable considering a great level of detail including infrastructure, rolling stock, signalling and automatic train protection (ATP). Second, timetabling tools should work as a whole, as well as in-terms of individual functions, i.e., a stepwise development is recommended. Third, the final timetable should satisfy specified values for performance measures such as feasibility, capacity occupation, robustness, and energy consumption (Goverde & Hansen, 2013). Finally, it is important to reduce the computation time of the planning tools.

In order to overcome the limitations in the state-of-the-art and answer the questions from practice, we developed a hierarchical framework of performance-based railway timetable design in the European FP7 project ON-TIME (Optimal Networks for Train Integration Management across Europe) (Goverde et al., 2016). In particular, the framework includes microscopic models presented in this paper and a macroscopic timetabling model that interact iteratively by adapting microscopic running and minimum headway times until the produced macroscopic timetable is proved feasible and stable.

The aim of this paper is to provide a methodology for timetable design that will cater for more structural insight into a timetable and make the process itself more efficient, which would result in timetables of a higher overall quality. In the past, we introduced a conceptual ON-TIME framework (Bešinović et al., 2014). In this paper, we describe the deterministic microscopic timetabling models and provide efficient automatic transformations between microscopic and macroscopic networks. Microscopic models compute accurate running and minimum headway times that are used as input to a macroscopic model, and verify that the timetables produced by the latter are feasible at the level of track sections. For timetable evaluation, and particularly the micro-macro framework, operational speed profiles may be recomputed numerous times. Thus, we define a new model for fast computing operational speed profiles, although various models based on optimal control exist in the literature. Stability is checked by verifying that the infrastructure capacity occupation respects the UIC guidelines (UIC, 2013). We propose an analytic model for capacity assessment that efficiently deals with both stations and corridors. Network transformations are required to provide the relevant data for specific computations. Aggregating the data to a macroscopic level allows the application of macroscopic optimisation models while considering a consistent operationally relevant railway infrastructure. After computing a macroscopic timetable, the reverse transformation is applied from macro to micro. This is done by recomputing the operational speed profiles with respect to the arrival/departure times

from the macroscopic timetable. All microscopic models can be used for designing and evaluating both periodic and non-periodic timetables and each model can be used individually or as a building element of the timetabling framework. The microscopic models have been tested on a part of the Dutch railway network including the main corridor Utrecht–Den Bosch–Eindhoven.

The main contributions of this paper are the following:

- Fast computation of operational train speed profiles from scheduled event times that enable microscopic timetable evaluation;
- Capacity assessment based on max-plus automata that compute the capacity occupation in stations more realistically than the current UIC method;
- Automatic conflict detection that accurately determines existing conflicts at the level of track sections;
- Consistent network transformations from micro to macro and vice versa.

The remainder is organized as follows. Section 2 gives the structure of the general framework. Section 3 describes the network and data modelling. It also includes conversions from micro to macro and vice versa. Section 4 presents a detailed description of the microscopic modules and their functions. Further, it introduces the basics of max-plus automata theory and its application to the UIC compression method. Section 5 illustrates the approach in a Dutch case study. Section 6 reflects on the developed models and finally Section 7 presents conclusions and future research.

4.2 The micro-macro timetabling approach

The ON-TIME project defined a framework for achieving high-quality railway timetables with an integrated set of state-of-the-art timetabling techniques. More details about the models used and the framework developed can be found in Goverde et al. (2016). One of the main objectives of the project was to build up ‘a scheduled train-path assignment application, with automatic conflict detection capabilities, that builds on the concept of robust timetables, has a unified network coverage, is microscopic at selected parts of the control area, is scalable, and able to connect to Traffic Management Systems, with user-friendly interfaces and execution states that correspond to the IM timetabling management milestones.’ This objective has been reached by the two-level functional framework represented in Figure 4.1, which indicates the interactions among the microscopic and macroscopic models.

Input data of the framework are microscopic characteristics of the infrastructure (e.g. track gradients, position of stations, switches), the rolling stock (e.g. mass, number of coaches, tractive effort-speed curve, resistance parameters), the signalling and ATP system (braking behaviour, signal aspect sequence) and the interlocking (e.g. local

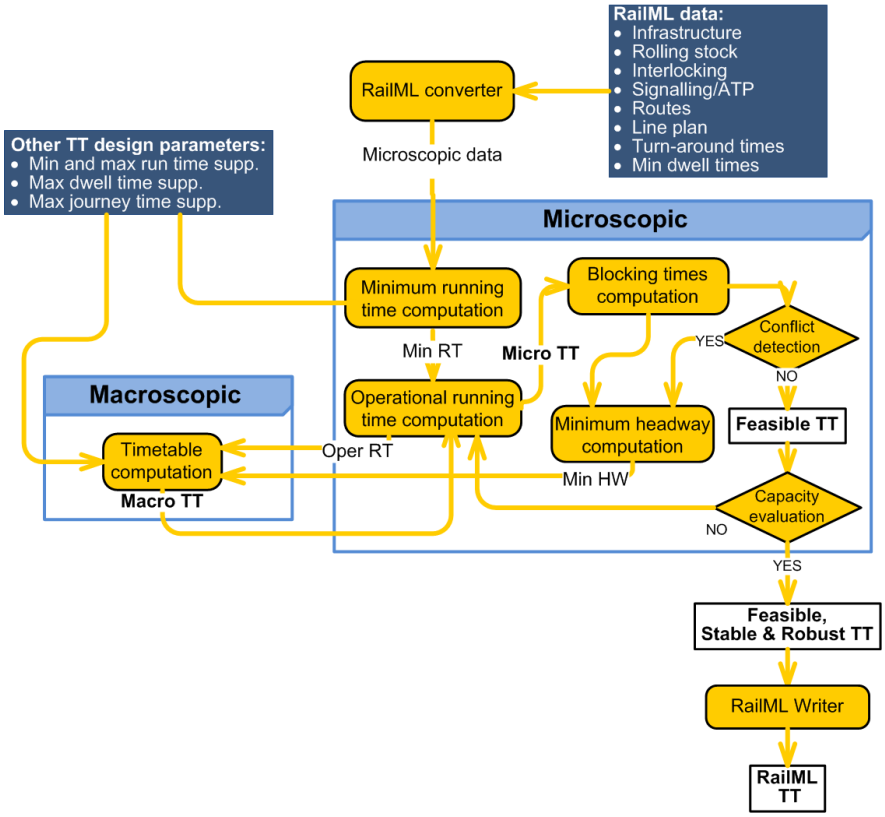

Figure 4.1: Scheme of the micro-macro framework for timetable design

feasible routes). Both input and output of the framework are in a standardized railway data format, known as RailML (RailML, 2015).

The timetabling computation is an iterative process of two models:

- A microscopic model that computes reliable train running and blocking times at a highly-detailed level and checks for feasibility and stability of the timetable,
- A macroscopic model that produces a timetable at aggregated network level, by identifying arrival/departure times at/from stations or junctions in order to optimize a given objective function (e.g. minimise journey times). This is an optimization model that can also provide timetables that are robust versus stochastic operation disturbances.

In the first iteration a timetable is not available yet, so the microscopic model computes minimum running times and blocking times, which are aggregated to macroscopic running times and minimum headway times and sent to the macroscopic model to calculate a timetable. When a macroscopic timetable has been produced this is sent back to the microscopic model which computes updated blocking times required for detecting track conflicts based on the operational running times (i.e. the running times including time supplements scheduled by the macroscopic timetable). If there are track conflicts, these are resolved and minimum headway times are computed which are

transferred to the macro model again. This iterative process is repeated until all track conflicts have been solved and the macroscopic timetable can be defined as feasible.

In the next step, the microscopic model evaluates the stability of the timetable. If the timetable is not stable enough, new operational running times are computed by e.g. increasing the value of time supplements and/or buffer times. This is performed until the timetable stability is also verified. For the transformations from the microscopic level to the macroscopic level, and vice versa, efficient procedures have been developed to aggregate and disaggregate input and output data. In general, microscopic models are necessary to 1) compute initial input data for the macroscopic timetabling model, 2) assess the timetable feasibility and stability when used independently and 3) guarantee operational feasibility and stability when included in the micro-macro framework.

4.3 Network and data modelling

As already pointed out by Tsiflakos and Owen (1993), we need to use structurally organized input data. In the past years a significant effort has been seen in defining a standardized railway data format RailML. This RailML data format is more and more adopted for communication between railway software tools, and therefore we also adopted this RailML data exchange format. The input to our models thus consists of a set of RailML files composed of: a) Microscopic infrastructure data, b) rolling stock data, including train formations, c) Interlocking, signalling and automatic train protection system (ATP) data, d) available routes, and e) train lines. A *train line* is defined with origin and destination points, stopping pattern at timetable points (stations, stops) and a corresponding rolling stock type. It also includes the service category, such as local or intercity, and the frequency represented in number of trains per hour. These data are converted to a suitable internal format of ascii data which is used by the microscopic models. Additional parameters, such as connections and transfer times, dwell times, and other timetable design parameters and norms are provided externally. The hierarchical framework for timetable design is composed of two network models that respectively represent the same network with a microscopic and a macroscopic level of detail.

4.3.1 Network modelling

Microscopic network

The microscopic model considers *homogeneous behavioural sections* for the accurate computation of train speed profiles and running times (Figure 6.2a). A homogeneous behavioural section is defined as a section with a certain length l, and constant characteristics of speed limit v_{lim}, gradient g and radius ρ. The microscopic network is based on a graph whose arcs are obtained by aggregating the homogeneous behavioural sections into track sections denoted by b. A track section corresponds to a track-free detection section, or several track-free detection sections including at most a single

switch. On the open track, a block is considered as one track section, while in interlocking areas one block may include multiple sections. The nodes of the microscopic network coincide with the joints between consecutive block/track sections or to infrastructure elements such as signals, switches and platforms. This level of infrastructure details allows very accurate infrastructure capacity assessment, feasibility checks and minimized wasted capacity, which is particularly important in highly utilized networks.

We distinguish between functions working on the behavioural section level of the infrastructure network on one hand, and the track section or block section level on the other. Computations of minimum and operational running times and corresponding speed profiles are applied on the former, while computation of blocking times and minimum headway times, conflict detection, and capacity assessment are applied on the latter. We also define a set of *microscopic timetable points* K, where each point k represents an infrastructural point of interest such as *stations* that provide passenger (and/or freight) interaction and allow train overtaking, *stops* that do not have enough tracks to facilitate overtaking or dwelling of more than one train, and *junctions* where two or more railway lines intersect or merge and no trains are scheduled to stop.

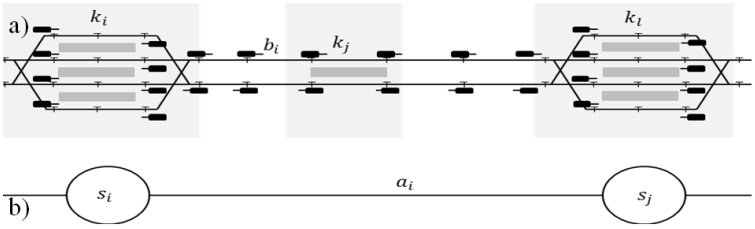

Figure 4.2: Representation of a (a) microscopic network and (b) macroscopic network

Macroscopic network

The macroscopic network $N = (S,A)$ is automatically produced from the microscopic one and used for the macroscopic timetabling model (Figure 6.2b). Nodes in a macroscopic network are referred as to *macroscopic timetable points*, $s \in S$. The potential candidates for s are stations and junctions from K. An arc $a \in A$ represents the corridor between two successive macroscopic points s_i and s_j. Each arc is comprised of a set of microscopic arcs, $a = (b_1, b_2, \ldots, b_n)$. The generation of a macroscopic network is explained in Section 3.3.

4.3.2 Timetables, trains and routes

We distinguish between a microscopic and macroscopic timetable. A *macroscopic timetable* (*macroTT*) includes scheduled running, dwell and transfer times as well as event times such as arrivals, departures and passages between and in macroscopic points. A *microscopic timetable* (*microTT*) includes all the aforementioned event times for microscopic timetable points and the corresponding train speed profiles defining the exact train behaviour.

The set of trains is indicated by T. For each train $t \in T$, $S_t \subseteq S$is a set of served macroscopic timetable points. We assume that for each train the route ρ_t (i.e., the sequence of traversed tracks without the corresponding travel times) is provided. Here, we differentiate between a microscopic route $\rho_t^{micro} = (b_1, b_2 \ldots, b_{n_t})$, where n_t is the number of microscopic arcs for train t, and a macroscopic one $\rho_t^{macro} = (a_1, a_2 \ldots, a_{m_t})$, where m_t is the number of macroscopic tracks for train t.

For each train $t \in T$ and each macroscopic arc a the minimum running time r_{ta}, the nominal running time $\underline{r}_{ta}$ including a running time supplement, and the maximum running time $\overline{r}_{ta}$ are given. All running times are computed by microscopic algorithms, while the nominal and maximum ones are given as input to the macroscopic model. The scheduled running times in macroTT are called operational running times and denoted as $\tilde{r}_{ta}$. Similarly, we define running times $r_{tk_1k_2}$, $\overline{r}_{tk_1k_2}$, $\tilde{r}_{tk_1k_2}$ between two microscopic points k_i and k_j, representing the minimum, maximum and scheduled ones, respectively.

For each train $t \in T$ and each microscopic point $k \in K$ the nominal dwell time $\underline{w}_{tk}$ and maximum dwell time $\overline{w}_{tk}$ is provided. Since the aim of timetable planning is to provide an acceptable quality of service, certain *design norms* need to be predefined. The set of these parameters consists of minimum transfer times, turnaround times, minimum and maximum running time supplements (%), and maximum allowed journey times of train lines (%).

4.3.3 Microscopic to macroscopic conversion

Algorithm 2 describes the automatic procedure for the micro to macro network and data transformations, which are similar to Schlechte et al. (2011). Our approach differs in two points. First, the algorithm of Schlechte et al. (2011) does not compute running or blocking times, but uses the commercial software OpenTrack to do so. Second, their algorithm performs a search over all infrastructure elements (i.e., block sections) to determine macroscopic points, while we do it exclusively over microscopic timetable points. Note that a set of microscopic points is quite extensive and includes much more than just stations and stops, but also each important junction, switch, crossing, movable bridge or platform. In terms of complexity, this means that our algorithm has significantly less work than that of Schlechte et al. (2011)., making our model computationally faster. The CPU time for our micro to macro conversion is under one second.

The conversion from microscopic to macroscopic models includes three steps: computing process times, generating a macroscopic network, and aggregating process times for the macroscopic network. The algorithm first computes the minimum running times and corresponding blocking times. Then, it aggregates microscopic arcs (track sections) b_i to macroscopic arcs $a = (b_1, b_2, \ldots, b_n)$. Each arc a is described with the number of tracks and its orientation (mono- or bidirectional). The former is determined by identifying different routes between two nodes using the function *DetermineTracks*,

while function *DetermineDirection* determines the latter. The subset of macroscopic points S is then derived from the microscopic points K. The algorithm compares all pairs of train routes separately. The macroscopic point is chosen based on the interplay between train routes. The microscopic point k is in S only if i) any two routes are converging, diverging or crossing in k, or ii) k is the origin or destination point of any route. For example, for two routes using microscopic points $\{k_1,\ k_2, k_3, k_5\}$ and $\{k_1, k_2,\ k_4\}$, respectively, the set of macroscopic points is $S = \{k_1,\ k_2, k_4, k_5\}$. Point k_2 is included because it is a diverging point (first criterion), while k_1, k_4 and k_5 satisfy the second criterion.

After initialising the macroscopic network, headways are determined at each macroscopic point s and for all possible interactions between each two train routes. The computation of the blocking times and minimum headway times are executed on the block section level of the infrastructure network. Once all process times are computed on the microscopic models, the algorithm performs the aggregation of process times and the discretisation of time. The function *AggregateProcessTimes* aggregates the microscopic running times (i.e., between each two microscopic timetable points) to aggregated process times between two timetable points in the macroscopic network. The minimum running time r_{ta} between two macroscopic points may comprise several microscopic running times and dwell times since $S \subset K$, i.e., not all micro points are in S. The nominal running time over a is obtained by adding a running time supplement λ_{min} to the minimum running times plus the intermediate dwell times:

$$\underline{r}_{ta} := \sum_{i=1}^{m} (1 + \lambda_{min}) r_{tk_i k_{i+1}} + \sum_{i=1}^{n} w_{k_i},$$

where arc a is bounded by some macro points $[s_i,\ s_j]$, m is the number of consecutive running sections, and n is the number of intermediate microscopic points between s_i and s_j. Likewise, the maximum running time $\bar{r}_{ta}$ over a is obtained with respect to a maximum running time supplement λ_{max}. Initially, λ_{min} is provided such as 5%. In any following iteration it is computed from the macroTT returned by the macroscopic timetable model.

The macroscopic model may use a coarser time granularity, so a time discretisation of process times is performed as well. The incorporated function represents an innovative rounding method that has the objective to control the rounding error by combining rounding up and rounding down. By applying *AggregateProcessTimes*, we obtain all process times that are necessary for macroscopic computation.

The network transformation is applied in the initial stage of timetable planning to provide the required network input to a macroscopic model since the given line requests (origin/destinations and stop patterns) are considered as fixed. Hence, the macroscopic network structure remains the same during all iterations. On the other hand, *AggregateProcessTimes* is run each time (e.g., iteration) a data input (for a macroscopic model) is adjusted based on the output of the microscopic models such as the updated train speed profiles, running times and headway times that need to be aggregated for

each new run of the macro model.

Algorithm 2 Micro to macro conversion

Input: Microscopic network M, microscopic points K, dwell times W,
timetable design norms λ, set of trains T
Output: macroscopic network $N = (S,A)$, macroscopic running,
dwell and headway times
Forall $t \in T$
Compute microscopic running times $R_{t,micro}$
Compute blocking times B_t
End Forall
Forall microscopic timetable points $k \in K$
Forall pairs of train lines
If k is origin or destination point OR lines converge OR lines diverge
OR lines cross
add k to macroscopic nodes: $S \rightarrow S \cup k$
End If
End Forall
End Forall
Forall adjacent timetable points $s \in S$
Create a macroscopic arc $a = (s_i, s_j)$
DetermineTracks of arc $a = \{b_i\}$, $i = 1, \ldots, n$
DetermineDirection of arc a
End Forall
Forall macroscopic timetable points $s \in S$
Compute minimum headway times $h_{st_i t_j}$
End Forall
AggregateProcessTimes for $N = (S,A)$

4.3.4 Macroscopic to microscopic conversion

After obtaining a macroTT, we need to translate it to a microscopic level of detail in microTT, see Algorithm 3. In other words, from the scheduled event times for the macroscopic timetable points we reconstruct the train speed profiles and scheduled times for all microscopic timetable points. To do so, we apply the following three steps for each train. Step 1 derives running time supplements for a macroscopic route ρ_t^{macro} and distributes them to the corresponding microscopic route ρ_t^{micro}. Step 2 determines the operational speed profile for the given time supplements (Section 4.2). Finally, the computation of blocking times concludes Step 3 (cf. Section 4.3). Step 1 is explained in more details in the following subsection and is followed by an example of the macro to micro conversion.

Allocation of running time supplements

In Step 1 we determine the running time supplements that are allocated in a given macroTT. Based on the scheduled running time (difference between the scheduled departure time and scheduled arrival time at the next considered point.) in macroTT, we

Algorithm 3 Macro to micro conversion

Input: microscopic network M, *macroTT*
Output: *microTT*
Forall trains $t \in T$
1. *Determine allocated running time supplements* Ψ_t
2. *Compute operational speed profiles (see Section 4.2)*
3. *Compute blocking times* B_t*(see Section 4.3)*

End Forall

compute the corresponding allocated running time supplement between two macroscopic points. We denote ψ_{ta} as the difference between the scheduled and minimum running time for macroscopic arc a of train t, $\tilde{r}_{ta}$ and $\underline{r}_{ta}$, respectively. This defines a vector Ψ_t of the time supplements ψ_{ta} between each two macroscopic timetable points over the corresponding route q_t^{macro}. This is done for all trains $t \in T$.

Recall that the not all microscopic timetable points are necessary also macroscopic, but $S \subset K$. This means that several microscopic timetable points may exist between two adjacent macroscopic points. By computing an operational train speed profile over an arc a and considering just a given time supplement ψ_{ta}, one may obtain an unequal distribution of time supplements between two consecutive microscopic timetable points. Hence, we need to migrate from time supplements over arcs, to the lower level, i.e., time supplements between each two microscopic points, which results in distributing time supplements in a more justified manner. In order to do so, we assign ψ_{ta} proportionally to all sections between each two adjacent microscopic points based on the running time over that section. So, each section k_1k_2 receives a portion: $\psi_{tk_ik_j} = \psi_{ta} r_{tk_ik_j}/r_{ta}$, where r_{ta} is the minimum running time between two macroscopic points over arc a, $r_{tk_ik_j}$ the one between two microscopic points k_i and k_j and $\psi_{tk_ik_j}$ is the corresponding running time supplement. By doing this, we enforce an equal time supplement distribution and prevent that some sections get no time supplements.

Figure 4.3 gives a graphical representation of the macro to micro transformation for a given train t operating between A and D. Let A and D be macroscopic timetable points, while B and C the microscopic ones. The macroscopic arc $a = (A,D)$. The train stops at all points. Solid lines represent scheduled running times, and dashed lines are the minimum ones. The macroscopic timetabling model produces (macroTT) the scheduled running time, $\tilde{r}_{ta}$, and corresponding minimum running time r_{ta}. Step 1 computes the running time supplement ψ_{ta}, as $\psi_{ta} = \tilde{r}_{ta} - \underline{r}_{ta}$ (Figure 4.3a). Then, ψ_{ta} is distributed proportionally between each two neighbouring microscopic points (Figure 4.3b). In the Figure 4.3c the dotted line is the static speed limit along the route. Step 2 computes the operational speed profiles for each section between two microscopic points (Figure 4.3c) which is explained in the following section.

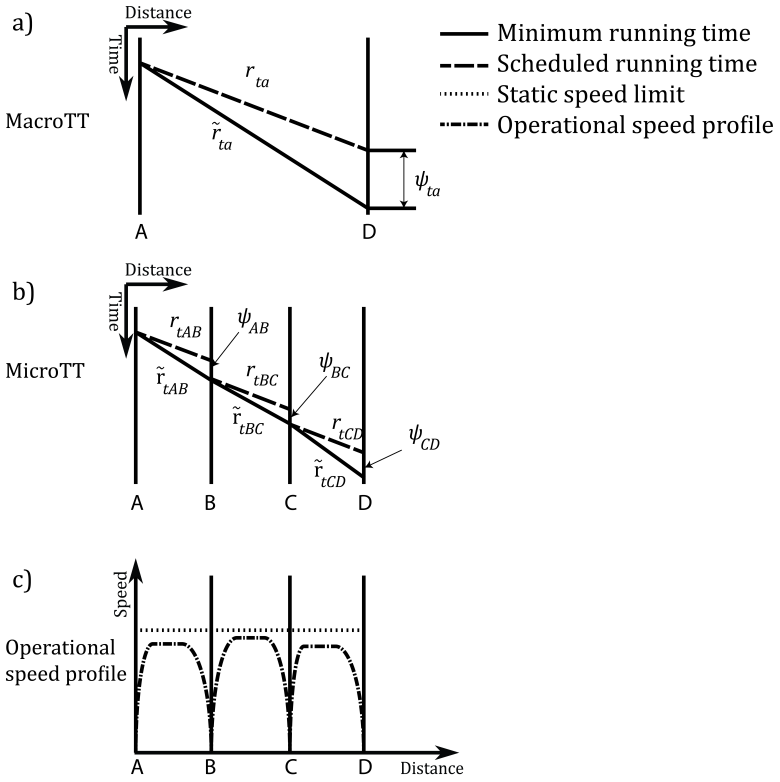

Figure 4.3: Macro to micro transformation

4.4 Microscopic computations

4.4.1 Minimum running times

The minimum running time is the time required for driving a train from one infrastructure point to another assuming conflict-free driving as fast as possible. Running times are computed using microscopic train dynamic models that require detailed rolling stock and infrastructure data, including route-specific static speed profiles.

Running times are modelled by means of the Newton's motion equations (Brünger & Dahlhaus, 2014). The tractive effort is assumed a piecewise function of speed consisting of a linear part and one or more hyperbolic ones. The resistance force is modelled based on the Davis resistance equation, a second-order polynomial of speed. The braking rate is defined as a single deceleration rate. A train speed profile and the associated running time are determined as function of distance (Bešinović, Quaglietta, & Goverde, 2013). These first-order ordinary differential equations are solved by the numerical Dormand-Prince method (Butcher, 2008), which is a variant of the more general Runge-Kutta approach. The output of this function constitutes microscopic running times $r_{tk_ik_j}$ for each $t \in T$ and where k_i and k_j are the subsequent microscopic points along the route ρ_t^{micro}. It also includes the corresponding train speed profiles, i.e., time-distance and speed-distance diagrams.

4.4.2 Operational running time computation

In Step 2, for each train $t \in T$ and corresponding Ψ_t, we compute the operational running time consisting of the detailed train speed profile and scheduled times at microscopic timetable points, which are used for further microscopic analyses as conflict detection and capacity assessment.

By definition, the scheduled running time contains time supplements added to the microscopic minimum running time to absorb a stochastic variation of train runs during real operations. In the initial stage of the timetable planning, the time supplement is usually 5% of the minimum running time, which is a common value for Netherlands Railways. At the microscopic level, the operational speed profile is obtained by applying cruising with a speed lower than the maximum speed. The insertion of cruising phases at lower speeds is realized by means of a customised bisection algorithm. This identifies the speeds and the cruising phases that return a running time equal to the operational one provided by the timetable.

The input of this model is therefore the arrival/departure times and the operational running times planned in macroTT. The output are microscopic train speed profiles that satisfies the operational running time in microTT. In the following, we leave out the indices in order to keep the text easier to read.

We focus on computing an operational speed profile between two consecutive stopping points. In order to acquire the operational profile we use an *operational parameter* p [%], which represents the ratio between the given static speed limit and an actual speed that should be used to consume the given time supplement ψ. Lower and upper bounds for p are 30 and 100, respectively. Lower bound prevents that a train cruises at unacceptably low speed. For example, if the maximum speed is 130 km/h, the minimum allowed speed would be 39km/h. Upper bound gives the minimum running time. The operational parameter is applied on open-track in order to exploit the running time supplement, while maintaining the maximum speed through areas with restricted speeds (i.e., sections with the maximum speed of 40km/h). The running time with respect to the operational parameter is computed by applying the running time function (described in Section 4.1) for adjusted static speed limits over the infrastructure. If several microscopic points exist between two adjacent macroscopic timetable points like stops at the open-track, then for each train line p is a vector with different values between each two microscopic points.

The function uses an adjusted bisection algorithm to find an operational parameter p with a corresponding operational speed profile as described in detail in Algorithm 4. The focus here is on a single section between two microscopic timetable points. The function inputs are the scheduled running time $\tilde{r}$ from the microTT and the microscopic minimum running times r as well as a tolerated error $\varepsilon_{tolerance}$ [s], which is applied as a stopping criterion. The algorithm introduces the currently computed running time $r_{current}$ for the given operational parameter and the absolute computed error ε_{abs}, i.e.,

the difference between $\tilde{r}$ and $r_{current}$. Initially, $r_{current}$ is set equal to the minimum running time and p is set to $p_{ub} = 100$.

In each repetition, the algorithm:

1. Computes a speed profile (and running time) for value p
2. Refines the search range $[p_{lb}, p_{ub}]$ for p depending on the relation of $\tilde{r}$ and $r_{current}$
3. Updates p and ε_{abs}.

Steps 1-3 are repeated until the absolute error satisfies the stopping criterion.

Algorithm 4 Computation of operational speed profile

Input: Micro network, time supplements (from Step 1), $\varepsilon_{tolerance}$, train lines T
Output: Operational speed profiles for all train lines
Initialize $p_{lb} = 30$, $p_{ub} = 100$
Forall tuples (train line, running section, time supplement)
  Set bounds for operational parameter p, $[p_{lb}, p_{ub}]$
  Initialize $r_{current} \leftarrow r$, $\varepsilon_{abs} \leftarrow |r_{current} - r_{oper}|$, $p \leftarrow p_{ub}$
  While $\varepsilon_{abs} > \varepsilon_{tolerance}$
    $r_{current} \leftarrow$*RunningTimeComputation*(p)
    If $r_{oper} - r_{current} > 0$
      Update lower bound $p_{lb} \leftarrow p + \frac{p_{ub} - p_{lb}}{2}$
      Update operational parameter $p \leftarrow p_{lb}$
    Else
      Update upper bound $p_{ub} = p - \frac{p_{ub} - p_{lb}}{2}$
      Update operational parameter $p \leftarrow p_{ub}$
    End If
  Update error $\varepsilon_{abs} \leftarrow |r_{current} - r_{oper}|$
  End While
End Forall

Consequently, the blocking times are computed for all operational speed profiles and feasibility and stability of the *microTT* is evaluated applying the algorithms described in the next section.

4.4.3 Blocking times

A blocking time is the time interval that a given section (block section or track detection section) is exclusively allocated to a single train and therefore blocked for other trains. In railways it is not allowed for two trains to be contemporary in the same block section. Blocking times are computed according to the classical blocking time theory (Hansen & Pachl, 2014).

As can be seen in Figure 4, the blocking time of a train relative to a given block section is composed of the following components: setup time t_{setup} [s] to set the route for the

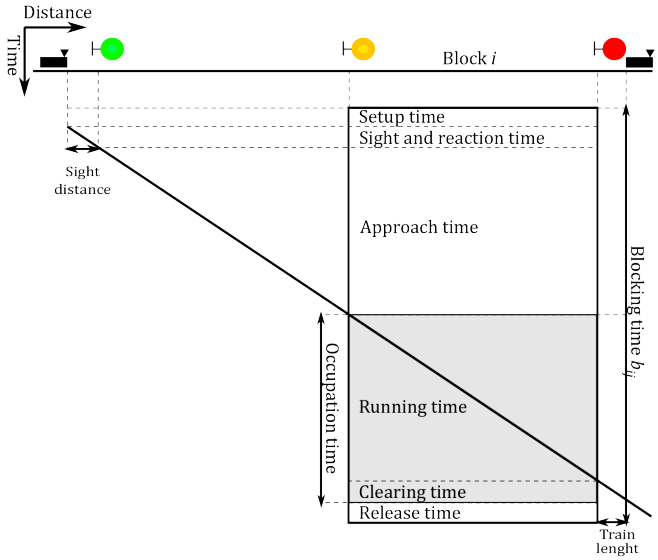

Figure 4.4: Blocking time stairway

approaching train; sight distance l_{sight} [m] or sight time t_{sight} [s] of the train driver when approaching the previous block section (approach signal); reaction time $t_{reaction}$ [s] of the driver, usually equal to 1.5 – 2 s; approach time $t_{approach}$ [s] needed by the train to cross the previous block section; running time t_{block} [s] of the train to cross the block section; clearing time t_{clear}[s] needed by the train to clear the block section over its train length; release time $t_{release}$ [s] needed to release the route after the train clearance. After having computed all these terms the blocking time d_{ij} of the train t relative to block i is obtained as:

$$d_{ti} = t_{setup,i} + t_{sight,ti} + t_{reaction,ti} + t_{approach,ti} + t_{block,ti} + t_{clear,ti} + t_{release,ti}.$$

The input to this function are the infrastructure characteristics and running times of trains. In particular, the operational running times, either from the initial iteration that include 5% of running time supplements or from macroTT, are used to produce the scheduled blocking time stairways. Note that the signalling system presented in Figure 4.4 represents a three-aspect two-block signalling system but different systems can be also modelled like four-aspect (UK signalling), the Dutch progressive speed signalling system, or the European Train Control System (ETCS) Level 1, 2 and 3.

Blocking times represent the main ingredient for the following functions, so we introduce it formally as $d_{ti} = (d^s_{ti}, d^e_{ti})$, where each blocking time d_{ti} of section i by train t

is specified from the start d^s_{ti} to the end d^e_{ti} of the blocking time. Each train t has an attributed list of blocking times $D_t = \{d_{t1}, d_{t2}, \ldots, d_{tn}\}$, where n is the number of track sections along the route ρ^{micro}_t.

4.4.4 Minimum headway time computation

A minimum headway time is the time separation between two trains at certain positions that enable conflict-free operation of trains (Hansen & Pachl, 2014)The minimum headway is computed based on the blocking times of each train for every macroscopic point, and for each pair of consecutive trains. In particular, for each pair of trains we calculate a set of minimum headways considering all the possible interactions between them such as both trains leaving a station, both trains entering a station or one entering and the other leaving.

We introduce the computation of the minimum headway at a timetable point $s \in S$. Let B_{ijs} be the set of blocks associated to conflicting routes (inbound or outbound) of train lines i and j in timetable point s, d^e_{il} be the end of blocking time d_{il} and d^s_{jl} the start of blocking time d_{jl}. Assume that both trains have the same reference event (i.e., departure, arrival or passing) time at s, e.g., equal to 0. Then the minimum headway h_{ijs} from train line i to j in timetable point s is computed as

$$h_{ijs} = \max_{l \in B_{ijs}} \left(d^e_{il} - d^s_{jl} \right). \tag{4.1}$$

4.4.5 Conflict detection and resolution (CDR)

The *CDR* model consists of two algorithms: conflict detection (CD) and conflict resolution (CR). The aim of the CDR is to verify the feasibility of the macroscopic timetable and to locally resolve potential conflicts by analysing the interaction between scheduled trains at the microscopic level. A track conflict occurs when two or more trains are scheduled to the same track section at overlapping periods of time. In other words, a track conflict is identified when the blocking times of two trains overlap fully or partially at a given track section. When a macroscopic timetable is available, we can test its feasibility at microscopic level using the CD procedure. This function takes as input the blocking time stairways produced for the operational running times. If there is an overlap between the blocking times of two different trains, this indicates a track conflict that must be solved. Specifically, track conflicts are solved by shifting trains in time until the blocking times do not overlap anymore. This shift initiates a change in the minimum headway between the trains. After all track conflicts have been detected, it is necessary to recompute the corresponding minimum headways. These new headways may be given to the macroscopic timetabling model to iteratively adjust the macroscopic timetable until no conflicts are detected anymore. Therefore, conflict-freeness is tested comparing the interaction of scheduled blocking times for each pair of trains, i.e., checking the possible blocking time overlaps between those two. The

blocking time overlap $c_{ij\varphi}$ from train line i to j at corridor φ is computed similarly as the minimum headway times as

$$c_{ij\varphi} = \max_{l \in B_\varphi} \left(d^e_{il} - d^s_{jl} \right), \tag{4.2}$$

where B_φ is the set of conflicting blocks at corridor φ. If $c_{ij\varphi} > 0$ then a conflict exists. Usually, a corridor corresponds to a macroscopic arc. In this way, the whole network is analysed by the conflict detection algorithm.

For the modelling purposes of CD we used a compact but efficient algorithm:

1. Sort the start and end times of the blocking time intervals over shared blocks.
2. Go through the sorted end times and build up the list of conflict pairs by looking at the preceding start time.

Algorithm 5 for CD is presented in the following. First, we initialise the set for observed conflicts Γ. The CD algorithm progresses through the list of track sections and for each $b \in B$ it generates the set D_b that includes blocking times of trains that traverse the b-th section. Then, D_b is sorted regarding the start and end times (d^s_{ti}, d^e_{ti}). For each pair of adjacent trains (t_i, t_{i+1}) the procedure checks the relation between the blocking time end of train t_i and blocking time start of train t_{i+1}, $d^e_t - d^s_{t+1}$. If this value is positive then a conflict exists. A conflict $\gamma \in \Gamma$ is described with a pair of conflicting trains t_1 and t_2, the corresponding track section b, and the total time in conflict, i.e., the overlap $\eta \leftarrow d^e_t - d^s_{t+1}$; formally, $\gamma = (b, t_1, t_{i+1}, \eta)$.

Algorithm 5 The conflict detection procedure

Input: set of track sections $b \in B$, set of blocking times $D_t \in D$
Output: set of conflicts Γ
Initialize $\Gamma := \emptyset$
Forall $b \in B$
    Create a set of trains T_b that use block b and
    corresponding blocking times D_b
    Sort D_b based on start of blocking times
    Create a pairing list of adjacent trains (t_i, t_{i+1})
    Foral pairs (t_i, t_{i+1})
        If $d^e_t - d^s_{t+1} > 0$
            $\eta \leftarrow d^e_t - d^s_{t+1}$
            Insert into Γ a conflict $\gamma = (b, t_1, t_{i+1}, \eta)$ between trains (t_i, t_{i+1})
        End If
    End Forall
End Forall

Once all the conflicts have been determined, the CR procedure described in the Algorithm 6 resolves existing conflicts between pairs of trains. The CR procedure 1)

computes the maximum overlap, 2) determines the associated headway (pair of trains and corresponding macro point) to be updated, and 3) updates the headway time for the maximum overlap. Recall that headways were defined for each macroscopic point, while a conflict may be located somewhere between two macro points. Therefore, we also need to choose the corresponding macro point to assign the updated headway.

Algorithm 6 The conflict resolution procedure

Input: tracks $a \in A$, conflicts $\gamma \in \Gamma$, $t \in T$, headways $h \in H$
Output: updated headway times H
Forall $a \in A$
  Forall pairs (t_1, t_{i+1})
    Step 1. Initialize a subset, Γ_{sub}, of conflicts that exist on arc a
    Step 2. Compute the maximum overlap for $\gamma \in \Gamma_{sub}$
    Step 3. Choose macro point s
    Step 4. Update $h_{t_1 t_{i+1} s} \leftarrow h_{t_1 t_{i+1} s} + c_{t_1 t_{i+1} a}$
  End Forall
End Forall

In the first step, the procedure determines the subset of conflicts $\Gamma_{sub} \subset \Gamma$ that corresponds to a pair of conflicting trains (t_1, t_2) at a given arc $a \in A$. Then, the maximum overlap $c_{t_1 t_2 a}$ is determined using (4.2). Step 3 finds the macroscopic point s for which the headway should be updated. This choice has been made based on the geographical distance between the track section with the maximum overlap and the surrounding macroscopic points, i.e., the closer point is selected. Finally, the relative headway $h_{t_1 t_{i+1} s}$ is increased by $c_{t_1 t_{i+1} a}$.

4.4.6 Capacity assessment

In this section, we define the idea of infrastructure capacity assessment. Our approach for capacity assessment is based on the timetable compression method, which is common practice. Timetable compression is the process of shifting train paths to each other as much as possible, bringing them to the (time) distance of minimum headway times. The total time needed for operating such a compressed timetable is the capacity occupation. Capacity assessment consists of determining capacity occupation and capacity occupation rate (share of used capacity expressed in %). We briefly introduce the max-plus automata theory and then apply it to compute capacity occupation. Note that in this section we use a common max-plus algebra notation that may differ from the rest of the paper. Our approach overcomes the current limitation of the UIC method and estimate the capacity for the station as a whole, and thus, includes all route dependencies in the station area. The capacity occupation $\mu(\varphi)$ of corridor φ can be obtained by:

$$\mu(\varphi) = \sum_{\{(i,j) \in W_\varphi\}} h_{ij\varphi} \ , \tag{4.3}$$

with W_φ the cyclic pattern of successive train pairs (i, j) in corridor φ, and $h_{ij\varphi}$ the

minimum line headway. The minimum line headway is computed similarly to a local minimum headway but with respect to all blocks on a corridor φ instead of a timetable point s. A corridor may be equal to a station area, an arc or comprise several adjacent arcs, $\varphi = \cup a_i$. We compute the capacity occupation for each corridor $\varphi \in \Phi$, applying an algorithm based on max-plus automata theory.

Basics of max-plus automata theory

Max-plus automata combines elements of the heaps-of-pieces theory and max-plus algebra and was introduced by Gaubert and Mairesse (1999). A max-plus algebra is a semiring over the union of real numbers and $\varepsilon = -\infty$, equipped with the two binary operations maximum ($\oplus$) and addition ($\otimes$). Let R_{max} be the set of real scalars and $-\infty$, then for $a, b \in R_{max}$ the operations are defined as:

$$a \oplus b = \max(a,b)\ , \quad a \otimes b = a + b.$$

The element $\varepsilon = -\infty$ is the neutral element for $\oplus$ and absorbing for $\otimes$. The element $e = 0$ is the neutral element for $\otimes$. Properties of max-plus algebra are similar to conventional algebra. We refer to Goverde (2007) for more details on max-plus algebra with application to railways.

A max-plus automaton H is a triple (Q, R, M), where:

- Q is a finite set of tasks, e.g., all possible train routes,
- R is a finite set of resources, e.g., block section or track detection section,
- M is a morphism $Q^* \rightarrow R_{max}^{|R|x|R|}$ which is uniquely specified by the finite family of $|R| \times |R|$-dimensional matrices $M(l)\ ,\ l \in Q$. Also, Q^* denotes a set of chosen train (partial) routes over a given corridor from Q, $Q^* \subset Q$.

We define a timetable as an ordered sequence of tasks, $w = l_1 \ldots l_n$. Therefore,

$$M(w) = M(l_1 \ldots l_n) = M(l_1) \otimes \cdots \otimes M(l_n).$$

A task is called an elementary task if R-dimensional row vectors $s(l)$ and $f(l)$ exist such that $s(l) \leq f(l)$ and

$$M_{ij}(l) = \begin{cases} e, & \text{if } i = j,\ i \neq R(l), \\ f_j(l) - s_i(l), & \text{if } i, j \in R(l), \\ \varepsilon, & \text{otherwise.} \end{cases} \tag{4.4}$$

Variables $s(l)$ and $f(l)$ represent the start and end time of task l, respectively. In the railway terms, task l is a (partial) route of a train line, while $s(l)$ and $f(l)$ correspond to occupation and release times of the i-th block, d_i^s and d_i^e, respectively.

Route	$s(r)$	$f(r)$
a	$[0,\varepsilon,15,25]$	$[25,\varepsilon,35,50]$
b	$[25,15,\varepsilon,0]$	$[50,35,\varepsilon,25]$
c	$[0,\varepsilon,20,90]$	$[30,\varepsilon,100,120]$

The *upper contour* $x(w)$ of a schedule w is defined as

$$x(w) = M(w) \otimes x(e),$$

where $x(e)$ is an R-dimensional vector corresponding to an empty schedule. A more extensive description of max-plus automata theory is given by Gaubert and Mairesse (1999) and Egmond (2000).

Application of max-plus automata to capacity occupation

The capacity occupation $\mu(w)$ of the schedule w is computed as

$$\mu(w) = \min\left(x(wa) - (f(a) - s(a))\right), \tag{4.5}$$

where schedule wa is a schedule for one cycle w and the first train service a that belongs to the next cycle, and $f(a) - s(a)$ is the blocking time stairway of the repeated train service a over all resources. This formulation corresponds to the Equation (4.3). The capacity occupation rate $C(w)$ is defined as $C(w) = \frac{\mu(w)}{P} \cdot 100\,[\%]$, where P is the scheduled cycle period.

Let us summarize the capacity occupation model. First, we define a set of arbitrary railway sections ϕ. A section $\varphi \in \phi$ may represent a corridor or a station (i.e., macroscopic timetable point). A corridor is bounded by a pair of macroscopic timetable points, e.g., $\varphi = (s_1, s_2, \ldots s_n)$. A station is treated similarly by accepting $\varphi = s$.Then, we determine a subset Q^* of train routes that are selected for train lines over section φ. Finally, the model computes the capacity occupation for each $\varphi \in \phi$ by using (6.6) and is represented with $\mu(\varphi)$.

Numerical example

Let us consider the following example for computing the capacity occupation in a station. Consider three trains a, b, c, schedule $w = abc$ and resources $r = 1, .., 4$, as in Figure 4.5a. Train route a uses resources $[1,3,4]$, b uses $[4,2,1]$ and c uses $[1,3,4]$. The train blocking times are given as

Note that ε represents an unused resource. The corresponding matrices M for routes a, b and c are defined using Equation (4.4) as follows:

$$M(a) = \begin{bmatrix} 25 & \varepsilon & 35 & 50 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 10 & \varepsilon & 20 & 35 \\ 0 & \varepsilon & 10 & 25 \end{bmatrix} \quad M(b) = \begin{bmatrix} 25 & 35 & \varepsilon & 0 \\ 35 & 20 & \varepsilon & 10 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 50 & 35 & \varepsilon & 25 \end{bmatrix},$$

$$M(c) = \begin{bmatrix} 30 & \varepsilon & 100 & 120 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 10 & \varepsilon & 80 & 100 \\ -60 & \varepsilon & 10 & 30 \end{bmatrix}.$$

The matrix M for schedule ab is computed as:

$$M(ab) = M(a) \otimes M(b) = \begin{bmatrix} 100 & 85 & 35 & 75 \\ 35 & 20 & \varepsilon & 10 \\ 85 & 70 & 20 & 35 \\ 75 & 60 & 10 & 25 \end{bmatrix}.$$

Similarly, train c is added to the schedule in the same manner, i.e., $M(abc) = M(ab) \otimes M(c)$. The upper contour of the schedule $abca$ is then computed as

$$x(abca) = M(abca) \otimes x(e) = \begin{bmatrix} 220 \\ 85 \\ 230 \\ 245 \end{bmatrix}.$$

The capacity occupation for the scheduled services abc is then computed as:

$$\mu(abc) = \min\left(x(abca) - (f(a) - s(a))\right) = \min\left(\begin{bmatrix} 220 \\ 85 \\ 230 \\ 245 \end{bmatrix} - \begin{bmatrix} 25 \\ \varepsilon \\ 20 \\ 25 \end{bmatrix}\right) = 195.$$

Note that $85 - (-\infty) = +\infty$. If the cycle period equals $P = 600s$, then the capacity occupation rate is

$$C(abc) = \frac{\mu(abc)}{P} \cdot 100\ [\%] = \frac{195}{600} \cdot 100\ [\%] = 32.5\ [\%].$$

Figure 4.5b provides a graphical representation of the compressed schedule $w = abc$. The coloured blocks represent the train occupation of the infrastructure, with one train movement depicted by the same colour. Note that train a is added twice in order to determine the earliest possible departure of a train from the following period. The red line represents the capacity occupation for schedule w, $C(w)$. The white space (between the x-axis and red line) depicts unused capacity which might be used to add extra trains.

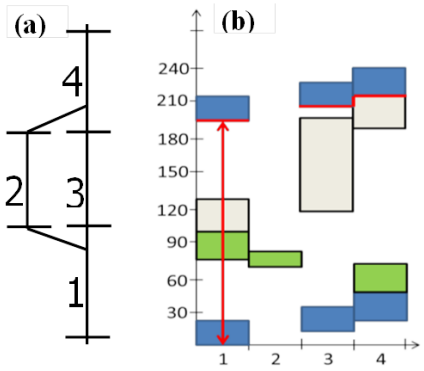

Figure 4.5: (a) Example infrastructure and (b) capacity occupation for schedule *abc*

4.5 Case study

In the case study we focus on two elements. First, we show the applicability of each function within the microscopic model. Besides that, we demonstrate the developed timetabling framework with all functionalities of the microscopic module applied to a real railway network. We apply the macroscopic model from Bešinović et al. (2016). However, any other macroscopic model could be used (e.g., Siebert and Goerigk (2013)).

We consider a real-life instance for train line services on the 80km long corridor Utrecht (Ut)-Den Bosch (Ht)-Eindhoven (Ehv) (Figure 6.6), a highly utilised part of the railway network in the central Netherlands. The values present number of tracks in stations or junctions and lines between depict number of track between two timetable points. The microscopic infrastructure includes various topology – double, triple and quadruple tracks. The microscopic graph M for the considered corridor includes around 1000 nodes and 1500 microscopic arcs considering infrastructure details like location of signals, switches, train detection points, the speed limits, slope gradients and curves. For running time computations, a detailed train dynamics have been modelled. The network included 13 microscopic timetable points such as stations, stops, junctions and bridges.

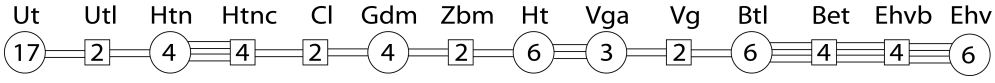

Figure 4.6: Case study infrastructure with macroscopic (circles) and microscopic (squares) timetable points

The original timetable on this network is periodic with half an hour pattern composed of 20 train lines, of which 12 are Intercity (IC) and eight are regional trains. Train lines originate and terminate at different stations along the corridor and have different stopping patterns. Regional trains stop at all stations, while ICs stop at limited stations.

4.5.1 Functionality of the microscopic model

We start by computing the minimum running times and the corresponding headway times, constructing the macroscopic network and aggregating the process times (Algorithm 2). Solving the equations for running time is performed over distance with computational accuracy set to 10^{-5} m, while $\varepsilon_{abs} = 1$ s. The accuracy of other models is one second. The average computation (CPU) time for the minimum speed profile for one train line was one second, while for the operational speed profile was four seconds. Generating macroscopic network resulted in seven macroscopic timetable points (important stations and junctions) and six macroscopic arcs. A total of 1000 headway times was computed in eight seconds. In the later iterations, a limited number of headways is usually updated, so the CPU time then is well under one second. The CPU times for conflict detection for the whole network and capacity assessment per corridor (or station) are on average three and one second, respectively. Lastly, network transformations, micro to macro and vice versa, take under one second as well. For testing purposes, we applied a macroscopic timetable model as in Bešinović et al. (2016) to generate a macroTT. Once a macroTT is obtained, the microscopic models evaluated its feasibility and stability. First, a microTT is generated by identifying the operational train speed profiles corresponding to the scheduled running times (Algorithm 3 and 4). The output of the Algorithm 4 for one train line is illustrated in Figure 4.7 and depicts the distance-speed diagram for the local train 6000 (blue dotted line) running over the corridor Ht-Ut. Such a speed profile corresponds to the scheduled running time where time supplements are exploited by cruising at a speed lower than the time-optimal speed profile, i.e., computed for the minimum running time (red solid line). The circles represent line-side signals, black solid line are gradients, black dashed line is the static speed limit.

The newly produced blocking times are used in the CDR model to detect possible conflicts between trains. The corridor included 600 track sections. Figure 4.8 gives the (partial) output of the blocking time computation for the different train services operating between Gdm and Ut. The diagram shows only the infrastructure that train 6000 uses, in order to clearly visualise actual conflicts between trains. The red box depicts a conflict of train services 6000 and 3500 between Utrecht Lunetten (Utl) and Ut. The minimum headway $h^{dd}_{6000,3500,Ut}$ between these two trains originally was 150 s while the maximum overlap of their conflicting blocking times (three in total) is $\max(48, 38, 38) = 48$ s. The track conflict is therefore resolved by shifting the train over an extent equal to the overlap. In this case, the minimum headway increases by 48 s, resulting in a new headway time $h^{dd}_{6000,3500,Ut} = 198$ s, so that the blocking times are touching but not overlapping. This new headway is sent to the macroscopic model together with the other updated headways and running times, for reproducing a new macroscopic timetable.

The capacity occupation for a given microTT is computed by applying the max-plus automata method. The capacity occupation for all corridors and stations is given in Tables 6.1 and 6.7.1, respectively. In addition, the last column in both tables shows

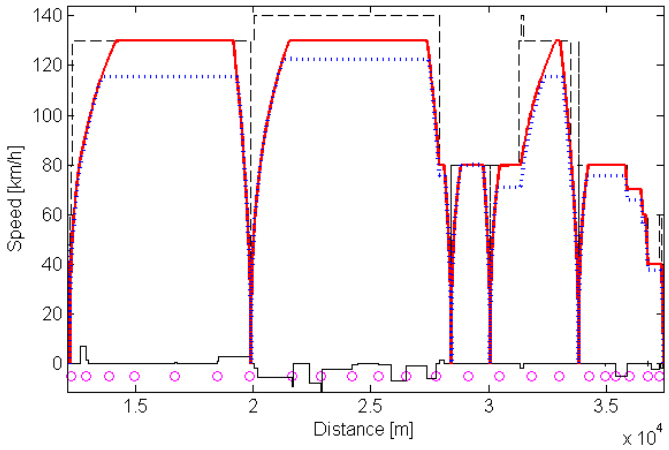

Figure 4.7: Train speed profiles for minimum running time (red solid line) and scheduled time supplements (blue dotted line). The maximum speed of the train is 130 km/h.

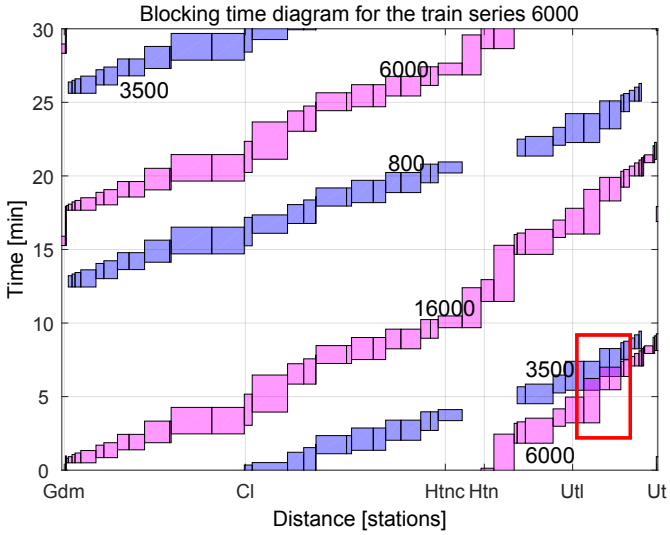

Figure 4.8: Blocking time diagram the corridor Gdm–Ut

the total number of resources used by all routes, which defines the size of matrix M (cf. Equation (5)), and thus the complexity of the computation. We describe here the capacity occupation for station Ht, which consists of six station tracks including four platform tracks. Fourteen trains operate each 30 minutes through Ht, which use in total 69 different infrastructure resources. Figure 4.9 shows the station layout and the output of the capacity assessment. The x-axis reports all the track detection sections belonging to the station. Note that their sequence does not follow a topological order. The y-axis denotes time, and the blocks show for each track detection section when they are used by a train service. The different colours of the blocking times correspond to distinct train routes through a station. In red we highlight the first train service of the next timetable period. We found that the capacity occupation time of station Ht is 1539 s (25.6 minutes) and the rate is 42.8% in a timetable period of 60 minutes. This means

Table 4.1: Capacity occupation at corridors

Corridor	Time (s)	Rate (%)	No of resources
Ut-Ht	1892	52.6	110
Ht-Ut	1924	53.4	104
Ehv-Ht	1320	36.7	90
Ht-Ehv	1372	38.1	91

Table 4.2: Capacity occupation at stations

Station	Time (s)	Rate (%)	No of resources
Btl	870	24.2	85
Ehv	930	25.8	37
Gdm	954	26.5	48
Ht	1539	42.8	69
Htn	900	25.0	24
Ut	844	23.4	58
Vga	934	25.9	34

that the timetable locally contains 2061s (57.2%) of time allowances. By comparing these values with those suggested by the UIC 406 Code, i.e., a minimum of 50%, it is concluded that Ht has an acceptable amount of time allowances, and therefore satisfies the stability norms.

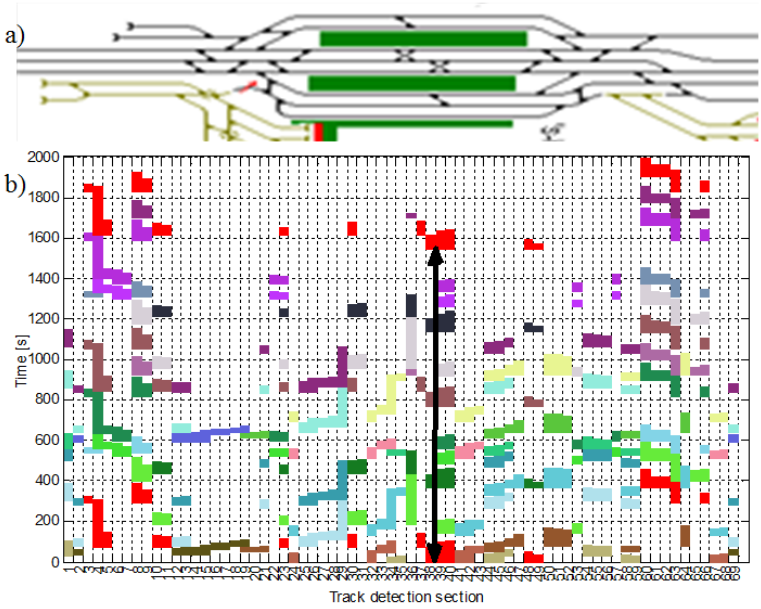

Figure 4.9: Station Den Bosch: (a) station layout and (b) capacity occupation

4.5.2 Testing the developed framework

In order to show the suitability of the microscopic models within the developed framework, we used the macroscopic timetabling model described in Bešinović et al. (2016).

Table 4.3: Characteristics of the macroscopic timetable after each iteration

Iteration	No of conflicting train pairs	Overlap time (s)
1	6	160
2	4	130
3	3	98
4	5	110
5	3	65
6	1	8
7	0	0

We present the computational results and the computed timetable, including the achieved values for the performance measures, i.e., feasibility and stability.

Table 4.3 presents the microscopic conflicts in the macroscopic timetable at the end of each iteration. The number of conflicting train pairs equals the number of headways that has been updated at the microscopic level. Overlap time is the sum over all maximum conflicts between two trains $\sum c_{t_1 t_2 a}$. In the first iteration, there are six conflicts that add up to 160 seconds of overlapping blocking times. In the second iteration, only four conflicts remain with a total overlap time of 130 seconds. In the subsequent iterations, all conflicts are resolved. It can be seen from the table that the approach can solve all conflicts successfully within several iterations, gradually reducing the number and size of total overlaps. However, resolving conflicts in one iteration may produce some new conflicts in the following iteration. But the algorithm converges to a timetable which is completely feasible both macroscopically and microscopically. The observed computation time for obtaining the feasible and stable timetable was about 14 minutes, with on average 2 minutes per iteration.

Figure 4.10 shows a time-distance diagram of the computed hourly timetable for the corridor Ut-Ehv. The vertical axis shows time in minutes downwards. The horizontal axis shows distance with the station positions indicated. The blue lines are IC trains, the magenta lines are local trains. Note that the sections Btl-Ehv and Htn-Htnc have four tracks where trains may cross each other. Figure 4.11 shows the corresponding blocking time diagram for the route of intercity train line 3500. Note that only the blocking times are shown for the trains running on the same tracks as train line 3500. The gaps in the blocking time stairways for some trains correspond to running on parallel tracks in stations or the four-track lines between Htn-Htnc and Btl-Ehv. Around Ht also some blocking times are visible corresponding to crossing trains from/to different corridors.

The optimized timetable shows periodic passenger trains with regular 15 minute services of both IC and local trains where two similar train lines follow the same route. Hence, effectively 15 min train services are realized instead of two separate 30 min train lines.

The blocking time diagram shows no overlapping blocking times and hence asserts

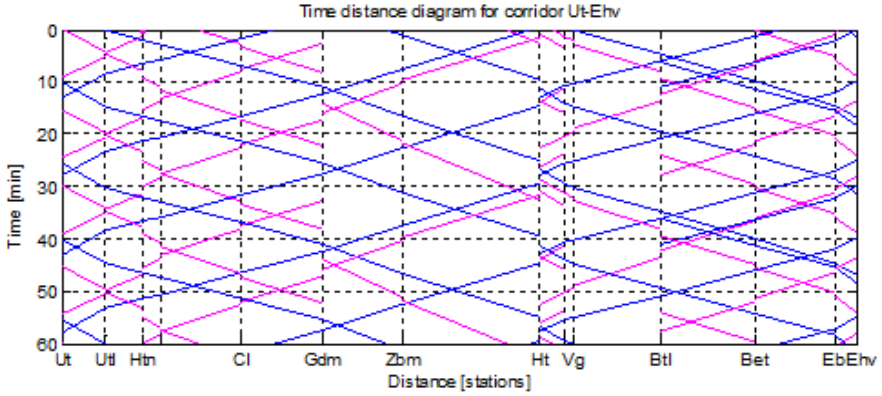

Figure 4.10: Time-distance diagram for corridor Ut–Ehv

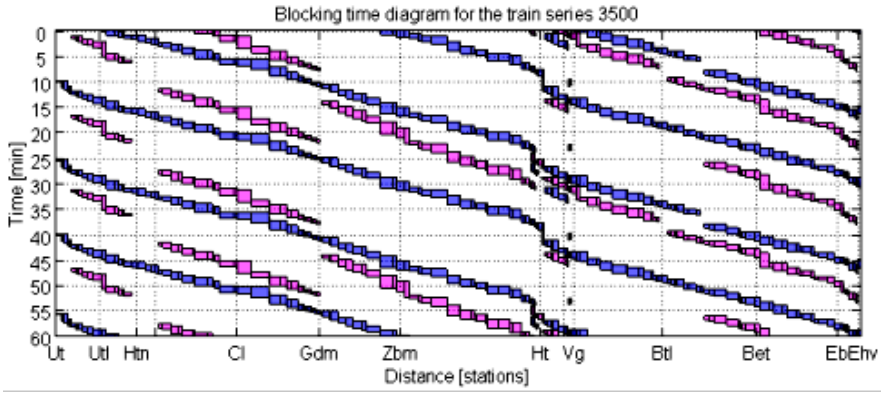

Figure 4.11: Blocking time diagram for corridor Ut–Ehv

that the timetable is conflict-free. Moreover, the timetable is robust which is illustrated by the buffer times (white space) between the train paths.

Finally, the obtained capacity occupation rates are below the recommended stability values of 65% for mixed traffic corridors in daily periods and 50% for stations defined by the UIC, which were the constraints of the timetabling algorithms. Corridor Ut-Ht is the heaviest used with the capacity occupation rate of 57.8%. Therefore, we may conclude that the produced timetable is also stable.

4.6 Practical reflection of the developed microscopic model

The developed framework has been evaluated by experts from the infrastructure managers Network Rail (UK) and Trafikverket (Sweden). Here we give a summary regarding the functionality of our microscopic model. The applied time precision of one second is highly appreciated, as it leads to minimizing the unused capacity and unrealizable running times. Also, its relevance is supported by the current efforts in this direction in the UK. They emphasized the ability of the model to compute highly detailed running and blocking times taking into account all route details at the track section level (speed restrictions, signalling, gradients). The implementation of the new

conflict detection and resolution algorithms that accurately assess the timetable feasibility gives valuable transparency to timetable planners. The importance of capacity occupation and stability norms was also stressed, but they also pointed out the need to standardize and configure the norms to reflect local (national) capacity standards. The overall comment is that 'The implemented functionality to timetable planning was reviewed as highly valuable and an advance on current practice'. This is also confirmed by the infrastructure manager ProRail and main railway undertaking NS from the Netherlands. The microscopic models are currently applied in a pilot project by ProRail and NS to evaluating the Dutch timetables at the national network level.

4.7 Conclusion

In this paper we have provided a methodology and new microscopic models for supporting the timetable design as well as the network and data transformations to manage communications between microscopic and macroscopic models. The main focus was on the microscopic models for computing reliable running and minimum headway times for the macroscopic model, as well as analysing the feasibility and stability of the macroscopic timetables at the microscopic level. Operational running times are calculated by integrating the Newton's motion formula and a fast bisection model that introduces cruising phases at lower speeds to cover the supplement times imposed by the timetable. Accurate headway computation is based on the blocking time theory. In this way, we could generate train process times in short time, even for very dense railway traffic. The timetable feasibility was checked by an efficient conflict detection model based on the blocking time theory, which automatically recomputes new minimum headway times if a conflict arise. The capacity assessment is realized by the new application of max-plus automata following the compression method. Our method allowed computing capacity occupation in stations as well as at corridors. If the capacity occupation rate satisfies technical thresholds the timetable is considered as stable.

The microscopic models were also integrated in an innovative timetabling framework to develop timetables that are operationally feasible and stable. The framework is completely general and based on the iterative interaction among macroscopic and microscopic models. Due to its modular development, the macroscopic model can be any optimization model for timetable computation.

We applied the microscopic models at a real Dutch railway network. Small computation times make us confident that the models could also be used on more complex instances, although no microscopic infrastructure data for other networks is available to us yet. In practice, microscopic models could be interfaced with existing timetabling tools to rapidly obtain good quality timetables that are also conflict-free and able to effectively absorb delays. We believe that presented microscopic models have a great potential for improving real-life applications for railway planning.

Chapter 5

A two-stage stability-to-robustness approach to robust railway timetabling

---

This chapter has been submitted for publication as:

Bešinović, N. & Goverde, R. M. P. A two-stage stability-to-robustness approach to robust railway timetabling, submitted.

5.1 Introduction

The constant growth of railway transport demand on one hand and the limiting existing infrastructure capacity on the other force the railways to constantly improve their processes and rise performance in managing existing and planned resources. This paper describes a timetable planning model that will support timetable designers to achieve more robust timetables that are less sensitive to delays. The traditional Train Timetabling Problem (TTP) aims at finding a train schedule on a railway network that satisfies operational constraints and maximizes the efficiency of the transportation services. Timetable efficiency is interpreted as providing fast services which assure a competitive offer to passengers. Due to the growing demand, the need for using the capacity in an optimal way becomes greatly important, and so, improving capacity utilization directly improves timetable stability. A timetable also has to handle daily stochastic disturbances within the network. Thus, a robust timetable is capable of coping with stochastic process time variations. Therefore, we focus on finding an efficient, stable and robust periodic timetable and define the problem as the Stable and Robust TTP (SR-TTP). The questions that arise in the timetable design are: 1. *How to use the infrastructure in the optimal way?* 2. *Where to insert time allowances in order to*

guarantee a good trade-off between an efficient and a robust TTP solution? 3. *What is the best objective function that generates the best timetable solutions?*

We distinguish between periodic and non-periodic TTPs, where the former introduces a certain cycle time and a given timetable repeats every cycle. A periodic TTP is commonly modeled as *a periodic event scheduling problem* (PESP), which was introduced by Serafini and Ukovich (1989). PESP is originally a feasibility problem which was used by Schrijver and Steenbeek (1994) to design railway timetables. Periodic TTPs are often modeled as mixed integer linear programs (MILP) by adopting an objective function as in Peeters (2003).

In this paper, we adopt a common way to tackle SR-TTP, that is, by introducing so-called *time allowances* in the planning phase. Time allowances are additional times allocated to the train- and passenger-related processes on top of the minimum process times, called time supplements, or between processes to reduce the time dependency between them, called buffer times. First, passengers want to travel fast so the goal would be to have as minimum time supplements as possible. On the other hand, adding some time supplements would increase trains running on time. In addition, adding buffer times reduce possible delay propagation between trains. So, the interplay of time allowances has a significant impact on the timetable quality.

We propose a two-stage model for finding stable and robust solutions to the periodic TTP. The model is based on PESP and introduces the concept of *stability-to-robustness*. The first stage aims at finding an optimal stable timetable structure that minimizes both capacity utilization and journey times. To do so, a model for minimizing the cycle time is developed. In the second stage, the objective is to find the optimal allocation of time allowances so as to maximize the robustness for a given timetable structure. To this aim, several objective functions are proposed. Additionally, we use a delay propagation model to evaluate the obtained timetable.

Benefits of our two-stage model for solving SR-TTP are summarized as follows. The stability-to-robustness model is the first macroscopic model that introduces stability together with efficiency and robustness for periodic TTP. Second, a transformation between the minimal cycle time optimization model and the cycle periodicity formulation is proposed. Third, we introduce new objective functions that build on the intermediate outcome of the model's first stage to provide higher quality solutions. Fourth, more promising cycle bases have been detected for our model. Fifth, we also showed that a thorough analysis of model weight factors should be considered to generate the best trade-off between efficiency and robustness. What is more, certain objective functions tend to allow more flexibility that lead to generating significantly different solutions. Finally, computational times were significantly smaller, which suggests that our two-stage model may be successfully used on bigger instances.

The remainder of the paper is as follows. Section 5.2 reviews existing timetabling and capacity estimation models. Section 5.3 describes the basic model for periodic timetabling. In Section 5.4, we define the two-stage formulation for robust periodic

TTP and give the connecting elements between the two optimization stages. In addition, several objective functions are proposed and the robustness evaluation model is explained. Section 5.5 presents computational results on real-life instances in the Netherlands. Finally, conclusions are given in Section 5.6.

5.2 Literature review

A timetable consists of event times such as arrival and departure times in stations and processes between events like running, dwelling and transfer times. Process times also include infrastructure constraints (i.e., headways) between events that guarantee safe operations. In periodic timetabling, a cycle time T is given and all events are selected in the interval between 0 and T.

For solving PESP, Schrijver and Steenbeek (1994) applied constraint programming to find a feasible timetable, while Kümmling, Großmann, Nachtigall, Opitz, and Weiß (2015) used SAT solvers to the same problem. In addition, local search heuristics have been applied to improve timetable quality by using modified simplex modulo approaches like in Nachtigall and Opitz (2008) and Goerigk and Schöbel (2013). By adding an objective function to the PESP formulation, the TTP can be solved using mixed integer programming (MIP) techniques (Peeters, 2003). Kroon and Peeters (2003) introduced a variable trip time model that considered lower and upper bounds on running and dwell times instead of so far fixed ones. Liebchen (2008) presented the first optimized railway timetable based on a PESP formulation in practice implemented at Berlin Metro. The goal of the model was to minimize passenger waiting times and reduces the number of rolling stock. Kroon, Peeters, Wagenaar, and Zuidwijk (2013) extended the PESP formulation with flexible connections like passenger and rolling stock dependencies. Borndörfer, Hoppmann, and Karbstein (2016b) introduced a new pseudo-polynomial time separation algorithm for cycle inequalities. Pätzold and Schöbel (2016) defined matching-based heuristic for solving PESP. In addition, a periodic TTP represents a strongly NP-hard problem (Nachtigall, 1993) while a robust version is even harder to solve (Kroon et al., 2008). For integrating passengers in periodic timetabling problems we refer to M. Schmidt (2014), M. Schmidt and Schöbel (2015), Robenek, Maknoon, Azadeh, Chen, and Bierlaire (2016), Borndörfer, Hoppmann, and Karbstein (2016a) and Gattermann, Großmann, Nachtigall, and Schöbel (2016).

A timetable includes multiple performance indicators that has to be considered in the design phase. Goverde and Hansen (2013) defined indicators like timetable efficiency, stability and robustness. Timetable efficiency reflects the amount of time supplements in the scheduled travel times (running, dwell and transfer times) which must be as small as possible to provide short journey times and seamless connections. Timetable stability is the ability of a timetable to absorb initial and primary delays, so that delayed trains return to their scheduled train paths without rescheduling. According to Goverde (2007) and Heidergott et al. (2014), network-level stability of a periodic timetable can

be expressed as the minimum cycle time λ of the timetable. Thus, the timetable is stable when the minimum cycle time is smaller then the given timetable cycle time T. Even more, a timetable is more stable when the minimum cycle time is as small as possible. The higher λ is, the smaller are the time allowances and hence the less stable is the timetable. Timetable robustness is the ability of a timetable to withstand design errors, parameter variations, and changing operational conditions. In the existing literature, efficiency has been most often considered and such models are referred as to nominal TTP (Cacchiani & Toth, 2012). Robustness attracts increasing research, while stability has been considered in just a few articles. Finally, all three indicators have been considered only in Bešinović et al. (2016) and Goverde et al. (2016), whom proposed an integrated iterative approach between microscopic and macroscopic models.

Bešinović et al. (2016) proposed a micro-macro approach that includes a macroscopic timetabling model (with fixed minimum headways) and microscopic models that check the feasibility and the stability of the timetable generated by the macro model. Due to extending running times at the macroscopic level, some original minimum headways could have become insufficient to guarantee a conflict-free run of trains. Then, for such cases, the minimum headways are recomputed and the macroscopic model is run again. In addition the microscopic models evaluated stability (as capacity occupation) and proposed adaptations to process times for macroscopic model. The micro and macro models communicate iteratively until no conflicts are found and the stability is under required stability margins. Goverde et al. (2016) introduced a general performance-based framework that also integrates models for ex-post improving energy-efficiency of timetables.

To answer the first question of better infrastructure use, we consider the timetable stability in the timetable planning. Stability is often presented as an outcome of the infrastructure capacity occupation; however, it has only been used for evaluating timetables. The idea of minimizing the cycle time for measuring stability was firstly introduced by Bergmann (1975) for a single-track line. He proposed a mixed integer formulation of a simplified periodic scheduling problem for a synthetic case study that considers a single-track railway line with all stations equipped with sidings and homogeneous fleet. Heydar, Petering, and Bergmann (2013) extended this model to tackle a single track, unidirectional rail line that adheres to a cyclic timetable and considered two types of trains. The objective was to minimize the capacity occupation and minimize the total dwelling time of local trains at all stations. Sparing and Goverde (2013) and Sparing (2016) developed an extension to the PESP model that minimizes the cycle time and train running times which is applicable to both lines and networks. They tested the model on a part of the Dutch railway network. Zhang and Nie (2016) further expanded Sparing and Goverde (2013) by adding flexible overtaking constraints and heuristics to speed up the computations. In addition, authors analysed the effect of some timetable design parameters on the minimum cycle time. Petering, Heydar, and Bergmann (2015) extended the model of Heydar et al. (2013) to allow selection of

platform tracks in a station and schedule train overtakings. In our research, we extend the approach of Sparing and Goverde (2013) and define a model for stable and robust timetabling.

For solving a robust periodic TTP several approaches have been found in the literature such as stochastic programming, convex programming, recovery-to-optimality, half-buffers (Peeters, 2003), delay resistance, passenger robustness, and flexible PESP (FPESP). Kroon et al. (2008) used stochastic programming for modeling and solving a robust periodic TTP. The model was used to modify a given periodic timetable to construct an improved timetable by minor adjustments of event times. In particular, it was used to redistribute time allowances to the processes in the original timetable so that the average delay of the realizations of the timetables of the trains is minimized. The authors described a stochastic optimization variant of PESP that incorporates stochastic disturbances of the railway processes. The model is composed of two parts: a timetabling part for determining the timetable (which shows many similarities with PESP) and a simulation part for evaluating the robustness of the timetable under construction. In order to keep the computation times at an acceptable level, the model fixes the precedence constraints. Indeed, the aim is to keep the structure of the timetable as in the input timetable and to optimally reallocate the time allowances, so that the resulting timetable is more robust This formulation for robust periodic TTP lead to large models which are time consuming to solve and thus, the model's application has been limited to small networks. More recently, Maróti (2016) proposed a more efficient model for solving the same problem as in Kroon et al. (2008) by applying convex programming and was able to obtain results for the complete Dutch network within one minute.

Goerigk and Schöbel (2014) initially proposed a recovery-to-optimality concept for improving robustness of non-periodic timetables. Their model uses predefined input scenarios of disturbances U (called uncertainty set) and the original timetable is being adjusted according to those. Goerigk (2015) extended an integer programming model for periodic timetabling and gave a bi-criteria local search algorithm for large-scale instances. His work is based on the concept of recovery-to-optimality. The model integrates two stages, it first computes an optimal solution for each disturbance scenario $l \in U$, and then solves a location problem to find an optimal solution for any scenario. The author considered two minimization versions of the location problem, the maximum distance, which is equal to the center location problem, and sum of distances, the median location problem. As concluded in the paper, this approach becomes easily computationally extensive and thus, inapplicable for real-life instances. Thus, a heuristic approach is applied, which iteratively performs a local search for minimizing travel time (maximizing efficiency) or maximizing robustness, depending whether a minimum robustness rate is satisfied or not. The best solution is given after a given number of iterations have been performed.

Peeters (2003) proposed an objective function for the PESP-based model that forces buffer times to the middle of their lower and upper bound ('half-buffer'). A on periodic

TTP, is limited to a given time period (cycle time). This would mean that having too much buffer time between two periodic events in one period, $i(T_1)$ and $j(T_1)$ in period T_1, may result in too little buffer between $j(T_1)$ and i from the following period, $i(T_2)$. He defined an additional variable that tracks the deviation from $(u-l)/2$ for all process times and minimize the sum of such values.

Liebchen, Schachtebeck, Schöbel, Stiller, and Prigge (2010) introduced a model for improving delay resistance of railway timetables. They assessed the delay resistance of a timetable by evaluating it subject to several delay scenarios to which optimum delay management was applied. The model was tested on real-world data of a part of the German railway network and the computational results suggested that a significant decrease of passenger delays can be obtained at a relatively small price of robustness, i.e. by increasing the train travel times.

Sels et al. (2016) developed a PESP-based model for designing robust timetables while minimizing the total expected passenger travel time. This model has been tested on the complete Belgian railway network and provided a reduced waiting time of 3.8 % compared to the existing timetable. In addition, a timetable evaluation showed improved punctuality.

Caimi, Fuchsberger, Laumanns, and Schüpbach (2011) proposed an extension of the PESP, the flexible periodic event scheduling problem (FPESP), in which intervals are used instead of fixed event times. These intervals may be considered as a robustness improvement. By applying FPESP, the output does not define a final timetable but an input for finding a feasible timetable on a microscopic level, which may be used to increase timetable robustness.

For non-periodic variants of the robust TTP, a few other approaches have been proposed in the literature like light robustness (Fischetti & Monaci, 2009) or recovery robustness (Liebchen et al., 2009) and heuristics based on the Lagrangian relaxation (Cacchiani et al., 2012). A comprehensive review of nominal and robust versions of TTPs is given in Cacchiani and Toth (2012). Note that all robust TTP are more complex and therefore more computationally extensive compared to their nominal TPP counterparts. Another stream of non-periodic timetabling includes modeling of passenger demand and computing the optimal timetables that satisfy given demand. Examples of such models are Barrena, Canca, Coelho, and Laporte (2014a); Barrena et al. (2014b).

In this paper, we give several contributions compared to the existing work on robust periodic timetabling. First, we incorporate stability and robustness in a macroscopic model for periodic TTP for the first time by introducing the two-stage stability-to-robustness approach. Second, differently from models for improving robustness of an existing timetable Goerigk (2015); Kroon et al. (2008); Maróti (2016), we focus on developing a new timetable. Third, stability-related models like Petering et al. (2015) and Heydar et al. (2013) do not generate a final timetable, but only a compressed one, and thus, these are considered as capacity assessment models rather than timetabling models. In addition, we use our model also for networks, while models in Heydar

et al. (2013) and Petering et al. (2015) were developed only for lines. Sparing and Goverde (2013) reached a final timetable just by proportionally extending all process times. Fourth, in order to successfully link stability and robustness we develop a connection between the two stages. Fifth, once the train orders are known, we are able to define robustness more explicitly then in Peeters (2003) as we determine the neighbouring arcs to include in the objective function. Differently from Kroon et al. (2008); Maróti (2016) and Goerigk (2015) we maximize only buffers, while minimizing time supplements. In this way, we get better suited timetables to current driver behaviors.

5.3 Problem description

Here we give the terminology used throughout the paper and define the problem formulation for solving a periodic SR-TTP. The macroscopic timetabling approach is based on a *periodic event-activity network* represented by a directed graph $N = (E, A, T)$, which is associated with a set of train lines Q. A *train line* $q \in Q$ defines a requested periodic train service characterized by its origin and destination, stopping pattern, and frequency within a given common timetable period T, referred as to *scheduled cycle time*. In the periodic event-activity network $N = (E, A, T)$, the set of events E consists of periodic arrival, departure and pass-through events for each train line in Q in each station along its route. This means that if an event i is scheduled at time π_i then it will also occur at times $\pi_i + k \cdot T$ for $k = 1, 2, \ldots$. Therefore, for each event we determine the event time in the basic period $\pi_i \in [0, T)$. Each train line $q \in Q$ consists of a sequence of process times $a = (i, j)$, where i and j are two consecutive events.

Set A represents process times $(i, j) \in A$, where i and j are two consecutive events and it can interpret various rules and restrictions. *Running times* are the times needed for a train to run between two timetabling points. A lower bound l_{ij} for the running time represents the nominal running time, which is the minimum running time increased by a certain percentage to satisfy stochastic train behavior. The upper bound u_{ij} is the maximum running time extension with respect to the passenger quality of service. The set of running processes is denoted as A_{run}. *Dwell times* are the durations of a train stop in a station. The minimum represent a time needed to board and alight the train, while the upper bound u_{ij} limits the waiting time for passengers. The set of dwell processes is denoted as A_{dwell}. A *passenger connection* is a transfer of passengers from a feeder to a connecting train in a station. The minimum transfer time l defines the necessary time to alight from the first train, walk to the departure platform, and board the second train. A set of connection processes is denoted as A_{conn}. *Minimum headway times* l_{ij} represent infrastructure capacity constraints between two trains. As such it reflects the railway safety system and may be accurately computed by using models described in Bešinović et al. (2017). The upper bound $u_{ij} = T - l_{ij}$ guarantees a minimum headway between trains in the reverse order. The set of minimum headways is denoted as A_{infra}. In case that a train line q has a frequency greater than one, $f_q > 1$, we introduce f_q copies of such a train line (Siebert & Goerigk, 2013). Departure and arrival events are additionally connected by *regularity activities* A_{reg}, whose lower and upper bounds are

set to T/f_q . Figure 5.1 shows an example of arrival and departure events for two trains in a station. Periodic constraints are given in the form $[l_{ij}, u_{ij}]_T$, where lower and upper bounds, l_{ij} and u_{ij}, hold for the cycle time T.

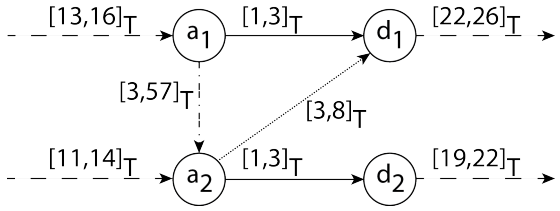

Figure 5.1: An extract of a periodic event-activity network for two trains stopping in a station with running (dashed line), dwell (full), transfer (dotted) and headway (dash-dotted) constraints

The aim of PESP is finding values π_i for all events $i \in E$ that satisfy constraints:

$$\pi_j - \pi_i + z_{ij}T \in [l_{ij}, u_{ij}].$$

The variable z_{ij} represents a modulo parameter that determines the order of events i and j within a period T for given bounds $[l_{ij}, u_{ij}]$ and equals 1 if $\pi_j < \pi_i$ or 0, otherwise. This binary property of z_{ij} holds assuming $l_{ij} \leq u_{ij}$, $0 \leq l_{ij} < T$ and $0 \leq u_{ij} - l_{ij} < T$.

Originally, PESP is a feasibility problem, which can be extended with a linear objective function and then solved by a mixed integer programming (MIP) formulation. Some of the choices for the optimization function are minimization of the total passenger journey time, minimization of required train units or maximization of the reliability of passenger connections. Peeters (2003) gives a detailed description on modeling objectives. In this paper, we adopt the common function of minimizing total journey time and define the nominal PESP, as:

$$(PESP-N) \qquad \text{Minimize} \sum_{(i,j)\in A} \tau_{ij}(\pi_j - \pi_i + z_{ij}T) \tag{5.1}$$

subject to

$$l_{ij} \leq \pi_j - \pi_i + z_{ij}T \leq u_{ij}, \qquad \forall (i,j) \in A \tag{5.2}$$

$$0 \leq \pi_i < T, \qquad \forall i \in E \tag{5.3}$$

$$z_{ij} \in \{0,1\}, \qquad \forall (i,j) \in A. \tag{5.4}$$

Here, τ_{ij} equals 0 if $(i,j) \in A_{\text{infra}}$, and 1 otherwise. As already stated, objective function (5.1) is the minimization of the total journey times. Constraint (5.2) defines bounds on the process times. Constraint (5.3) gives the periodicity of the events and schedules them in the interval $[0,T)$. Constraint (5.4) defines modulo operation as a binary decision variable. In the remainder of the paper, we refer to the original PESP with objective (5.1) as to the nominal timetable model PESP-N and the cycle time T as nominal cycle time.

Time allowances are additional times allocated to the processes on top of the minimum process times or minimum headway times. We differentiate between *time supplements* and *buffer times*. A time supplement is assigned to a single train process like running or dwelling to reduce possible stochastic behaviour of a given train. These variations may be due to a changed rolling stock, weather conditions or higher demand of passengers. Buffer times are inserted between trains, i.e., empty slots, to reduce the time dependency between them and therefore reduce possible delay propagation from an initially delayed train to other ones. In terms of arcs A in a periodic event-activity network, we treat differently running, dwelling and connection processes, $A_{\text{run}} \cup A_{\text{dwell}} \cup A_{\text{conn}}$ and headways A_{infra} by assigning them different time allowances. In particular, time supplements are given to process arcs, while buffer times to headway arcs. Figure 5.3 shows the distinction between time supplements and buffer times.

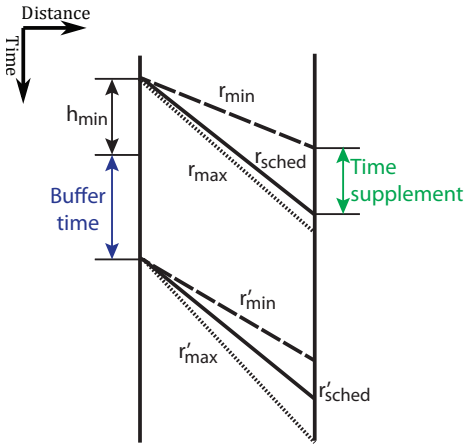

Figure 5.2: An example of a running time supplement for train r and a buffer time between trains r and r'. Subscripts min, sched and max define nominal, scheduled and maximum running times, respectively; h_{min} is the minimum headway between r and r'.

The scheduled cycle time (or timetable period) T defines a predetermined scheduled cycle time and in railway terms is usually 30, 60 or 120 minutes. The minimum cycle time is the shortest time duration in which all the events are feasible, meaning that all dependency constraints such as minimum running times and minimum headway times required by the infrastructure are respected.

The Stable and Robust periodic TTP can now be formulated as follows. Given are the requested train lines, periodic event-activity network and lower and upper bounds for running and dwell times, headways, and transfer times. The goal is to find a train schedule that is robust to minor delays during operations and at the same time provides efficient services and uses the infrastructure capacity minimally.

5.4 Two-stage model formulation

We propose the new concept stability-to-robustness which integrates a two-stage model for finding robust solutions of the periodic TTP. The first stage aims at finding an *op-*

timal stable timetable structure, i.e., the train orders that minimize the cycle time and thus allow maximimal buffer time for stability (Section 5.4.1). The obtained scheduled activities from Stage 1 are used as input to Stage 2 which determines an optimal allocation of time allowances within the timetable (Section 5.4.2). In Section 5.4.3 we discuss different objective functions considered in Stage 2. Finally, an optimization model for the robustness evaluation is deployed (Section 5.4.4).

To improve robustness, we focus on the interplay of time supplements and buffer times. Commonly, a certain amount of time supplement is preassigned to train running and dwell times. Thus, the corresponding lower bounds in timetabling models represent technical minimum running (dwell) times extended by some minimum time supplement that is necessary to withstand variations in train behavior. It is known that the minimum cycle time and thus the infrastructure capacity use depends on timetable structure (Hansen & Pachl, 2014). For example, assume a unidirectional track and two types of trains, slow and fast, both with frequency $f = 3$ trains per hour. These trains may be scheduled bundled such as three slow trains and then three fast ones; and such a timetable consumes the least infrastructure capacity. However, for passenger convenience, this should be avoided and trains of different types should run interchangeably (Figure 5.3). This problem becomes more difficult for more complex networks with different types of trains and various frequencies and stop patterns.

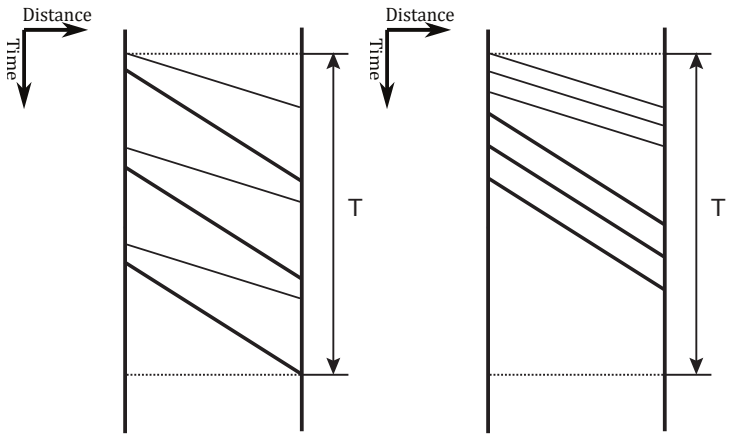

Figure 5.3: Track capacity occupation depending on train speed and train order: (a) maximally heterogeneous and (b) maximally bundled

Stage 1 allocates only additional time supplements that provide a better consumption of the railway infrastructure and are necessary to make all events feasible. Hence, we may consider these time supplements as beneficial (for the operators). Afterwards, in Stage 2 the aim is to increase robustness of a timetable by smart allocation of buffer times while keeping time supplements intact. Assuming that the initially allocated time supplements are sufficient to compensate variations caused by a single train, we accept that further addition of time supplements would be unfavorable for both passengers and operators. First, excessive running times are considered as reduced travel efficiency meaning that passengers want to travel as fast as possible. Second, train drivers do not always exploit existing time supplements to arrive on time in a station. On the

contrary, they tend to run faster which may result in arriving (or passing) too early a certain timetable point and catching up with the preceding train causing an unnecessary conflict and resulting braking and possible waiting in rear of a red signal (De Goffau, 2013). Even more, such early trains tend to be late in arriving at the terminal stations (Cerreto, Nielsen, Harrod, & Nielsen, 2016). Then such a train consumes even more energy and time, and thus also infrastructure capacity. In order to prevent this issue, the goal is to minimize allocation of further time supplements and introduce them only when they significantly improve the timetable robustness.

5.4.1 Finding an optimal stable timetable structure

The first stage aims at finding an optimal stable timetable structure, that is, the train orders that minimize infrastructure capacity use and thus maximize buffer time for stability. The performance of the timetable may be evaluated by determining its (network) minimum cycle time. The *minimum cycle time* is the shortest time duration in which all the events scheduled in the timetable are feasible for all precedence constraints such as minimum running times and minimum headway times required by the infrastructure (Goverde, 2010). In other words, the minimum cycle time represents the network capacity occupation while the scheduled event times within this model reflect a compressed timetable. We denote minimum cycle time as λ. It is easy to understand that if a train schedule uses less time on infrastructure resources, then there is more remaining time allowances to be distributed among the activities to increase the robustness of the timetable. Therefore, the goal of the first stage is to find the *optimal stable timetable structure* that consumes the minimal infrastructure capacity and as such corresponds to minimizing the cycle time. The difference between the minimum and scheduled cycle time defines the available time allowances on the critical circuit (considered in Stage 2). Therefore, the output of this model is referred to as the most stable timetable structure, i.e., the structure that uses the infrastructure in the optimal way and therefore leaves the most time allowances. The problem of finding the optimal stable timetable structure is formulated by taking the cycle time T as variable, and then solving the problem to minimize the cycle time λ. In addition, we use minimization of journey times as a secondary objective term with a small weight α to prevent an excessive extension of journey times. The new MILP formulation of the first stage problem is then the following:

$$(PESP-\lambda) \qquad \text{Minimize} \quad \lambda+\alpha \sum_{(i,j)\in A} \tau_{ij}(\pi_j-\pi_i+z_{ij}T) \tag{5.5}$$

subject to

$$l_{ij} \leq \pi_j-\pi_i+z_{ij}\lambda \leq u_{ij}, \qquad \forall (i,j)\in A \tag{5.6}$$

$$0 \leq \pi_i < \lambda, \qquad \forall i \in E \tag{5.7}$$

$$z_{ij} \in \{0,1\}, \qquad \forall (i,j)\in A \tag{5.8}$$

$$0 < \lambda \leq \lambda_{max}. \tag{5.9}$$

The objective function (5.5) is minimizing the cycle time and total journey times. Constraint (5.6) defines bounds on the process times. Constraint (5.7) sets the events in a periodic interval $[0,\lambda)$. Constraint (5.8) defines the modulo parameter as an integer decision variable. Constraint (5.9) defines λ to be strictly positive and smaller than a given upper bound λ_{max}. In order to preserve Constraint (5.8) it should hold $0 \leq l_{ij} < \lambda$ and $0 \leq u_{ij} - l_{ij} < \lambda$. If this is not the case, additional dummy nodes should be modeled as in Sparing and Goverde (2013). In the remainder of the paper, we refer to this model as PESP-λ.

Since the scheduled cycle time T from (5.1) is substituted with a decision variable λ, the constraint (5.6) becomes nonlinear because of the nonlinear terms $z_{ij}\lambda$. This can be linearized by introducing new variables $y_{ij} = z_{ij}\lambda$ and the following set of constraints (Sparing & Goverde, 2013):

$$0 \leq y_{ij} \leq \lambda \tag{5.10}$$
$$y_{ij} \geq \lambda - M(1 - z_{ij}) \tag{5.11}$$
$$y_{ij} \leq M z_{ij} \tag{5.12}$$

Here, M is a suitable upper bound for the objective value λ and it is defined by solving an original PESP formulation and finding a feasible solution to given constraints and fixed T. When a feasible solution exists for T, then $M = T$. In addition, an upper bound on λ is set to T, $\lambda_{max} = T$, as it is proven that $\lambda \leq T$ by finding a feasible solution. Constraints (5.10)-(5.12) allow exactly the values of 0 or λ to y_{ij}, depending on the modulo value z_{ij}. When z_{ij} equals 0, y_{ij} is also 0, while z_{ij} equals 1, y_{ij} takes λ. Note that by solving PESP-λ, it may happen that λ becomes bigger than a desired value of T. Then, the timetable is structurally unstable (Goverde, 2007, 2010). If that is the case, we first relax lower and upper bounds (see Bešinović et al. (2017)).

The output of PESP-λ is the minimum cycle time λ and (π, z), where π are the event times for all $i \in E$ and z are the modulo parameters for each arc $(i,j) \in A$. The associated scheduled activity times can be directly computed as:

$$\tilde{a}_{ij} = \pi_j - \pi_i + z_{ij}\lambda, \qquad (i,j) \in A_{\text{run}} \cup A_{\text{dwell}}. \tag{5.13}$$

The obtained *minimal scheduled activity times* $\tilde{a}_{ij}$ provide the optimal cycle time and as such should be maintained as close as possible when distributing buffer times in Stage 2. In particular, Stage 1 fixes the train orders and minimum passenger times.

5.4.2 Optimal distribution of time allowances

In Stage 2, the aim is the *optimal allocation of time allowances* so as to maximize the robustness. The goal is to increase the time separation between trains while preventing the extension of running, dwell or connection times. In other words, we maximize buffer times, while minimizing extra time supplements.

The train orders and available time allowances are known after solving PESP-λ, and the focus is on the problem of the optimal allocation of the time allowances within the given timetable structure. Here we exploit an alternative MILP formulation of PESP which solves the problem using activity times a_{ij} of arcs $(i,j) \in A$ instead of solving for events $i \in E$ and event times π_i where a_{ij} presents the periodic difference between event times π_j and π_i as $a_{ij} = \pi_j - \pi_i + z_{ij}T$. Such a formulation is known as the *cycle periodicity formulation* (CPF) (Peeters, 2003).

In the CPF, the cycles in the periodic event-activity network represent the essential graph structure for defining a feasible solution. For a cycle c an arc is assigned with a variable d_{ij} equal to 1 if a_{ij} is a forward arc and $d_{ij} = -1$ if a_{ij} is a backward arc. A *cycle basis* of a graph is a family of cycles that spans all cycles of the graph.

Let $N = (E,A,T,l,u)$ be a periodic event-activity network representing a PESP instance, and $a \in \mathbb{Q}^{|A|}$. Then for every cycle c in N, we have

$$\sum_{(i,j)\in c} d_{ij}a_{ij} = p_c T, \qquad p_c \in \mathbb{Z}, \tag{5.14}$$

$$\text{and} \qquad l_{ij} \le a_{ij} \le u_{ij}. \tag{5.15}$$

In other words, the sum of all arcs a_{ij} corresponding to periodic events π_i and π_j along a cycle must be equal to an integral multiple of T. Here, p_c is called cyclic periodicity integer variable.

The CPF model may use various cycle bases in terms of a number of cycles and their structure. Yet, it is sufficient to consider fundamental cycles from which all other cycles may be obtained (Liebchen & Peeters, 2009). We restrict ourselves to a strongly connected graph. The necessary number of cycles is $|C| = m - n + 1$, where $m = |A|$ and $n = |E|$. Different approaches have been considered for constructing cycle bases. Liebchen, Proksch, and Wagner (2008) reported that the cycle basis generated over a *minimum spanning tree* (MST) gives good overall results. Thus, we adopt their approach and generate a cycle basis C that consists of fundamental cycles of a MST with respect to the span between maximum and minimum process times, $u_{ij} - l_{ij}$.

Let d_{ij}^- equal -1 if (i,j) is a backward arc, and $d_{ij}^- = 0$ otherwise. Likewise, d_{ij}^+ equals 1 if (i,j) is a forward arc, and $d_{ij}^+ = 0$ otherwise.

Proposition 1 (Odijk, 1996) *Let $N = (E,A,T,l,u)$ be a periodic event-activity network representing a PESP instance. Then for any cycle c in C, the following inequali-*

ties hold:

$$l_c \leq p_c \leq u_c \tag{5.16}$$

$$\text{with} \qquad l_c = \left\lceil \frac{1}{T} \left(\sum_{(i,j)\in c} d_{ij}^{-} u_{ij} + \sum_{(i,j)\in c} d_{ij}^{+} l_{ij} \right) \right\rceil \tag{5.17}$$

$$\text{and} \qquad u_c = \left\lfloor \frac{1}{T} \left(\sum_{(i,j)\in c} d_{ij}^{-} l_{ij} + \sum_{(i,j)\in c} d_{ij}^{+} u_{ij} \right) \right\rfloor. \tag{5.18}$$

The search space of CPF may be reduced considerably by setting appropriate bounds for p_c. The lower bound (5.17) on a cycle l_c is computed by summing the maximum time allowances of backward arcs and the minimum time allowances of forward arcs. Similarly, the upper bound (5.18) is the sum of minimum times of backward arcs and maximum time allowances of forward arcs.

Denote by s_{ij} the time allowance of $(i,j) \in A$. Naturally, s_{ij} takes only positive values, $s_{ij} \geq 0$, as we allow only realizable process times, meaning that $a_{ij} \geq l_{ij}$ for all $(i,j) \in A$. The relation between the scheduled process time a_{ij} and the time allowance s_{ij} is known and defined as $a_{ij} = l_{ij} + s_{ij} + z_{ij}T$. Hence, we have $s_{ij} = a_{ij} - l_{ij} - z_{ij}T \geq 0$. This allows us to focus on the time allowances s and we rewrite (5.14) as

$$\sum_{(i,j)\in c} d_{ij} s_{ij} - p_c T = -\sum_{(i,j)\in c} d_{ij} l_{ij}. \tag{5.19}$$

The corresponding upper bound on the time allowances is defined as the difference between upper and lower bound for a_{ij}. So, $s_{ij} \in [0, u_{ij} - l_{ij}]$. The CPF formulation for solving optimal times allowances $s = [s_{ij}]$ reads as follows:

$$(CPF - s) \qquad \text{Maximise} \quad f(s) \tag{5.20}$$

subject to

$$\sum_{(i,j)\in c} d_{ij} s_{ij} - p_c T = -\sum_{(i,j)\in c} d_{ij} l_{ij}, \qquad \forall c \in C \tag{5.21}$$

$$l_c \leq p_c \leq u_c, \qquad \forall c \in C \tag{5.22}$$

$$s_{ij} \in [0, u_{ij} - l_{ij}], \qquad \forall (i,j) \in A \tag{5.23}$$

$$p_c \in \mathbb{Z}, \qquad \forall c \in C. \tag{5.24}$$

The objective function (5.20) is maximizing a given function of s that will be defined in Section 5.4.3. Constraint (5.21) bounds the total time allowance on the fundamental cycles. Constraints (5.22) define bounds on p_c as presented by Proposition 1. Constraint (5.23) defines bounds on time allowances s_{ij}. In the remainder of the paper we refer to the problem of optimizing the allocation of time allowances as CPF-s. Using the timetable structure from Stage 1, we define several improvements to the CPF-s for-

mulation. Recall that time supplements are assigned to process arcs while buffer times to headway arcs.

Improvement 1 *Given an optimal stable timetable structure from PESP-λ, a tighter formulation of constraints* (5.21) *for processes may be obtained by substituting lower bounds l_{ij} with scheduled process times $\tilde{a}_{ij}$ for $(i,j) \in A \setminus A_{\text{infra}}$.*

In Stage 2, the goal is to maintain the running times from Stage 1, meaning that minimal scheduled process times $\tilde{a}_{ij}$ are taken as an input to be preserved or eventually increased minimally. In this way, the optimal minimum cycle time λ is maintained. Note that this improvement is not applied to buffer times, as we want them to be as flexible as possible to achieve a better timetable robustness.

As a consequence of Improvement 1, we also get the following.

Improvement 2 *Given an optimal stable timetable structure from PESP-λ, then* (5.23) *can be replaced by $s_{ij} \in [0, u_{ij} - \tilde{a}_{ij}]$ for $(i,j) \in A \setminus A_{\text{infra}}$.*

Here, the upper bound (5.23) is improved for every $\tilde{a}_{ij} > l_{ij}$, $(i,j) \in A \setminus A_{\text{infra}}$, and remains unchanged otherwise. One may notice that the amount of time allowances for a critical circuit equals $T - \lambda$ while for other (non-critical) circuit it is always greater than $T - \lambda$. From Stage 1, all cycles in C have the following property:

$$\left(\sum_{(i,j) \in c} d_{ij} \tilde{a}_{ij} \right) \bmod \lambda = 0. \tag{5.25}$$

Since the train order from Stage 1 defines the optimal stable timetable structure, we need to fix event orders from Stage 1 to Stage 2. To do so, we first introduce the relation between modulo parameters z_{ij} and cyclic periodicity integer variable p_c. The modulo parameters z_{ij} and cyclic periodicity integer variable p_c for a cycle c are related by

$$p_c = \sum_{(i,j) \in c} d_{ij} z_{ij},$$

meaning that p_c equals the directed sum of modulo values.

In order to fix train orders from Stage 1, the modulo parameters z_{ij} for all $(i,j) \in A$ are fixed and given in Stage 2 as follows.

Improvement 3 *Given an optimal stable timetable structure $(\pi, \tilde{z})$ from PESP-λ where z defines event orders, then cyclic periodicity integer variables p_c are computed and fixed as:*

$$p_c = \sum_{(i,j) \in c} d_{ij} \tilde{z}_{ij}, \qquad \forall c \in C.$$

According to Nachtigall (1999) and Peeters (2003), this constraint secures the sequence of events in the cycle. Therefore, we use this proposition to fix the order of events from Stage 1 to Stage 2 and this also fixes integer variables in CPF-λ-s. Note that Improvement 3 relaxes CPF to the linear programming (LP) formulation, which renders the model to be easily solvable.

In order to further speed up the model's computation time, we introduce new cycle bases by the MST approach and apply different edge weights. In particular, we exploit scheduled process times as well as lower and upper bounds to generate different MSTs.

Improvement 4 *We use the timetable structure obtained in Stage 1 and adapt scheduled activity times $\tilde{a}_{ij}$ as graph weights for constructing fundamental cycle bases. New cycle bases X are constructed:*

- $X_{\tilde{a}}$*: MST subject to arc weights according to the minimally scheduled activity times* $\tilde{a}_{ij}$
- X_l*: MST subject to weights* l_{ij}
- $X_{\lambda l}$*: MST subject to weights* $\lambda - l_{ij}$
- $X_{\lambda \tilde{a}}$*: MST subject to weights* $\lambda - \tilde{a}_{ij}$
- $X_{\tilde{a}l}$*: MST subject to weights* $(\tilde{a}_{ij} - l_{ij})$
- $X_{\lambda \tilde{a} l}$*: MST subject to weights* $\lambda - (\tilde{a}_{ij} - l_{ij})$

We generated six new cycle bases with different characteristics, and the task is to determine the preferable cycle basis (see Section 6.5.2).

Finally, the linear programming (LP) model of the CPF in Stage 2 used for the optimal distribution of time allowances consists of solving the following:

$$(CPF - \lambda - s) \qquad \text{Minimize} \quad f(s) \tag{5.26}$$

subject to

$$\sum_{(i,j)\in c} d_{ij} s_{ij} = p_c T - \sum_{(i,j)\in c} d_{ij} a_{ij}, \qquad \forall c \in X \tag{5.27}$$

$$p_c = \sum_{(i,j)\in c} d_{ij} \tilde{z}_{ij}, \qquad \forall c \in X \tag{5.28}$$

$$s_{ij} \in [0, u_{ij} - \tilde{a}_{ij}], \qquad \forall (i,j) \in A \setminus A_{\text{infra}} \tag{5.29}$$

$$s_{ij} \in [0, u_{ij} - l_{ij}], \qquad \forall (i,j) \in A_{\text{infra}} \tag{5.30}$$

Note that constraints (5.27)-(5.29) represent improved variants of (5.21)-(5.23) by exploiting the Improvements. Also, a new cycle basis is used, where X represents one

from Improvement 4. The procedure to obtain back periodic event times π_i is as follows. Given time allowances s_{ij} from CPF-λ-s and the MST X. Choose an arbitrary event $r \in E$ and set $\pi_r = 0$. Assuming N is a connected graph, for all other events $i \in E$, we take the undirected path P_{ri} in X from r to i and set

$$\pi_i = \left(\sum_{(i,j) \in P_{ri}} d_{ij}(l_{ij} + s_{ij}) \right) \bmod T.$$

5.4.3 Objective functions for Stage 2

Distributing time allowances is a complex task, maybe not computationally but for sure tactically and it often does not depend on one goal but on multiple ones. On one hand, we want to run trains on time and to ensure that a certain amount of time supplement is assigned to minimum process times. Second, big extensions of running and/or dwell times should be avoided as it would lead to inefficient solutions and would be negatively perceived by passengers. In addition, we aim to prevent trains being scheduled too closely in time which will help reducing the occurrence of delays propagating between consecutive trains. Note that certain time supplements have been allocated already in Stage 1 to obtain the optimal stable timetable structure. Therefore, in the second stage, the goal is to maintain them as much as possible, but distribute buffer times in a clever way. For this purpose, we introduce different weights for buffer times and time supplements, w_b and w_s, respectively. Thus, we consider the second stage as a bi-objective problem where w_b takes a positive and w_s takes a negative value. The chosen objective function for (5.26) may have a significant effect on the robustness and efficiency of the final solution and thus, on its quality. Therefore, we propose and test several objective functions for solving CPF-λ-s in Stage 2.

In Stage 1 of the stability-to-robustness approach, the PESP-λ model used all headway constraints between each pair of train lines in order to satisfy timetable feasibility. However, in Stage 2, adding buffer times to all headway arcs does not necessarily influence the timetable robustness. First, adding more buffer between two trains already significantly separated in time does not add on the robustness. Second, adding buffer between two trains that are not consecutive (i.e., not running exclusively after each other) may also be unprofitable. Thus, we want to focus on allocating buffer between two consecutive trains, and stretch away such neighboring trains (i.e., their corresponding events) in order to allow more flexibility in reducing consecutive delays, that is providing better timetable robustness. To do this, we create a set of essential headway arcs $\hat{H}$, where $\hat{H} \subset A_{\text{infra}}$, that would benefit the most from extra buffer time. This makes our objective functions more compact and explicit by targeting the subset of right headway arcs.

Set $\hat{H}$ is determined using the procedure explained in Algorithm 7. First, the procedure selects headway arcs (and corresponding events) that belong to a single station, which gives a subset A_k with arcs $(i, j) \in A_{\text{infra}}$ that have at least events i or j occur in station

Algorithm 7 Determine essential headways

$\hat{H} = \varnothing$
For $k \in K$
  $A_k \leftarrow \{(i,j) | (i \in E_k \vee j \in E_k) \wedge (i,j) \in A_{\text{infra}}\}$
  Make an undirected graph $(A_k \rightarrow \bar{A}_k)$ with edges $e_{ij} = \min(\tilde{a}_{ij}, T - \tilde{a}_{ji})$
  Find a minimum spanning forest over $\bar{A}_k$: $F = MSF(\bar{A}_k)$
  Assign all forest arcs from F to essential headways: $\hat{H} \leftarrow F$
  For non-forest arcs $(i,j) \in A_k \setminus F$
    if length of $e_{ij} < ShortestPath_{ij}$ *in* F
      $\hat{H} \leftarrow (i,j)$
    EndIf
  EndFor
EndFor

k. All events belonging to station k are included in set E_k.

Second, we assume an edge for each arc (i,j) in A_k and assign the following weight to each edge $e_{ij} = \min(\tilde{a}_{ij}, \lambda - \tilde{a}_{ij})$. This has been done to determine the relevant (time) dependencies between events i and j and translate them in a single scheduling period between 0 and T. The minimum spanning forest (MSF) F_k is determined for station k. The MSF represents a set of minimum spanning trees, meaning that multiple connected components may exist in A_k. Note that it is common to expect multiple MSTs in a station as a result of physically separated infrastructure that creates independent routes in stations for (groups of) trains. For example, trains running in opposite directions through a station often do not share infrastructure and thus operate independently. All forest arcs $(i,j) \in F_k$ are assigned to essential headways $\hat{H}$.

In addition, it may occur that there exists a shorter (i.e., more restrictive) edge between i and j in A_k but which is not in F, i.e., a non-forest edge. Check the example of five events in Figure 5.4. In order to include such arcs in the essential headways, we look at all non-forest arcs. If a non-forest arc (i,j) has a weight smaller than the shortest path (SP) between i and j using only MSF arcs, then such arc (i,j) is also included in $\hat{H}$. Therefore, in order to increase timetable robustness, we maximize buffer times over arcs in $\hat{H}$.

Consider five trains running through a station with passing event times $\pi_1 = 0$, $\pi_2 = 7$, $\pi_3 = 5$, $\pi_4 = 11$ and $\pi_5 = 19$ minutes, respectively, the minimum headways between each two events equal 1 minute (in total, 6 arcs), the optimal minimum cycle time is $\lambda = 28$ minutes and the cycle time is $T = 30$ minutes (Figure 5.4). Note that if process time a_{ij} in a half period is over $\lambda/2 = 15$ minutes, then e_{ij} assigns $\lambda - \tilde{a}_{ij}$. For example, the edge $(1,5)$ weighs $e_{15} = \min(19, 28-19) = 9$. The corresponding MSF includes edges $\{(1,3),(2,3),(2,4),(4,5)\}$ and evaluates to 18. Comparing the remaining non-forest edges $\{(1,2),(3,4),(3,5),(1,5)\}$ with the corresponding shortest paths, it is concluded that only edge $(1,5)$ has a smaller weight than its SP, and thus this edge is also added to $\hat{H}$. Finally, $\hat{H} = \{(1,3),(2,3),(3,4),(4,5),(1,5)\}$ and the corresponding arcs are taken in the objective functions for CPF-λ-s. We now propose several objectives. First,

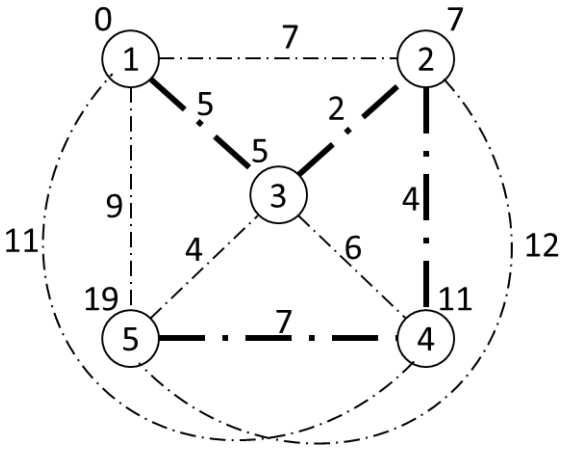

Figure 5.4: An extract of a periodic event-activity network for five events a station with headways (dash-dotted) constraints. Scheduled events are given in circles, the corresponding event times are attached to circles, and edges are accompanied with its weights. Bold edges represent the minimum spanning tree.

MaxBuffer is the function that maximizes a weighted sum of headway buffers while minimizing the passenger-related process times,

$$\text{Maximize} \sum_{(i,j)\in\hat{H}\cup A_{\text{run}}\cup A_{\text{dwell}}\cup A_{\text{conn}}} w_{ij}s_{ij}, \tag{5.31}$$

where w_{ij} is equal to w_b if $a_{ij} \in \hat{H}$ and equal to w_s otherwise. Note that $w_b > 0$ and $w_s < 0$. This objective function focuses on obtaining linear time allowances. This means that there is no distinction between smaller and bigger time supplements, instead all are treated as equally important. Thus, MaxBuffer tends to unevenly distribute buffer times assigning big buffers to a few activities while nothing to the others.

In general, having arcs $(i,j) \in A_{\text{infra}}$ of the length in the middle of the time window $[l_{ij}, u_{ij}]$ would be the most beneficial for timetable robustness as train events would be separated as much as possible. We apply this idea of spreading trains in time. Let us use essential headway arcs in $\hat{H}$. We define the parameter m_{ij} as

$$m_{ij} = \frac{u_{ij} - l_{ij}}{2},$$

similarly as in Peeters (2003). We also introduce a new variable β_{ij} as the difference between buffer times $s_{ij} \in \hat{H}$ and parameter m_{ij}, $\beta_{ij} = s_{ij} - m_{ij}$. Variable β_{ij} may take the both bigger and smaller values than m_{ij} and thus, the objective is to minimize the sum of $|\beta_{ij}|$. In order to preserve $|\beta_{ij}|$ linear, two new constraints are added for all $(i,j) \in \hat{H}$:

$$\beta_{ij} \geq s_{ij} - m_{ij},$$
$$\beta_{ij} \geq m_{ij} - s_{ij}.$$

and the corresponding objective is defined as

$$\text{Minimize} \quad \sum_{(i,j)\in\hat{H}} \beta_{ij}.$$

This objective function forces buffer times $s_{ij} \in \hat{H}$ toward the middle of the time window m_{ij} with window bounds l_{ij} and u_{ij} and thus dividing corresponding events as much as possible. Due to the proposed bi-objective function that maximizes both robustness and efficiency, we introduce the objective *MaxHalfBuffer* as

$$\text{Minimize} \quad \sum_{(i,j)\in\hat{H}} \beta_{ij} - \sum_{(i,j)\in A_{\text{run}}\cup A_{\text{dwell}}\cup A_{\text{conn}}} w_{ij}s_{ij}. \tag{5.32}$$

During the second stage optimization, the resulting timetable may have a high value of the objective value, such as the total sum of buffer times, but it may still have processes with no buffer time allocated. For example, many headways receive big buffer times, while a certain number of them receives no buffer times at all. Therefore, to encourage allocating buffers to all processes, we introduce the objective function *MaxMin* that maximizes the minimum value of buffer times

$$\text{Maximize} \quad \min_{(i,j)\in\hat{H}} s_{ij}. \tag{5.33}$$

The objective function (5.33) can be reformulated to a standard maximization linear program by introducing a new variable t as

$$\text{Maximize} \quad t \tag{5.34}$$

subject to

$$t \leq s_{ij}, \quad \forall (i,j) \in \hat{H}. \tag{5.35}$$

Therefore, MaxMin is accommodated with a second objective of minimizing time supplements, referred to as *MaxMin+*.

$$\text{Maximize} \quad \min_{(i,j)\in\hat{H}} s_{ij} + \sum_{(i,j)\in A_{\text{run}}\cup A_{\text{dwell}}\cup A_{\text{conn}}} w_{ij}s_{ij}. \tag{5.36}$$

Objective (5.36) is modified in a same way as (5.33). Setting the objective over $\hat{H}$ is without loss of generality as to the headway arcs that are in $A_{\text{infra}} \setminus \hat{H}$ since the corresponding buffer times are always at least equal (but usually bigger than) to the ones in $\hat{H}$. Thus, we may exclude them from the objective function.

Note that only MaxMin is a single objective function, while all other are bi-objective that maximize buffer times and minimize time supplements. Also, defining objective functions over essential headway and not over all possible ones is an added benefit of a two-stage stability-to-robustness approach. This was not possible to define in single-stage models (e.g., Caimi, Fuchsberger, Laumanns, and Schüpbach (2011); Liebchen

(2008); Peeters (2003)), since the order was not defined in advance.

5.4.4 Robustness evaluation model

We use a delay propagation model to evaluate the obtained timetable. The *robustness evaluation model* is defined as an optimization model that aims at minimizing the sum of all delays for a given set of disturbance scenarios. The objective is to minimize the average cumulative delay from the network over all delay scenarios. We consider a time horizon over $\bar{H}$ hours in order to capture the propagation over a longer period of time. In transportation networks with dense traffic, it may commonly happen that a train from one hour causes a delay to a following train in a next period. In the delay propagation model, we assume a train order to be fixed as is in the planned timetable, so no reordering measures within station areas are taken into account.

For modelling the delay propagation, we use the formulation of the precedence constraints:

$$\pi_j - \pi_i \geq l_{ij} \tag{5.37}$$

where π_i and π_j are the fixed scheduled event times from the optimization and l_{ij} is the minimum process time over arc (i, j). The objective is to minimize the cumulative delay over a given network. Depending on the size and the distribution of the disturbance input, it may be partially or totally compensated within the time horizon. Let b_{ij} be fixed binary parameters from two-stage optimization defining whether a process (i, j) crosses the hour. Thus, b_{ij} equals 0 if $\pi_i \leq \pi_j$, and 1 otherwise. The robustness evaluation model is implemented as a Monte Carlo simulation of R replications over an optimization model that aims at minimizing the sum of all delays for a given disturbance scenario $\delta^{(r)}$ in replication r:

$$D^{(r)} = \text{Minimize} \sum_{i \in V} \sum_{h \in H} D_i(h) \tag{5.38}$$

subject to

$$\pi_j^{(r)}(h + b_{ij}) - \pi_i^{(r)}(h) \geq l_{ij} + \delta_{ij}^{(r)}(h), \qquad \forall (i,j) \in E, \forall h \in H, \tag{5.39}$$

$$\pi_i^{(r)}(h) \geq \pi_i + hT, \qquad \forall i \in V, \forall h \in H, \tag{5.40}$$

$$\pi_i^{(r)}(h) - \pi_i - hT \leq D_i(h), \qquad \forall i \in V, \forall h \in H \tag{5.41}$$

Here, $H = \{1, \ldots, \bar{H}\}$ and $\pi_i^{(r)}(h)$ defines a simulated event time for event i in hour h and realization r with respect to the input disturbances. Constraint (5.39) guarantees that the precedence constraints are satisfied for the minimum process times plus a disturbance $\delta_{ij}^{(r)}(h)$ in the h-th hour. Constraint (5.40) satisfies the market needs that a train may not depart earlier than scheduled in the h-th hour. Finally, (5.41) defines all delay variables $D_i(h)$ as non-negative values. Given a timetable output from CPF-λ-s, the robustness evaluation model tests it against a large set of R replications of disturbances, one at a time, collecting the information on the value of the objective

function and providing the average cumulative delay

$$\bar{D} = \frac{1}{R} \sum_{r=1}^{R} D^{(r)}$$

as a quality measure of the timetable robustness.

5.5 Experimental results

5.5.1 Case scenarios

We illustrate the benefits of the two-stage stability-to-robustness approach for designing stable and robust periodic timetables on three highly utilized parts of the network in the Netherlands. Figure 5.5 gives a graphical representation of the considered networks. The first considered network N1 is adopted from Kroon, Huisman, and Maróti (2014) and consists of 6 major stations and is served by 12 passenger and one freight train line. The nominal timetable period is $T = 60$ minutes.

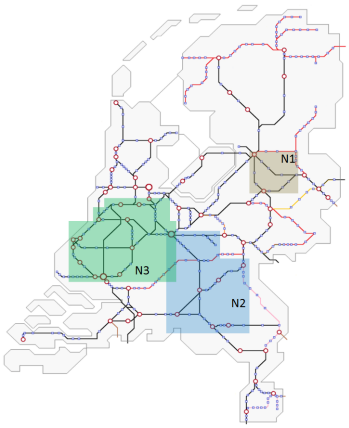

Figure 5.5: Considered networks: N1, N2 and N3

The second network N2 is a central part of the Dutch railway network (Bešinović et al., 2016), consisting of the railway network bounded by the four main stations Utrecht (Ut), Eindhoven (Ehv), Tilburg (Tb) and Nijmegen (Nm), with a fifth main station s-Hertogenbosch (Ht) in the middle and 20 additional smaller stations and stops. Four corridors connect Ht to the other main stations. The train line plan consists of four intercity lines and six local train lines with a frequency of two trains per hour each.

The third network N3 is more complex and includes a bigger portion of the Dutch railway network, called Randstad area, which represents a third of the complete national network. We consider only IC train lines in this network. Table 5.1 gives the overview on the characteristics of the used networks like the number of train lines (Q), stations (K), events (E) and processes (A) (i.e., total, run, dwell and headways) as well as average nominal running times (L_{run}), dwell times (L_{dwell}), average minimum journey times over train lines (av minJT) and minimum headway times (L_{infra}).

The scheduled cycle time is 3600 seconds (i.e., 1 hour). For the first two networks, all lines have a frequency of two trains per hour. Hence, it is possible to consider only one train of each train line and a half timetable period, which is $T/2 = 1800$ seconds. After the timetable is generated for $T/2$, it can be duplicated w.l.o.g. in order to obtain services for the complete hour.

Some of the timetabling parameters are given in the following. Lower bounds of running times, defined as nominal running times, are equal to the minimum running times increased by a minimum allowed time supplements of 5%. This is a common value for planning purposes that is needed to compensate for uncertain train driving behavior. The upper bounds represent 30% increment from the minimum running times, which reflects the commercial need of a train operating company. Dwell times represent the required times to board/alight and are accepted somewhat bigger than the technical minimum to passengers convenience. Passenger connection times are computed in a similar fashion. Usually, a train operating company defines norms for a technical minimum dwell and passenger connection times. The minimum passenger connection times is assumed 4 minutes. After initial experiments, the weights of time supplements the the secondary objective in PESP-λ are set to $w_s = 0.0001$; and weights of buffer times and time supplements in CPF-λ-s are set to $w_b = 0.1$ and $w_s = -1$, respectively, unless stated otherwise. In addition, to scale the two objectives in MaxMin+, being minimal buffer time and time allowances, we used the multiplication factor of 10^7 for the former.

We implemented the two-stage model in Matlab using the Yalmip toolbox (Löfberg, 2004) and the commercial solver Gurobi version 6.0.2. All computations have been evaluated using an Intel i7 PC with 2.8 GHz and 8GB RAM. The reported computation times represent the average over 30 repeated runs of each model. We also set the time limit on the computation time to 1800 s and in case that an optimal solution is not achieved, we report the optimality gap.

Table 5.1: Network characteristics

Net	$\|Q\|$	$\|K\|$	$\|E\|$	$\|A\|$	$\|A_{\text{run}}\|$	$\|A_{\text{dwell}}\|$	$\|A_{\text{infra}}\|$	L_{run}	L_{dwell}	av minJT	L_{infra}
N1	14	6	56	97	34	8	51	1084	60	795	181
N2	20	14	152	592	108	24	449	528	49	384	90
N3	74	86	1116	7896	834	208	6854	125	51	110	180

The robustness of the produced timetables is evaluated using the robustness evaluation model (Section 5.4.4). The simulation setup consists of the following input: disturbances of process times are modeled according to an Exponential distribution with distribution parameter ρ, the simulation horizon is $\bar{H} = 4$ hours, and the number of replications equals $R = 30$. Parameter ρ represents the percentage of the minimum process time (e.g., run, dwell) and defined as $\rho = \mu \cdot a_{ij}$. Parameter μ varies from 0 to 10 % of the minimum process times. Note that similar distributions have been used at Netherlands Railways (NS) (Maróti, 2016). From the formulation in Section 5.4.4 can be seen that maximizing the robustness equals minimizing the cumulative delay. Thus,

we report the average cumulative delay in the experiments. In this section, we perform the following analyses:

- Quantifying the effect of the various cycle bases - Improvement 4,
- Performance evaluation of stability-to-robustness models and comparison with PESP-N,
- Sensitivity analysis of robustness for different disturbance scenarios,
- Measuring impact of the time supplements' weights on the generated timetables.

5.5.2 Testing cycle bases

Before detailed analyses of computed timetables, we tested the effect of the Improvement 4 on the computation times. Table 5.2 reports the obtained cycle basis width, and average absolute and relative running times for defined cycle bases. It can be seen that the MST with weights $\lambda - l_{ij}$ performs the best for the network N1, while the originally proposed cycle basis C is one of the worse performing ones by having 149.2 % bigger computation time than $X_{\lambda l}$. Similar results were observed for N2 and N3, and so we accept cycle basis $X_{\lambda l}$ for all further experiments and all test networks.

Table 5.2: Performance of cycle bases on N1

Cycle basis	Arc weights	CPU time [s]	CPU time [%]
$X_{\lambda l}$	$\lambda - l_{ij}$	0.0137	100.0
$X_{\lambda ul}$	$\lambda - (u_{ij} - l_{ij})$	0.0149	108.3
X_{al}	$a_{ij} - l_{ij}$	0.0227	165.2
$X_{\lambda a}$	$\lambda - a_{ij}$	0.0289	210.3
X_l	l_{ij}	0.0309	225.2
C	$u_{ij} - l_{ij}$	0.0342	249.2
X_a	a_{ij}	0.0346	252.0
$X_{\lambda al}$	$\lambda - (a_{ij} - l_{ij})$	0.0425	309.1

5.5.3 Testing the two-stage model on different network instances

We used the single stage PESP formulation PESP-N as reference to compare with the timetables generated by the two-stage stability-to-robustness approach. PESP-N was applied with two objective functions: one minimizing total train journey times (*N-MinTrainTimes*) and second minimizing time supplements and maximizing buffer times *N-MaxBuffer*, which is similar to MaxBuffer (5.31) and sums time allowances over A. In all two-stage cases, PESP-λ is solved in Stage 1, while CPF-λ-s with different objective functions being MaxBuffer, MaxMin, MaxMin+, HalfBuffer and HalfBuffer+ is applied in Stage 2. Table 5.3 gives the statistics of the obtained results for all instances. In particular, it shows the considered network (Net), model and used weights for time supplements and buffer times; and then, presents the results like objective value ($objval$), cycle time λ, time supplement (running, dwell and total), average

Table 5.3: Results on minimum cycle times, objective functions, time allowances and computation times

Net	Model	Obj function	w_s	w_b	objval	$\lambda[s]$	Running time supp [s]	Dwell time supp [s]	Total time supp [s]	Avg time supp [%]	Min Buffer [s]	Zero Buffers	CPU time [s]	Opt gap [%]
N1	PESP-λ	Min λ	-1	0	1621	1621	130	0	130	0.45	0	5	0.04	0
N1	CPF-λ-*s*	MaxBuffer	-1	0.1	272	1800	130	0	130	0.45	0	4	0.01	0
N1	CPF-λ-*s*	MaxMin	-1	0.1	-30	1800	628	199	827	2.26	30	0	0.01	0
N1	CPF-λ-*s*	MaxMin+	-1	0.1	-29499495	1800	407	110	517	2.10	29	0	0.01	0
N1	CPF-λ-*s*	HalfBuffer	-1	0.1	11357	1800	1553	379	1932	5.02	0	4	0.01	0
N1	CPF-λ-*s*	HalfBuffer+	-1	0.1	12758	1800	596	379	975	2.07	0	4	0.01	0
N1	PESP-N	N-MinTrainTimes	-1	0	-240	1800	0	0	240	0.84	0	2	0.14	0
N1	PESP-N	N-MaxBuffer	-1	0.1	4851	1800	0	0	240	0.84	0	10	1.09	0
N2	PESP-λ	Min λ	-1	0	1561	1560	2279	3710	5989	4.96	0	8	7.96	0
N2	CPF-λ-*s*	MaxBuffer	-1	0.1	1529	1800	2785	3903	6688	6.24	0	27	0.03	0
N2	CPF-λ-*s*	MaxMin	-1	0.1	-10	1800	4688	4164	8852	12.42	10	0	0.03	0
N2	CPF-λ-*s*	MaxMin+	-1	0.1	-9603126	1800	3027	3883	6910	7.27	10	0	0.03	0
N2	CPF-λ-*s*	HalfBuffer	-1	0.1	82024	1800	4112	4620	8732	10.98	0	16	0.06	0
N2	CPF-λ-*s*	HalfBuffer+	-1	0.1	85846	1800	3207	4031	7238	7.88	0	25	0.04	0
N2	PESP-N	N-MinTrainTimes	-1	0	-6605	1800	126	2978	6605	0.37	0	33	14.45	0
N2	PESP-N	N-MaxBuffer	-1	0.1	14489	1800	383	3033	7016	1.13	0	36	1800.69	10.36
N3	PESP-λ	Min λ	-1	0	3067	3060	210	240	450	0.43	0	39	0.69	0
N3	CPF-λ-*s*	MaxBuffer	-1	0.1	-2205	3600	210	240	450	0.43	0	19	0.08	0
N3	CPF-λ-*s*	MaxMin	-1	0.1	-54	3600	6328	4421	10749	10.11	54	0	0.15	0
N3	CPF-λ-*s*	MaxMin+	-1	0.1	-54249851	3600	331	268	599	0.54	54	0	0.13	0
N3	CPF-λ-*s*	HalfBuffer	-1	0.1	973209	3600	15278	12836	28114	35.36	0	37	0.14	0
N3	CPF-λ-*s*	HalfBuffer+	-1	0.1	990553	3600	8113	5527	13640	15.11	0	32	0.14	0
N3	PESP-N	N-MinTrainTimes	-1	0	0	3600	25	0	25	0.07	0	43	0.69	0
N3	PESP-N	N-MaxBuffer	-1	0.1	362018	3600	2355	700	3055	4.76	0	3	1800.39	1.89

time supplement (Avg time supp), minimum allocated buffer (Min buffer), number of headway arcs without any buffer allocated (Zero buffers), computation time and optimality gap. The minimum cycle time λ was computed in PESP-λ and given as fixed for other models as $\lambda = T$. The reported time supplement is the amount of the time that has been allocated within a model for PESP-λ and PESP-N, while for CPF-λ-s, it represents the sum of time supplements in both stages, since the time supplements assigned in Stage 1 are given as input and fixed in Stage 2 as defined in Improvement 1.

The minimum cycle time λ for the optimal stable timetable structure was 1621 s, 1560 s and 3060 s for the three considered network instances, respectively.

Analyzing the objective values, MaxMin and MaxMin+ are straightforward to understand. For MaxMin, the negative value is the minimal buffer time and for MaxMin+, the minimal buffer is multiplied with 10^7 and the remainder is the sum of time allowances. For example, MaxMin for N1 reported objval of -30 and the corresponding minimum buffer time is exactly 30 s and thus, no zero buffers were obtained. On the other hand, other objective values are more complex. For MaxBuffer, N-MinTrainTimes and N-MaxBuffer, the computed value represents the sum of all time allowances. For HalfBuffer and HalfBuffer+, (part of) the value is the sum of the deviated headway values from the middle of the scheduling interval m_{ij}. Comparing Halfbuffer and Halfbuffer+, the latter resulted in higher $objval$ due to the included time allowances in the objective function. As a consequence, HalfBuffer+ reports less total time supplements, e.g., being 2.07 % compared to 5.02 % for N1. The same relation was observed for all networks and even more, for functions MaxMin and MaxMin+.

By looking at each network, we observe that the models performed significantly different in finding a stable and robust periodic timetable which resulted in various amounts of time supplements allocated, ranging from 130 to 1932 s for N1, 5989 to 8852 s for N2 and 450 to 28114 s for N3. These all generated an average time supplement rate from 0.45 %, as MaxBuffer for N1, to 35,36 % as MaxMin for N3. In general, MaxBuffer tends to allocate less time supplements compared to other two-stage models. For PESP-N models, N-MinTrainTimes is purely focused on reducing train-related process times which resulted in only limited time supplements, totaling up to 0.7 % for all networks, Meanwhile, N-MaxBuffer allocated slightly more time supplements and at most 4.76 % for N3.

At the other extreme, MaxMin and HalfBuffer allocated significantly the most time supplements in the resulting timetables as these models do not include minimization of time supplements in the objective function. The most time supplements were assigned by HalfBuffer in N3, 35.36 %.

PESP-λ often had the most headways with no buffers which is expected due to the fact that the main goal was minimizing cycle time λ (and train-related times as the secondary objective) and consequently, headway arcs were kept as small as possible. This is exactly the main characteristic of the defined network stability/capacity measure.

PESP-λ reported 5, 27 and 36 zero buffers for instances N1, N2 and N3, respectively. And then in Stage 2 models, the available buffers, defined by $T - \lambda$, were distributed according to the given objective functions. All models reported less zero buffers than in Stage 1 (The only exception was HalfBuffer+ for N2 creating 39 zero buffers.). MaxMin found solutions with minimal buffer times of 30, 10 and 54 s, for N1, N2 and N3, respectively. However, these came at the expense of greater time supplements which are often bigger than for other objectives. The other models generated some zero buffers. MaxMin+ seemed to have a good balance between minimal buffer and time allowances and resulted in up to 7.27 % of average time supplements (for N2).

For the two-stage approach, solving PESP-λ for networks N1, N2 and N3 took 0.04 s, 0.03 s and 0.69 s, respectively. In addition, solving CPF-λ-s with different objective functions was always faster than its corresponding Stage 1 problem. For instance, CPU time for N1 was always up to 0.01 s, for N2 ranged from 0.03 to 0.06 s and for N3 – from 0.08 to 0.12 s. Comparative single stage models PESP-N always needed more CPU time than the two-stage ones. Most notably, N-MinBuffer was not able to reach the optimal solution within the given time limit and instead, the optimality gap of 10.36 % and 1.93 % was observed for N2 and N3, respectively. N-MinTrainTimes reported comparable CPU times for N1 and N2, while for N2 was sgnificantly larger, 14.45 s. So, it may be seen that the computation time of our two-stage approach may be even significantly lower compared to the single stage model PESP-N, particularly to N-MaxBuffer. This is a quite important observation because the existing approaches for solving robust TTP suggest quite extensive computational efforts.

5.5.4 Evaluating robustness of the two-stage approach

In order to gain more insight in the timetable robustness using different objective functions, we tested them on different disturbance scenarios that range from slight random delays to bigger and more often occurring ones. To this end, we evaluate the two-stage models MaxBuffer, MaxMin, MaxMin+, HalfBuffer, and HalfBuffer+, and compare with the single stage ones N-MinTravelTimes and N-MaxBuffer, by applying values for μ between 0 and 10% of the minimum process times. Clearly, $\mu = 0$ represents an undisturbed scenario. Figure 5.6 gives the obtained average delay $\bar{D}$ as the robustness cost for all objective functions. It can be seen that for small disturbances, $\mu < 3$, all designed timetables were able to withstand (most of) the disturbances. In all cases, the N-MinTrainTimes model generates the most delays as a result of having the least time supplements. On the other hand, HalfBuffer and MaxMin are the least affected by various disturbances due to the large time allowances. In particular, these two functions performed the same for N2, while HalfBuffer was better for instances N1 and N3. MaxBuffer, MaxMin+ and HalfBuffer+ performed similar for N1 and N2, while HalfBuffer+ tends to perform better and generates less delays for N3 and $\mu > 4$.

For network N3, N-MaxBuffer generated smaller average delay $\bar{D}$, compared to the two-stage solutions like MaxBuffer, MaxMin+. Mostly, this was a result of more time

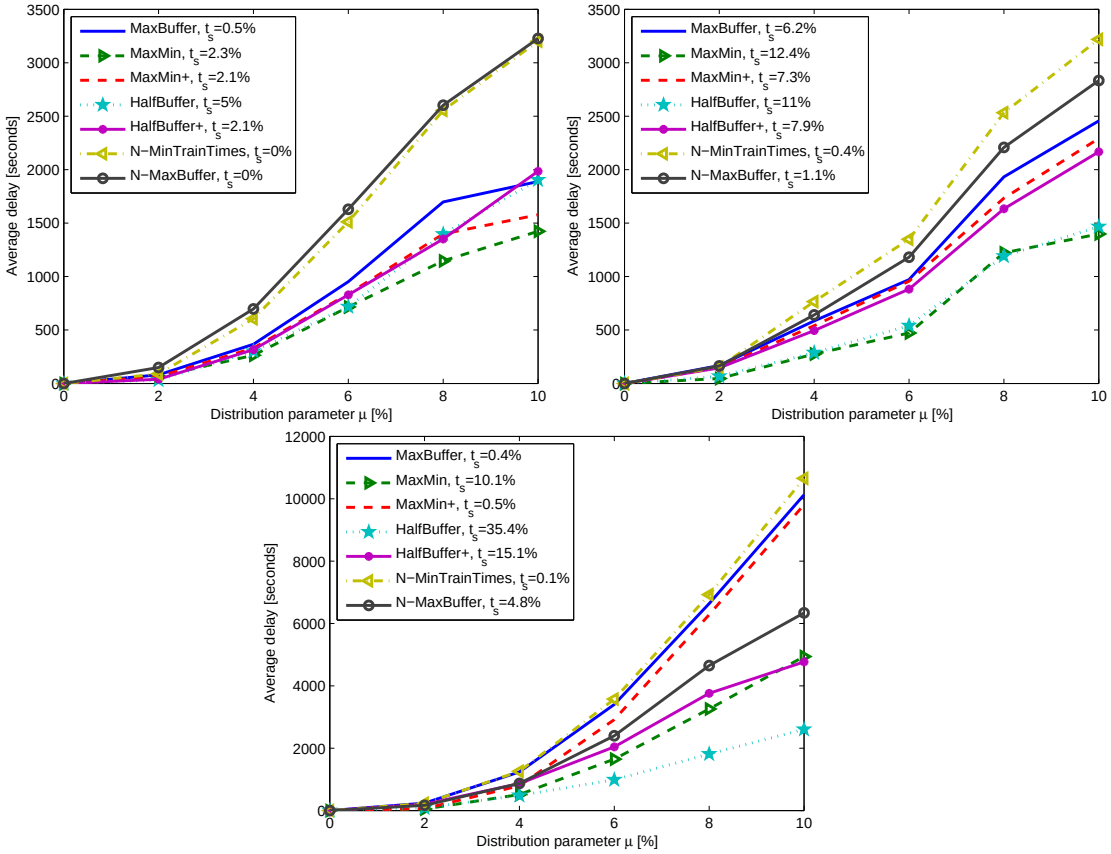

Figure 5.6: Comparison of timetable robustness: Average delay $\bar{D}$ for scenarios N1, N2 and N3 for distribution parameter $\mu = [0, 10]$

supplements in N-MaxBuffer compared to the latter two, i.e., 4.8% compared to 1.8% and 2.8%, respectively.

One of the main advantages of the stability-to-robustness approach is that by having the optimal stable timetable structure, the corresponding timetable allows a significant (i.e., maximal) amount of time allowances at the most critical cycle too. Such notion of stability is not recognized in PESP-N models. As a result, we may expect less delays for a timetable computed with our approach when disturbances occur at the critical cycle.

In order to show this advantage of the two-stage stability-to-robustness approach, we rerun the robustness evaluation by allowing disturbances only on the critical cycle. Figure 5.7 depicts the clear difference between single-stage PESP-N and the two-stage model, resulting in almost all initial disturbance being absorbed by the two-stage models. All computed solutions, managed to absorb the smallest disturbances $\mu \leq 1$. However, both PESP-N generated significant delays for bigger values of μ, while CPF-λ-s managed to absorb all delays while $\mu \leq 9$. Finally, for $\mu = 10$ only delays of several seconds were generated by all two-stage functions.

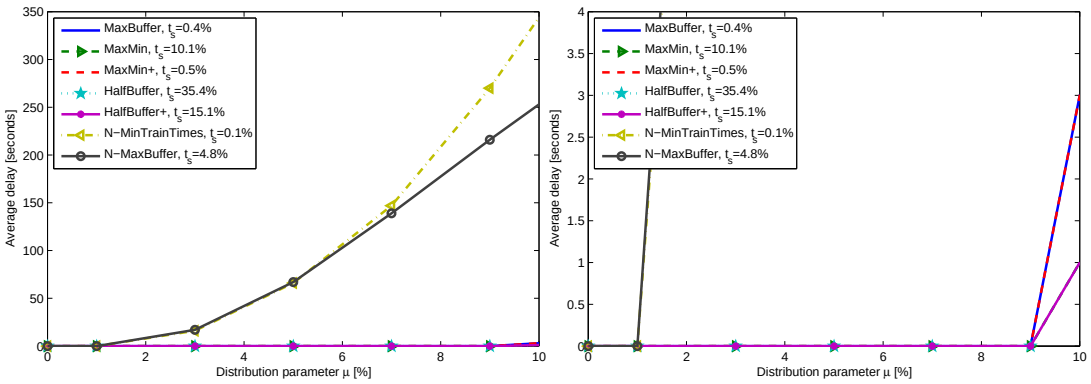

Figure 5.7: Comparison of timetable robustness: Average delay $\bar{D}$ for scenario N3 and varied distribution parameter μ on the critical cycle, complete (left) and zoomed (right).

5.5.5 Sensitivity analysis on time allowances' weights

The computed timetables may be affected by the weights assigned to time allowances. Thus, we perform a sensitivity analysis focusing on the network N3, where PESP-N computed more robust solutions for some cases. The purpose of this sensitivity analysis is twofold: first, to determine the behavior of objective functions in CPF-λ-s influenced by w_s, and second, to recognize promising values of w_s that could generate better timetable solutions.

The weight for buffer times w_b is assumed as fixed, while w_s takes values from the interval $[-30, 0.001]$. Table 5.4 gives the statistics of solutions for different values of w_s. The structure is similar to Table 5.3.

Figure 5.8 reports the allocated time supplements rate (in %) for given weights w_s. The bottom horizontal solution (solid star-marked line) is defined by the optimal stable timetable structure obtained by solving PESP-λ that was used as a fixed reference, and it represents the total time supplement of 0.43 %. Note that MaxMin and HalfBuffer are unaffected by w_s because time supplements are not included in these objective functions (MaxMin and HalfBuffer maximize only buffer times), and thus, we obtain horizontal lines that are independent of w_s.

For the remaining functions, MaxBuffer, MaxMin+ and HalfBuffer+, we see a reduction of allocated time supplements with a given greater importance which is driven by lowering w_s. The solution of MaxMin+ (HalfBuffer+) for $w_s = 0$ becomes equal to MaxMin (HalfBuffer) as these two functions become essentially the same. Note that the objective values ($objval$) between MaxMin and MaxMin+ (as well as HalfBuffer and HalfBuffer+) slightly differ since MaxMin+ (HalfBuffer+) still includes the weighted buffer times as the secondary objective. Even more, MaxMin+ included the multiplication factor as opposed to MaxMin. As soon as w_s becomes smaller than 0, the allocated time supplements become smaller, meaning that MaxMin+ and HalfBuffer+ are bounded from above by their (time supplement) non-weighted counterpart

Table 5.4: Sensitivity analysis of CPF-λ-*s* objective functions for network N3

Obj function	w_s	objval	Rtsupp	Dtsupp	Tsupps	avg Rtsupp Rate	Min Buffer	Zero Buffers	CPU time
MaxBuffer	-30	-2205	210	240	450	0.43	0	19	0.09
MaxBuffer	-10	-2205	210	240	450	0.43	0	19	0.09
MaxBuffer	-5	-2205	210	240	450	0.43	0	19	0.09
MaxBuffer	-4	-2205	210	240	450	0.43	0	19	0.09
MaxBuffer	-3	-2205	210	240	450	0.43	0	19	0.09
MaxBuffer	-2	-2205	210	240	450	0.43	0	19	0.08
MaxBuffer	-1	-2205	210	240	450	0.43	0	19	0.08
MaxBuffer	-0.8	-2205	211	240	451	0.44	0	19	0.11
MaxBuffer	-0.6	-2205	211	240	451	0.44	0	19	0.10
MaxBuffer	-0.4	-2214	292	240	532	0.50	0	19	0.09
MaxBuffer	-0.2	-2256	847	662	1509	1.32	0	20	0.09
MaxBuffer	-0.1	-2371	906	781	1687	1.52	0	15	0.09
MaxBuffer	-0.05	-2434	873	819	1692	1.91	0	16	0.18
MaxBuffer	-0.01	-2483	965	732	1697	1.75	0	18	0.11
MaxBuffer	0	-2496	7607	4554	12161	15.12	0	38	0.08
MaxMin+	-30	-54235075	331	268	599	0.54	54	0	0.14
MaxMin+	-10	-54248508	331	268	599	0.54	54	0	0.12
MaxMin+	-5	-54249254	331	268	599	0.54	54	0	0.11
MaxMin+	-4	-54249403	331	268	599	0.54	54	0	0.12
MaxMin+	-3	-54249552	331	268	599	0.54	54	0	0.11
MaxMin+	-2	-54249702	331	268	599	0.54	54	0	0.12
MaxMin+	-1	-54249851	331	268	599	0.54	54	0	0.12
MaxMin+	-0.8	-54249881	331	268	599	0.54	54	0	0.12
MaxMin+	-0.6	-54249910	331	268	599	0.54	54	0	0.12
MaxMin+	-0.4	-54249940	331	268	599	0.54	54	0	0.12
MaxMin+	-0.2	-54249970	331	268	599	0.54	54	0	0.12
MaxMin+	-0.1	-54249985	331	268	599	0.54	54	0	0.12
MaxMin+	-0.05	-54249993	331	268	599	0.54	54	0	0.12
MaxMin+	-0.01	-54249999	331	268	599	0.54	54	0	0.11
MaxMin+	0	-54250000	6328	4421	10749	10.11	54	0	0.12
HalfBuffer+	-30	1048805	210	281	491	0.43	0	15	0.13
HalfBuffer+	-25	1048400	292	317	609	0.52	0	15	0.12
HalfBuffer+	-20	1047453	353	325	678	0.64	0	15	0.12
HalfBuffer+	-10	1037498	1124	1668	2792	1.94	0	35	0.12
HalfBuffer+	-7	1027993	1886	2278	4164	3.47	0	33	0.13
HalfBuffer+	-6	1023950	2413	2376	4789	4.47	0	26	0.12
HalfBuffer+	-5	1019363	2842	2417	5259	5.73	0	24	0.13
HalfBuffer+	-4	1014342	3215	2737	5952	6.60	0	26	0.12
HalfBuffer+	-3	1008078	4135	3174	7309	8.48	0	30	0.13
HalfBuffer+	-2	1000600	5184	3743	8927	10.51	0	29	0.13
HalfBuffer+	-1	990553	8113	5527	13640	15.11	0	32	0.14
HalfBuffer+	-0.8	987400	9237	7590	16827	17.35	0	33	0.13
HalfBuffer+	-0.6	984070	9885	7415	17300	18.39	0	33	0.14
HalfBuffer+	-0.4	980641	9886	7981	17867	18.78	0	34	0.19
HalfBuffer+	-0.2	977022	10476	8700	19176	19.54	0	36	0.14
HalfBuffer+	-0.1	975127	10603	9027	19630	20.18	0	36	0.15
HalfBuffer+	-0.05	974168	10941	8689	19630	20.67	0	36	0.13
HalfBuffer+	-0.01	973401	10693	8937	19630	20.19	0	36	0.14
HalfBuffer+	0	973209	15278	12836	28114	35.36	0	37	0.12

functions (i.e., MaxMin and HalfBuffer). In addition, since the optimal stable timetable structure computed in Stage 1, defines the input for Stage 2, solutions from CPF-λ-s can never be lower (i.e., have less time supplements) than the one computed in Stage 1, i.e., all solutions are bounded with the output of PESP-λ.

All three w_s-dependent functions, MaxBuffer, MaxMin+ and HalfBuffer+, have decreasing behavior of allocated time supplements with increased importance of the corresponding weights. However, they all have different degrees of steepness. MaxBuffer has a steep reduction of time supplements for $w_s = [-0.6, 0)$, while for $w_s < -0.6$, the amount of time supplements is equal to PESP-λ, i.e., 0.43%. Thus, MaxBuffer seems very sensitive to w_s over a small range very close to 0. Such behavior could be due to the buffer times in MaxBuffer being assigned weights that are of the same scale as w_s ($w_b = 0.1$) and thus, the sums of weighted time supplements and buffer times are rather comparable as well. On the other hand, as soon as w_s reduced below -0.6, the sum of weighted time supplements overweighs the buffer times. MaxMin+ seems rather insensitive to w_s values between $(0, -100]$ and always generated the same amount of time supplements equal to 0.54 %. This may be due to the fact that we applied the multiplication factor of 10^7 to the minimal buffer time z compared to the weighted sum of time allowances. Only for $w_s = 0$, the solution of MaxMin+ equals the one of MaxMin. In addition, we ran MaxMin+ for $w_s = -10^8$ and obtained the solution equal to PESP-λ. HalfBuffer+ has a more gradual decrease compared to MaxBuffer, as it generates significantly different solutions for $w_s = [-15, -0.01]$ which allocated time supplements ranging from 0.64 % to 20.67%. When lowering w_s below -15, the amount of time supplements changes only minimally and it finally reaches the lower bound, imposed by PESP-λ, for $w_s = -30$. Solutions having time supplements bigger than 15% may be considered inefficient, meaning that they include abundant time supplements, while the ones having smaller than 5% may not be robust enough, i.e., having insufficient time supplements. Thus, the most promising weight values for HalfBuffer seem to fall between -10 and -1.

Due to the degree of steepness, ie., sensitivity to w_s, HalfBuffer+ seems preferable objective function over MaxBuffer (and MaxMin+) to provide significantly different timetables for various w_s.

These observations bring us to the second question in the analysis: *which values of w_s produce a timetable with a good trade-off between efficiency and robustness?* To that purpose, we undertook the robustness analysis of the timetables obtained by applying different w_s while keeping w_b fixed. Based on the results in Table 5.4 and Figure 5.8, we evaluated MaxBuffer and HalfBuffer+ with additional values for w_s. In particular, we used MaxBuffer with $w_s = -0.05$, and HalfBuffer+ with $w_s \in \{-30, -5, -3, -2\}$ Figure 5.9 reports the functions MaxBuffer and HalfBuffer+ and N-MinTrainTimes and N-MaxBuffer. MaxBuffer (HalfBuffer+ with $w_s = -30$) allocated only 1.9% (0.4 %) of time supplements and thus, generated big average delays for all μ. On the other hand, HalfBuffer+ with $w_s = \{-5, -3, -2\}$ demonstrated results comparable and even better than N-MaxBuffer. These three solutions were also comparable to N-MaxBuffer

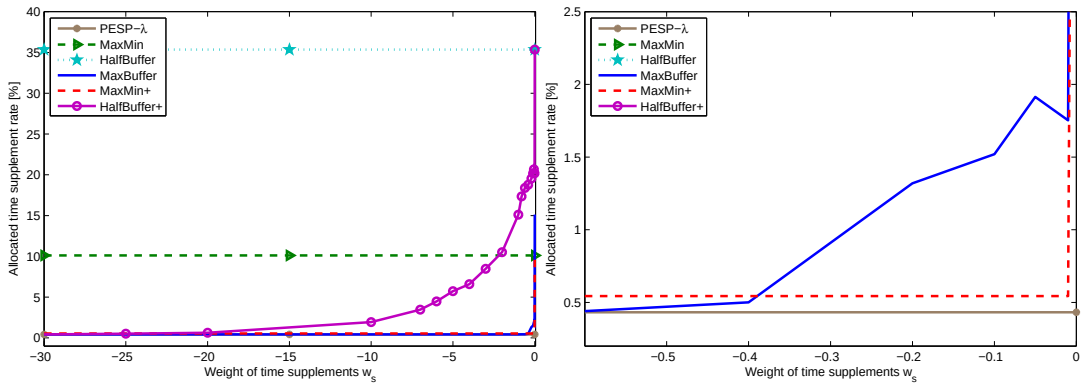

Figure 5.8: Comparison of timetables: Allocated time supplements for $w_s = [-30, 0]$ (left) and zoomed to $w_s = [-0.6, -0.001]$ (right)

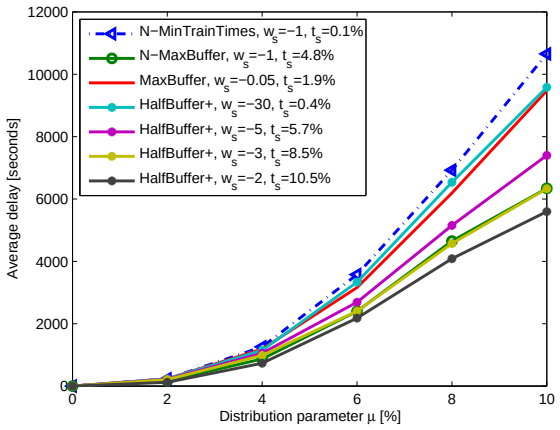

Figure 5.9: Comparison of timetables: disturbance scenarios vs average delay for variable w_s and functions MaxBuffer and HalfBuffer+ and N-MinTrainTimes and N-MaxBuffer.

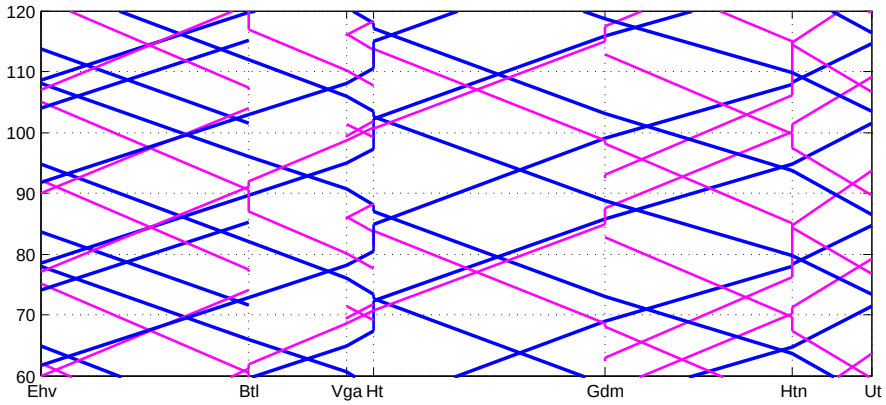

Figure 5.10: Time-distance diagram for corridor Ehv-Ut with HalfBuffer+ and $w_s = -2$ (for N2)

in terms of allocated time supplements. Finally, Figure 5.10 gives the time-distance diagram for HalfBuffer+ with $w_s = -2$.

Based on these results, we may conclude that HalfBuffer+ seems to be a preferable choice for the objective function for the stability-to-robustness model that tend to generate the best solutions with a good trade-off between efficiency and robustness. In addition, it allows flexibility to compute provide significantly different timetables when time supplements weight is varied. During the timetable planning, this is particularly important when different timetable alternatives need to be generated and afterwards evaluated to determine the best performing timetabling solution. However, it is important to gain thorough understanding of each objective function behavior. And only then, by adjusting given weights, one could get the most satisfying solutions that are both efficient and robust. In addition, a choice of the objective may even differ for different networks. In general, the choice should be made according to a range of factors, such as the desired robustness against the amount of expected delays, acceptable amount of time supplements and/or allowed upper bounds for process times. Also, the weights of buffer times and time supplements should be considered carefully.

5.6 Conclusion

In this paper, we studied the robust train timetabling problem in railway networks. We proposed a two-stage stability-to-robustness model that incorporated three important parameters of timetable design: efficiency, stability and robustness. The first stage focused on stability, the second stage on robustness, while efficiency was considered in both stages. For the first time, a timetable model explicitly considered the infrastructure capacity occupation to obtain a stable and robust timetable. An alternative formulation for optimally allocating time allowances was introduced. Five objective functions (MaxBuffer, MaxMin, MaxMin+, HalfBuffer, HalfBuffer+) were defined to generate timetables which were evaluated a posteriori and compared with existing single-stage PESP models.

The two-stage stability-to-robustness model was tested on real-life Dutch railway networks. The produced timetables were, in most cases, better than the ones computed by existing single-stage PESP models, meaning that they generated smaller amount of average delays. MaxMin and HalfBuffer tend to generate solutions that incorporate an excessive amount of time supplements and as such may be inefficient for passengers. MaxBuffer, MaxMin+ and HalfBuffer+ usually created the most robust solutions that were also efficient, that is, only limited time supplements were allocated. We also showed that a thorough analysis of the models' weight factors should be considered to generate the best trade-off between efficiency and robustness. Objective function MaxBuffer+ seems too sensitive to changing weights, while HalfBuffer+ tends to allow more flexibility to generate different solutions. Given these observations, HalfBuffer+ seems to be the most promising objective function for the stability-to-robustness approach.

The reported computational times for the two-stage approach were significantly lower compared to the single stage timetabling model.

Future work could include testing the current model on bigger instances. Consequently, techniques to speed up the computation times may be necessary. Also, a thorough analysis of model parameters and weight factors should be considered to generate the best solutions. In addition, the model can be used to identify bottlenecks and suggest possible infrastructure improvements. Finally, the computational efficiency leaves more time to planners to generate different alternative timetables and evaluate them before implementing in practice. Although tested only on limited instances, our two-stage model could help improving timetable robustness.

Chapter 6

An integrated micro-macro approach to robust railway timetabling

---

This chapter has been published as:

Bešinović, N., Goverde, R. M. P., Quaglietta, E., & Roberti, R. (2016). An integrated micromacro approach to robust railway timetabling. *Transportation Research Part B: Methodological*, 87, 14-32.

6.1 Introduction

The recent growth in the demand for railway passengers and freight encourages infrastructure managers to improve efficiency of their networks in terms of higher infrastructure occupation and service quality (e.g., punctuality and travel times). Upgrading the infrastructure or the signalling system may help to achieve these objectives with the downside of massive financial investments. A cost-effective alternative is represented by designing effective timetables that can absorb everyday variations in running and dwell times while exploiting network capacity as much as possible. This means allocating as many trains as possible to the available infrastructure while guaranteeing sufficient time allowances (i.e., supplements and buffer times) to reduce delay propagation. In this context, timetable design must rely on accurate computations of realisable train paths and buffer times. Only a robust construction of train time-distance paths allows designing dense timetables which are operationally feasible, i.e. free from track conflicts including constraints imposed by the infrastructure layout and the safety and signalling systems.

In the literature, timetabling problems are most commonly modelled at a macroscopic level, usually referred to as the Periodic Event Scheduling Problem (PESP) (Serafini & Ukovich, 1989) or Train Timetabling Problem (TTP) (Caprara et al., 2002). Cacchiani

and Toth (2012) gave an extensive review of variants of PESP and TTP, covering both nominal and robust approaches. Most of the models assume a macroscopic infrastructure, and as such neglect microscopic details (e.g. signal position, switches) important for accurate timetabling. These models tend to use predefined norms for timetabling constraints such as default time values for train separation at stations and as such cannot guarantee timetable feasibility in practice. Therefore, macroscopic models must be upgraded or integrated with more detailed models if operational feasibility of the timetable must be ensured. To this end, different approaches have been proposed in the literature based on a hierarchical integration of timetabling models with different levels of detail. Schlechte et al. (2011) presented a bottom-up approach which first aggregates microscopic running and headway times to be used by a macroscopic model that identifies an optimised timetable for a given utility function, and then checks its feasibility by simulating it at a microscopic level. Middelkoop (2010) described the tool ROBERTO which uses a microscopic infrastructure model to compute accurate running and headway times which are then input to the macroscopic timetabling model DONS (Kroon et al., 2009). De Fabris et al. (2013) introduced a mesoscopic timetabling model which simplifies representation of station layouts to combine fast computation of macroscopic models with the accuracy of microscopic models. Caimi, Fuchsberger, Laumanns, and Schüpbach (2011) extended PESP by proposing the flexible periodic event scheduling problem (FPESP), where intervals are used instead of fixed event times. By applying FPESP, the output does not define a final timetable but an input for finding a feasible timetable on a microscopic level (Caimi et al., 2011).

The main shortcomings with these approaches are that some do not perform any feasibility check of the timetable produced (De Fabris et al., 2013; Kroon et al., 2009), while others do not consider any iterative modification to the timetable if it is proved infeasible at the microscopic level (Caimi et al., 2011, Schlechte et al. (2011)). Another way of using a microscopic model to improve the outcome of a macroscopic model has been developed for the purpose of real-time railway traffic management. Kecman, Corman, D'Ariano, and Goverde (2013) tested the behaviour of various macroscopic models and compared them with a microscopic one in order to determine the level of detail and operational constraints that are necessary to incorporate at the macroscopic level. Within the European project ON-TIME (Optimal Network for Train Integration Management across Europe), we have developed a hierarchical iterative tool for the optimized design of railway timetables which combines microscopic, macroscopic and fine-tuning models (Goverde et al., 2016; ON-TIME, 2016). In this paper, we focus on the integration of a microscopic and a macroscopic model, which interact by iteratively updating macroscopic parameters that are re-computed at the microscopic level until the produced timetable is proved feasible, stable, and robust. This micro-macro timetabling approach has been applied to a real case study in the Netherlands. Experimental results show that our algorithms compute in short time a high quality timetable having an infrastructure occupation that satisfies UIC recommendations on capacity norms.

The main contributions presented in this paper are:

- A new timetabling approach that for the first time incorporates robustness in a micro-macro framework,
- An integrated iterative approach for computing a microscopically conflict-free and stable timetable that is optimized at a network level,
- An automatic procedure to adapt macroscopic input by constraint relaxation and tightening methods at the microscopic level,
- A macroscopic timetable optimization model with a post-processing Monte Carlo stochastic robustness evaluation of the generated timetables.

The remainder of the paper is organized as follows. The next section defines the problem statement of this paper and introduces the framework of the micro-macro approach. Section 6.3 describes the network and data modelling at different level of details and the automatic transformations between these networks. Section 6.4 and 6.5 elaborate on the microscopic and macroscopic models of the approach, while Section 6.6 describes the applied constraint adaptations between successive iterations. The case study is presented in Section 6.7, while conclusions are provided in Section 6.8.

6.2 Problem description

We adopt the definitions from (Goverde et al., 2016). A line service is defined with origin and destination points, stopping pattern, i.e. served timetable points (stations, stops), and a corresponding rolling stock type. It also includes the service category, such as local or intercity, and the frequency represented in number of trains per hour. A train path is the infrastructure capacity needed to run a train between two places over a given time period (EC, 2001). A conflict is determined as an overlap (in time and space) between two train paths and entails that one train cannot use the railway infrastructure without interfering with another train. Timetable efficiency reflects the amount of time allowances in the scheduled travel times (running, dwell and transfer times) which must be as short as possible to provide short journey times and seamless connections. Timetable feasibility is the ability of all trains to adhere to their scheduled train paths. A timetable is feasible if (i) the individual processes are realisable within their scheduled process times, and (ii) the scheduled train paths are conflict free, i.e., all trains can proceed undisturbed by other traffic. Timetable stability is the ability of a timetable to absorb initial and primary delays so that delayed trains return to their scheduled train paths without rescheduling. This is directly connected with the infrastructure occupation rate. The higher this rate, the lower are the time allowances and hence the less stable is the timetable. Timetable robustness is the ability of a timetable to withstand design errors, parameter variations, and changing operational conditions.

We distinguish between a microscopic and macroscopic timetable. A macroscopic timetable, or MacroTT, includes scheduled running, dwell and transfer times, as well as event times such as arrivals, departures and passages between and at relevant timetable points (introduced in Section 3.1). A microscopic timetable, or MicroTT, includes scheduled running, dwell and transfer times as well as event times such as arrivals, departures and passages for microscopic timetable points (introduced in Section 6.3.1) and the corresponding train speed profiles defining the exact train trajectories. A MacroTT is the outcome of the macroscopic model and is analysed by the microscopic one, while the MicroTT is the final output of the developed framework.

Given infrastructure and rolling stock characteristics, and the requested line services, the Train Timetabling Problem (TTP) consists of finding a feasible, efficient, stable and robust microscopic timetable.

6.2.1 The timetable planning framework

We propose a micro-macro framework to solve the TTP and design a railway timetable. The structure of such a framework is shown in Figure 6.1, which indicates the interactions among the different models, their functions as well as the input-output data flow. The framework is implemented using a standardized RailML interface (Bosschaart et al., 2015; RailML, 2015). In particular, the RailML data required for the initialization of the models relate to characteristics of the infrastructure (i.e., gradients, speed limits, positions of stations, switches), rolling stock (i.e., mass, length, braking rate, tractive effort-speed curve), interlocking (i.e., alternative local routes), signalling system (e.g., position and type of signals, automatic train protection (ATP) behaviour), and routes/stopping pattern of the train services to be scheduled. Both input and output data of the framework are in RailML format, which is being developed with the goal of becoming a standard in Europe for communication among railway software tools.

Timetabling computation is an iterative process of two different models: a microscopic and a macroscopic model. The microscopic model computes reliable train running and headway times at a highly-detailed local level and checks for feasibility and stability of the timetable. The macroscopic model has an aggregated infrastructure representation and produces a timetable for the entire network by identifying arrival/departure times at/from stations or junctions which optimize a given objective function (e.g., minimize journey times). Such a macroscopic model includes methods for estimating delay propagation to assess produced timetables in terms of robustness versus stochastic operation disturbances.

In the first iteration a timetable is not available yet, so the microscopic model computes minimum running times and headways that are sent to the macroscopic model to calculate a timetable. The achieved macroscopic timetable is sent back to the microscopic model that, based on the operational running times (i.e. the running times including time supplements scheduled by the macroscopic timetable), computes train blocking times necessary for detecting track conflicts. If there are track conflicts, these

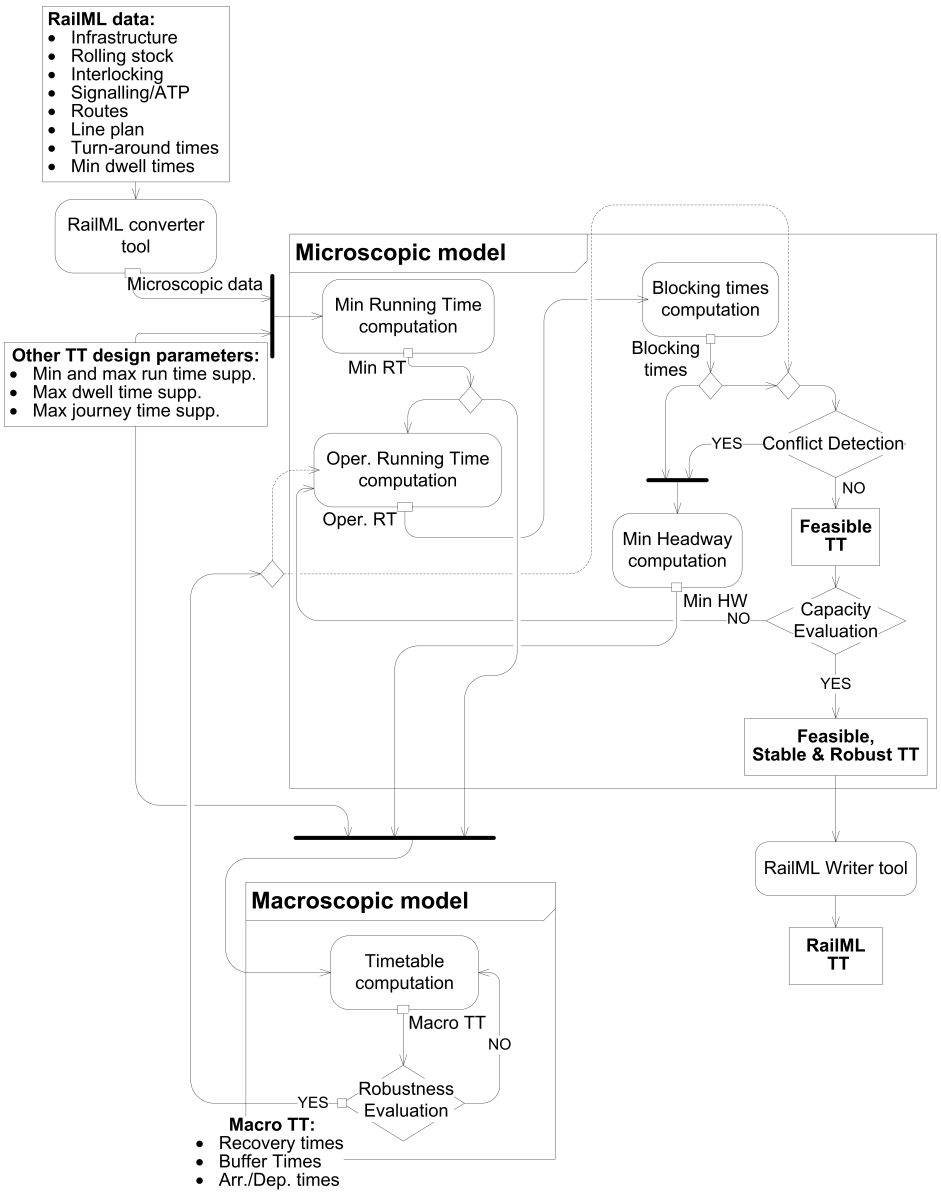

Figure 6.1: Functional scheme of the micro-macro framework

are solved and new headways and running times are computed and transferred to the macro model again. The macroscopic model solves an optimization problem which incorporates heuristics with an integer linear programing problem minimizing a weighted sum of running, dwell and transfer times, and a robustness cost. The robustness cost is defined as the delay settling time obtained from a Monte Carlo simulation of the delay propagation for given candidate timetable solutions. This iterative process is repeated until no more track conflict is detected and the timetable is thus feasible at both the macroscopic and microscopic levels. Once feasibility is achieved, the microscopic model evaluates the stability of the timetable (i.e., the capability in absorbing delays). If the timetable is not stable, new operational running times are computed by e.g. increasing the value of time supplements and/or buffer times. This is performed until timetable stability is verified to have reached the required level (UIC,

2013). Transformations from the microscopic level to the macroscopic and vice versa require appropriate procedures that have been developed to aggregate/disaggregate input and output data. This interaction continues until the timetable produced by the macroscopic model is proved to be microscopically feasible and stable. As a result, the final output of the framework is a feasible and stable timetable with a suboptimal trade-off between efficiency and robustness.

6.3 Network and data modelling

In our approach, input and output data are exchanged between models with different levels of detail, so consistency in data flows must be guaranteed. This is achieved by automatic data transformation (aggregation/disaggregation) processes that we describe in this section.

6.3.1 Network representation

The hierarchical framework for timetable design is composed of two models that represent the same network but with a microscopic and a macroscopic level of detail.

The microscopic graph $G = (X, B)$ represents the network at the level of homogeneous behavioural sections (i.e. sections with constant values of speed limit, gradient and curvature radius). Each infrastructure section (arc), $b_i \in B$, is described with constant characteristics of speed limit v_i, gradient g_i and radius ρ_i and given length l_i, i.e., $b_i = (v_i, g_i, \rho_i, l_i)$. Microscopic nodes $x \in X$ represent both points in which these characteristics change and infrastructure elements like block section joints, switches, and station platforms. This detailed microscopic model is used to aggregate the homogeneous behavioural sections in block sections for open tracks and in track-free detection sections for interlocking areas which are required for considering sectional-release route-locking behaviour. This level of detail is important for computing blocking times, conflict detection, and infrastructure occupation.

On top of the microscopic network, a discrete set of microscopic timetable points $K \subset X$ is defined. A *microscopic timetable point* $k \in K$ represents an infrastructure point where an interaction exists between a train and passengers (boarding and alighting) or cargo (loading and unloading), or between two trains (converging and diverging tracks). These microscopic points therefore define important discrete events at stations, junctions, bridges or tunnels.

The macroscopic network is represented by a multi-graph $N = (S, A \cup E)$, where the vertices represent a set $S \subseteq K$ of macroscopic points corresponding to the microscopic timetable points that allow interaction between trains, i.e., meeting, overtaking, or connections. In the remainder, we refer to macroscopic points as *timetable points*. The macro tracks are defined as mono-directional $a = (s_1, s_2)$ or bidirectional $e = s_1, s_2$. The sets of arcs A and edges E represent mono-directional and bidirectional tracks between pairs of timetable points, respectively. For each timetable point s_i inS, the

capacity cap_{s_i} for each station is known and corresponds to the number of tracks. For each pair of timetable points $s_i, s_j \in S$, we consider that the number of directed arcs (mono-directional tracks) $\alpha_{s_i s_j}$ from i to j, the number of directed arcs $\alpha_{s_j s_i}$ from j to i, and the number of edges (bidirectional tracks) $\beta_{s_i s_j}$ are known.

Inputs from the basic architecture consists of a set of data expressed in RailML format, namely: a) Microscopic infrastructure data, b) rolling stock data, including train formations, c) Interlocking, signalling and ATP, d) available routes, and e) train line requests. These data are converted to a suitable internal format of ASCII data that is used by the microscopic computation models. Additional parameters, such as connections, dwell times, timetable design parameters and quality norms are provided externally.

6.3.2 Trains, train lines and routes

Let T be the set of all trains in the network and L be the set of all lines of trains, i.e. $L = L_1, L_2, , L_{\bar{l}}$, where $\bar{l} = |L|$, $L_j \subseteq T$, $j = 1, \cdots, \bar{l}$, $\bigcup_{j=1}^{L_j} = T$, and $L_j \cap L_k = \emptyset$, $1 \leq j$, $k \leq |L|$, $j \neq k$. For each line L_j, $j = 1, \cdots, \bar{l}$, the period time per_j is given, representing the ideal interval time between the stops at each station of two consecutive trains from the same line. Moreover, let $S_j \subseteq S$ be the set of stations served by line $j \in L$. For each train $t_j \in T$, $l(j)$ indicates the corresponding line. We assume that the trains of a given family $L_j = t_1, t_2, , t_{|T_j|}$, $j = 1, \cdots, \bar{l}$, are ordered in increasing order of departure times.

Moreover, we assume that for each train, its route ρ_t (i.e., the sequence of traversed tracks without the corresponding travel times) is provided. Finally, we differentiate between a microscopic route $\rho_t^{micro} = (b_1, b_2, \cdots, b_{n_t})$, where n_t is the number of microscopic arcs, and a macroscopic one $\rho_t^{macro} = (a_1, a_2, \cdots, a_{m_t})$, where m_t is the number of macroscopic tracks.

For each train $t \in T$ and each arc $a \in A$, (each edge $e \in E$, resp.) the minimum running time r_{ta} (r_{te})), the nominal running time r_{ta} (r_{te}), and the maximum running time r_{ta} (r_{te})) are given. All running times are computed by microscopic algorithms, while the nominal and maximum ones are given as input to the macroscopic module. Also provided for each train $t \in T$ is the maximum journey time $\bar{J}_t$ from origin to destination.

The algorithms developed within the ON-TIME project, both microscopic and macroscopic, are suitable for both periodic and non-periodic timetabling. In the non-periodic case, each line contains exactly one train. Therefore, the microscopic tools may accept both trains and train lines without any adjustment of the algorithms.

6.3.3 Other parameters

For each train $t \in T$ and each station $s \in S_t$, the nominal dwell time w_{ts} and maximum dwell time $\bar{w}_{ts}$ is provided. Let Q be the set of all connections between pairs of trains at given stations. Then each connection $q = (t_1, t_2, s)$ with $t_1, t_2 \in T$, $s \in S$, is characterized by a nominal and a maximum connection time, u_q and $\bar{u}_q$, respectively, which need to be respected by connection constraints. Since the aim of timetable planning

is to provide an acceptable quality of service, certain design norms need to be predefined. The set of these parameters consists of minimum and maximum transfer times, turnaround times, minimum and maximum running time supplements (%), and maximum allowed journey times of train lines (%). The set of timetable design norms is named Λ.

6.3.4 Microscopic to macroscopic conversion

Microscopic data are migrated to the macroscopic level using the procedure described in Algorithm 8, which is comparable with the one implemented by Schlechte et al. (2011). The conversion from microscopic to macroscopic is instead done in two steps. First, the subset S from K is derived. The algorithm compares all pairs of train lines. The macroscopic nodes are chosen based on the interplay between train routes. The set S includes only microscopic timetable points if two train routes are converging, diverging or crossing. This process of defining the set of timetable points is done automatically. Second, it consists of the aggregation of microscopic arcs b_i to macroscopic arcs a or edges e, $a = (b_1, b_2, \ldots, b_n)$ or $e = (b_1, b_2, , b_n)$. For each arc a and edge e the following data is determined: 1) the number of tracks $(\alpha_{ij}, \beta_{ij})$ by identifying different routes between two nodes using function *DetermineTracks*, and 2) the orientation (mono- or bidirectional) for each of them by function *DetermineDirection*.

We will describe the inclusion into S with an example. Consider two train lines that use the same successive microscopic timetable points k_1, k_2, k_3. Then the set S will include k_1, k_3, since k_1, k_3 are the origin and destination of both lines, while $k_2 \notin S$ as both trains use the same points before and after this. As another example, consider two train lines, where the first train line uses k_1, k_2, k_3 and the second one k_1, k_2, k_4, then S will be equal to k_1, k_2, k_3, k_4 since the train lines diverge in k_2, which makes $k_2 \in S$.

After having initialised the macroscopic network, the microscopic model is activated to compute microscopic minimum running times (over homogenous behavioural sections), blocking times and minimum headways. Headways are determined for all possible interactions between each two train routes (inbound-inbound, in-out, out-in, out-out) at every macroscopic timetable point s. The last two are executed on the block section level of the infrastructure network.

Once all process times are computed on the microscopic model, we carry out the aggregation of process times and the discretisation of time. The function *AggregateProcessTimes* is introduced to provide mitigation from microscopic running times (i.e., between any consecutive microscopic points) to aggregated process times between any consecutive timetable points in the macroscopic network. Since the macroscopic model uses a coarser time granularity, the time discretisation of process times is performed as well. The incorporated function represents an innovative rounding method that has the objective to control the rounding error by combining rounding up and rounding down. By applying AggregateProcessTimes, we obtain all process times that are necessary for macroscopic computation.

Algorithm 8 Microscopic to macroscopic conversion

Input: Microscopic network, microscopic timetable points, dwell times, timetable design norms, set of train lines
Output: $N = (S, A \cup E)$, process times
For all microscopic timetable points $k \in K$
  For all pairs of train lines
    If k is origin or destination point OR lines converge OR lines diverge OR lines cross
      Include k in the set of macroscopic timetable points
    End If
  End For
End For
For all adjacent timetable points
  DetermineTracks
  DetermineDirection
  If track is used in both directions
    Create edge e
  Else
    Create arc a
  End If
End For
Running times computation
Blocking times computation
For all macroscopic timetable points $s \in S$
  Minimum headway computation
End For
AggregateProcessTimes

6.3.5 Macroscopic to microscopic conversion

To convert data from the MacroTT to the MicroTT, we use the process described in Algorithm 9. Substantially, from the scheduled event times for the macroscopic timetable points we have to reconstruct the corresponding train trajectories and scheduled times for all of the microscopic timetable points. The details of each module used in this description are given in Sections 6.4 and 6.5.

Starting with MacroTT, we determine the scheduled running time over each macroscopic edge (arc), as the difference between the arrival in one station and the departure from the preceding station. Further, we compute the corresponding allocated running time supplement ψ_t as the difference between the scheduled and minimum running time for each train $t \in T$, where T is the set of all trains. This defines a vector Ψ_t of the time supplements ψ_t over each two macroscopic timetable points. For each train $t \in T$ and the corresponding Ψ_t we compute the operational running time consisting of the detailed train trajectory and scheduled times at microscopic timetable points, which are used for further microscopic analyses. Consequently, the blocking times are computed for all trains.

Algorithm 9 Macroscopic to microscopic conversion

Input: microscopic network, MacroTT
Output: Feasible, stable and robust timetable
For all trains **do**
    Determine allocated running time supplements
    Compute operational running times
    Compute blocking times
End For
Conflict detection
If timetable is not conflict-free
    Constraint Tightening
    Run Macroscopic optimization
Else
    Capacity evaluation
    If capacity norms are not satisfied
    Constraint Relaxation
    Run Macroscopic optimization
    End If
End If

Once blocking times are computed, conflict detection and capacity assessment are performed, and if one of those is not satisfied then train process times (headways and running times) are updated by applying constraint tightening and/or relaxation (cf. Section 6.6). Updated process times are sent again to the optimisation model described in Section 6.6 to re-compute MacroTT. After a new MacroTT has been obtained, Algorithm 9 starts from the beginning. Once MacroTT satisfies an acceptable quality of service, the algorithm terminates. The final output of the model is a feasible, stable and robust timetable.

6.4 Microscopic timetabling

The microscopic module consist of the following computation functions: minimum and operational running times, blocking and headway times, conflict detection and resolution and capacity assessment. A description of each function is given in the next paragraphs.

6.4.1 Running times

The minimum running time is the time required for driving a train from one infrastructure point to another infrastructure point assuming conflict-free driving as fast as possible. In this section we will focus on the running time between two stations. Running time computation considers detailed characteristics of the infrastructure (i.e., gradients, speed limits, positions of stations, switches), rolling stock (i.e., mass, composition, braking rate, tractive effort-speed curve), signalling system (e.g., position and type of signals), and routes/stopping pattern of the train services to be scheduled.

Running times are computed for every line service by means of dynamic Newton's motion equations (Hansen & Pachl, 2014) according to the implementation described in Bešinović et al. (2013).

As already explained, MacroTT represents the scheduled running time between two macroscopic timetable points, which consists of the sum of the minimum running times and time supplements (usually 5%-7% of the minimum running time) to recover from statistical variations during real operations. This means that from the scheduled running time given by MacroTT we must be able to retrieve the corresponding microscopic train trajectory (speed-distance, time-distance diagrams) in the MicroTT that satisfies that scheduled running time (described in Section 6.3.5). Such train trajectories incorporate the available time supplements. The operational running time represents the recomputed train trajectory that satisfies the scheduled running time between two timetable points. This trajectory will exploit associated running time supplements by applying cruising with a speed lower than the maximum speed, and to do so we implemented a customised bisection algorithm (Bešinović et al., 2017).

6.4.2 Blocking times

The blocking time of a section of track (block section or interlocked route) is the time duration that the section is exclusively allocated to a train and therefore blocked to other trains. Blocking times are computed in function *Blocking times computation* by applying the procedure described in Hansen and Pachl (2014).

The blocking time of a train for a given block section is composed of the following components: setup time $t_{setup,ti}$ to set the route for the train approaching, sight and reaction time $t_{sight,ti}$ of the train driver when approaching the approach signal, approach time $t_{approach,ti}$ needed by the train to traverse the braking distance from the approach signal to the main signal, the running time $t_{block,ti}$ of the train to traverse block section i, the clearing time $t_{clear,ti}$ to clear the block section over the train length, and the release time $t_{release,ti}$ to release the route after the train clearance. After having provided all these terms the blocking time d_{ti} of the train t relative to block i is obtained as:

$$d_{ti} = t_{setup,ti} + t_{sight,ti} + t_{approach,ti} + t_{block,ti} + t_{clear,ti} + t_{release,ti}. \tag{6.1}$$

Each blocking time d_{ti} of section i by train t is specified from the start d^s to the end d^e of the blocking time. Hence, $d_{ti} = (d^s_{ti}, d^e_{ti})$.

6.4.3 Minimum headway times

The minimum headway time between two trains is the time separation that prevents the trains from having track conflicts with each other. Here we introduce the computation of one minimum headway at a timetable point $s \in S$. Let B_{ijs} be the set of shared blocks associated to routes of both trains i and j in timetable point s, d^e_{il} be the end of blocking time d_{il} and d^s_{jl} the start of blocking time d_{jl}. Let us assume that both trains have the

same reference event (i.e., departure, arrival or passing) time at s, e.g., equal to 0. Then the minimum headway h_{ijs} from train i to j in timetable point s is computed as

$$h_{ijs} = \max_{l \in B_{ijs}} (d^e_{il} - d^s_{jl}). \tag{6.2}$$

For each pair of trains $t_1, t_2 \in T$ and each timetable point $s \in S$, we compute the nominal headway time $h^{dd}_{t_1t_2s}$ ($h^{da}_{t_1t_2s}$) between the departure of train t_1 from timetable point s and the departure (arrival) of train t_2 from (at) timetable point s, and the nominal headway time $h^{ad}_{t_1t_2s}$ ($h^{aa}_{t_1t_2s}$) between the arrival of train t_1 at timetable point s and the departure (arrival) of train t_2 from (at) timetable point s. Any headway time is equal to 0 whenever the two trains do not meet at a timetable point.

6.4.4 Conflict detection

The aim of conflict detection is to verify the feasibility of the macroscopic timetable by checking: *a*) the absence of track conflicts and *b*) the realisability of scheduled process times (i.e., running times, dwell times, turnaround times). Track conflicts are detected as partial or full overlaps of the blocking times provided by the *BlockingTimesComputation* function. Therefore, conflict-freeness is tested comparing the interaction of scheduled blocking times for each pair of trains, i.e., checking the possible blocking times overlap between them. The blocking time overlap $c_{ij\varphi}$from train line i to j at corridor φ is computed similarly as the minimum headway times:

$$c_{ij\varphi} = \max_{l \in B_\varphi} (d^e_{il} - d^s_{jl}), \tag{6.3}$$

where B_φ is the set of successive blocks at corridor φ, and the scheduled start and end of the blocking times are used. If $c_{ij\varphi} > 0$ then a conflict exists. Usually, a corridor corresponds to a macroscopic arc (or edge). In this way, the whole network is considered by the conflict detection algorithm, and not only timetable points.

Realisability is tested by checking if the scheduled running and dwell times exceed the corresponding minimum technical values. Note that the macroscopic timetabling model is such that it always provides realisable scheduled times, so the realisability check can then be omitted.

6.4.5 Infrastructure occupation

Infrastructure occupation is defined as the time share needed to operate trains according to a given timetable pattern taking into account scheduled running and dwell times. The infrastructure occupation $\mu(\varphi)$ of corridor φ can be obtained by:

$$\mu(\varphi) = \sum_{\{(i,j) \in W_\varphi\}} h_{ij\varphi}, \tag{6.4}$$

with W_φ the cyclic pattern of successive train pairs (i,j) in corridor φ, and $h_{ij\varphi}$ the minimum line headway on this corridor. The latter is computed similarly as local minimum headway, where blocks at a corridor φ, instead of a timetable point s, are considered. A corridor may be equal to an arc (or edge) or comprise several adjacent arcs (edges), $\varphi = \cup_i a_i$. We compute the infrastructure occupation for each corridor $\varphi \in \Phi$, applying an algorithm based on max-plus automata theory (Bešinović et al., 2017; Gaubert & Mairesse, 1999).

6.5 Macroscopic timetabling

The macroscopic timetable optimization algorithm iteratively communicates with the microscopic module in order to achieve a timetable that is both macroscopically and microscopically feasible.

The macroscopic optimization module receives as input the following data from the microscopic calculation module: *a*) railway infrastructure aggregated at macroscopic level (including capacity of arcs/edges and macroscopic timetable points), *b*) a set of train lines to schedule with the corresponding routes, *c*) headway times between pairs of train lines meeting at timetable points, *d*) nominal and maximum running and dwelling times along the routes of each train line, and *e*) a set of connections (where a connection states that, at a given timetable point, the departing time of a train must be within a given time interval from the arrival time of the previous train).

The macroscopic timetable computation provides the microscopic module with a macroscopic timetable that consists of a set of paths (at most one for each train) and the indication of the trains that are cancelled – trains can be cancelled if all corresponding feasible paths violate some of the constraints defined on the network (e.g., headway times, capacity). A path of a train is an ordered sequence of tracks and provides, for each of the traversed tracks, the times where the trains enter and leave the track. Furthermore, Monte Carlo simulation is applied to obtain a timetable with improved robustness. The timetable provided by the macroscopic module is not only feasible, but also robust.

From an algorithmic point of view, the macroscopic timetable computation consists of a *randomized multi-start greedy heuristic* (hereby referred to as *MacroHeu*) that iteratively generates a set of feasible timetables and, among them, selects the one having the minimum cost. The cost of a timetable takes into account properly defined penalties plus a specific penalty that considers its robustness. To assess the robustness of a timetable, a number of different scenarios (each one featuring a randomly generated delay for each train) are considered and evaluated in terms of absorption of the delays.

The macroscopic timetable computation algorithm may be run several times in a loop with the microscopic module exchanging information based on tightening and/or relaxing constraints, in order to guarantee that the final macroscopic timetable is also feasible from a microscopic perspective. For this reason, MacroHeu needs to imple-

ment a relatively easy methodology that can provide a good solution within limited computing times (i.e., tens of seconds).

6.5.1 Optimization algorithm

The problem addressed by the macroscopic module can be formulated as an ILP model with four different types of binary and integer variables and an exponential number of constraints.

Let P_l be set of all feasible paths of each train of line $l = 1,\dots,\bar{l}$, where a path p is an ordered sequence of arrival and departure times for each timetable point of the set S_l, defined as $(\tau^D_{t_1 s_1 p}, \tau^A_{t_1 s_2 p}, \tau^D_{t_1 s_2 p}, \dots, \tau^D_{t_1 s_{|S_l|-1} p}, \tau^A_{t_1 s_{|S_t|} p}, \tau^D_{t_2 s_1 p}, \dots \tau^A_{t_{|T_j|} S_{|S_l|} p})$, where τ^D_{tsp} represents the departure time at timetable point $s \in S_l \setminus \{|S_l|\}$ of train t over path p and τ^A_{tsp} is the arrival time at timetable point $s \in S_t \setminus \{s_1\}$ of train t. A path p is feasible if it satisfies the following constraints:

- all timetable points of the set S_l are visited according to the given route ρ^{macro}_t ;
- the total maximum journey time $\bar{J}_t$ is not exceeded;
- for each $i = 1,\dots,|S_l|-1$, the difference between the arrival time $\tau^A_{ts_{i+1}p}$ at timetable point s_{i+1} and the departure time $\tau^D_{ts_i p}$ at timetable point s_i is at least r_{ta} and does not exceed $\bar{r}_{ta}$, with $a = (s_i, s_{i+1})$, that is $r_{ta} \le \tau^A_{ts_{i+1}p} - \tau^D_{ts_i p} \le \bar{r}_{ta}$;
- for each $i = 2,\dots,|S_l|-1$, the difference between the departure time $\tau^D_{ts_i p}$ and the arrival time $\tau^A_{ts_i p}$ at timetable point s_i is at least w_{ts_i} and does not exceed $\overline{w}_{ts_i}$, that is $w_{ts_i} \le \tau^A_{s_{i+1}p} - \tau^D_{s_i p} \le \overline{w}_{ts_i}$.

In addition, the following penalties are defined:

- π^{canc}_l: penalty paid for cancelling the trains of line $l \in L$;
- π^{runn}_l: penalty for each time unit of running time exceeding the nominal one for trains of line $l \in L$;
- π^{dwell}_l: penalty for each time unit of dwell time exceeding the nominal one for trains of line $l \in L$;
- π^{conn}_q: penalty for the connection time exceeding $\underline{u}_q$ for connection $q \in Q$;
- $\overline{\pi}^{conn}_q$: penalty for missing connection $q \in Q$.

The cost c_p of path $p \in P_l$ that is assigned to the trains of line $l = 1,\dots,\bar{l}$, is given by the running and dwell time exceeding the nominal ones penalized according to penalties $\pi^{runn}_l$and π^{dwell}_l, respectively, that is:

$$c_p = \pi^{runn}_l \sum_{t \in L_l} \sum_{i=1}^{|S_t|-1} \left(\tau^A_{s_{i+1}p} - \tau^D_{s_i p}\right) - \pi^{dwell}_l \sum_{i=2}^{|S_t|-1} \left(\tau^D_{s_i} - \tau^A_{s_i}\right).$$

Let $\mathcal{U}$ be the set of all paths cliques, where each path clique $U \in \mathcal{U}$ is a subset of the paths of all lines (i.e., $U \subseteq \bigcup_{l \in L} P_l$) which may have conflicts with each other. This means that at most one of such paths can be in the solution simultaneously because any pair of those paths violate constraints on headway times and/or station/arc capacity.

By introducing the following sets of variables:

- Binary variable x_p equal to 1 if path $p \in P_l$ of trains of line $l \in L$ is selected (0 otherwise),
- Binary variable ξ_l equal to 1 if all trains of line $l \in L$ are cancelled (0 otherwise),
- Integer variable y_q representing the connection time exceeding $\underline{u}_q$ for connection $q \in Q$ if connection $q \in Q$ is not missed,
- Binary variable χ_q equal to 1 if connection $q \in Q$ is missed (0 otherwise),

the problem addressed by the macroscopic module can be formulated as the following ILP:

$$\min \sum_{l \in L} \sum_{p \in P_l} c_p x_p + \sum_{l \in L} \pi_l^{canc} \xi_l + \sum_{q \in Q} \pi_q^{conn} y_q + \sum_{q \in Q} \overline{\pi}_q^{conn} \chi_q \tag{6.5}$$

such that

$$\xi_l + \sum_{p \in P_l} x_p = 1 \qquad l \in L \tag{6.6}$$

$$\xi_{l(t_1)} + \xi_{l(t_2)} \leq 2\chi_q \qquad q = (t_1, t_2, s) \in Q \tag{6.7}$$

$$\sum_{p \in P_{l(t_2)}} \tau_{tsp}^D x_p - \sum_{p \in P_{l(t_1)}} \tau_{tsp}^A x_p \geq \underline{u}_q - M\chi_q \qquad q = (t_1, t_2, s) \in Q \tag{6.8}$$

$$\sum_{p \in P_{l(t_2)}} \tau_{tsp}^D x_p - \sum_{p \in P_{l(t_1)}} \tau_{tsp}^A x_p \leq \underline{u}_q + y_q + M\chi_q \qquad q = (t_1, t_2, s) \in Q \tag{6.9}$$

$$\sum_{p \in U} x_p \leq 1 \qquad U \in \mathcal{U} \tag{6.10}$$

$$x_p \in \{0, 1\} \qquad p \in P_l, l \in L \tag{6.11}$$

$$\xi_l \in \{0, 1\} \qquad l \in L \tag{6.12}$$

$$y_q \in \left\{0, \overline{u}_q - \underline{u}_q\right\} \qquad q \in Q \tag{6.13}$$

$$\chi_q \in \{0, 1\} \qquad q \in Q \tag{6.14}$$

where M is a large enough number.

The objective function (6.5) guarantees that the timetable achieved minimizes the total cost, given by the sum of: (a) the cost of the paths selected; (b) the cost for cancelling trains; (c) the cost for exceeding the nominal connection time; and (d) the cost for

missing connections. Constraints (6.6) impose on the model that each train is either cancelled or scheduled. Constraints (6.7) ensure that if one or both trains corresponding to a connection are cancelled, then a penalty for missing the connection is paid. Constraints (6.8) state that if both trains of a connection are scheduled, then the difference between the corresponding departure and arrival times at the station where the connection takes place must be not less than $\underline{u}_q$. Constraints (6.9) trigger the penalty for exceeding the nominal connection time for each connection having both trains scheduled. The term $M\chi_q$ in constraints (6.8) and (6.9) is used to prevent counting the penalty of the exceeding connection time when the connection is missed. Constraints (6.10) are clique constraints that impose on the provided timetable to be conflict-free (i.e., no headway and capacity constraints are violated); notice that constraints (6.10) can also be used to model simultaneous arrival or departures of pairs of trains at given stations, if these have to be considered as hard constraints. Constraints (6.11)-(6.14) set the domains of the variables of the model.

6.5.2 The macroscopic heuristic

Algorithm 10 Step-by-step description of Macroscopic optimization

Input: macroscopic network, set of trains, routes, headway times, running times, dwell times, connections
Output: a feasible and robust macroscopic timetable $MacroTT$ of cost $c^{MacroTT}$
Initialize $MacroTT := \emptyset$ and $c^{MacroTT} := \infty$
For $iter = 1, \ldots, \overline{iter}$ **Do**
  Initialize $CurrTT := \emptyset$, $c^{CurrTT} := 0$, and $LeftLines := L$
  While $LeftLines \neq \emptyset$ **Do**
    Randomly select a line l from $LeftLines$
    Determine (see Appendix A) the min-cost path $p \in P_l$ for line l that does not conflict with any of the paths of the set $CurrTT$
    If a path $p \in P_l$ was found **Then**
      Update $CurrTT := CurrTT \cup \{p\}$ and $c^{CurrTT} := c^{CurrTT} + c_p$
    Otherwise
      Update $c^{CurrTT} := c^{CurrTT} + \pi_l^{canc}$
    End If
    Update $LeftLines := LeftLines \setminus \{l\}$
  End While
  If $c^{CurrTT} < c^{MacroTT}$ **Then**
    Compute (see Appendix B) the robust cost cr^{CurrTT} of timetable $CurrTT$
    If $c^{CurrTT} + cr^{CurrTT} < c^{MacroTT}$ **Then**
      Set $MacroTT := CurrTT$ and $c^{MacroTT} := c^{CurrTT} + cr^{CurrTT}$
    End If
  End If
End For

Algorithm *MacroHeu* is a randomized multi-start greedy heuristic that computes a

number of heuristic solutions for formulation (6.5)-(6.14) and returns the ”best” one found to the microscopic module. A step-by-step description of *MacroHeu* is provided in Algorithm 10.

The initialization phase of *MacroHeu* consists of setting $MacroTT$ equal to the empty set and its cost $c^{MacroTT}$ equal to 0, where $MacroTT$ is the subset of macroscopically conflict-free paths (at most one for each line) that is returned as output of the procedure. Then an iterative procedure starts running $\overline{iter}$ times, where $\overline{iter}$ is a parameter that is set equal to 1000 in the computational experience reported in Section 6.7. At each of these $\overline{iter}$ iteration, $CurrTT$ represents the incumbent solution (i.e., it is a subset of conflict-free paths), c^{CurrTT} is the cost of timetable $CurrTT$, and $LeftLines$ is the subset of lines that have to be processed in the current iteration $iter$. Each iteration $iter$ consists of two main steps: first, an attempt to find a macroscopically feasible timetable is made by iteratively selecting a line and running a procedure that finds the least-cost feasible path compatible with the paths of the set $CurrTT$ (where the cost of such a path does not consider only the cost of the path itself but also the cost deriving from penalties related to connections); second, timetable $CurrTT$ is assessed in terms of robustness, compared with the best timetable $MacroTT$ found so far, and the best timetable $MacroTT$ is possibly updated accordingly. Therefore, at each iteration, algorithm *MacroHeu* makes an attempt to finding a macroscopic feasible timetable by iteratively fixing variables x_p to 1 (and variables ξ_l if no feasible path is found for line $l \in L$), while increasing the cost of timetable $CurrTT$ as little as possible.

Two crucial sub-routines are used in each iteration of *MacroHeu*. The first sub-routine finds the least-cost (with respect to the current timetable $CurrTT$) path for a line $l \in L$. The second sub-routine assesses the robust cost cr^{CurrTT} of timetable $CurrTT$. These two sub-routines are described in Appendix A and B, respectively.

6.6 Constraint updating

When running our micro-macro framework, we sequentially adapt process times at micro level and return to the macroscopic model to re-compute the MacroTT again. In fact, process times of the micro model represent and define constraints for the macroscopic timetabling model. If MacroTT has conflicts at the microscopic level, then macro constraints on train process times are recomputed by the microscopic model according to two main processes, namely: 1) constraints tightening and 2) constraints relaxation. In this section we describe these procedures in detail.

6.6.1 Constraints tightening

Once the macroscopic timetable is obtained, the microscopic model runs the conflict detection algorithm that tests the feasibility of the produced timetable, i.e., the potential existence of conflicts. If a conflict is observed, then the time separation between two trains has to be increased to satisfy safety constraints. By doing so, the degree of freedom for scheduling a train path (i.e., its search space) gets smaller, since constraints

on minimum headway times between two train paths are introduced. We call the procedure of increasing the constraints on minimum headway times *constraint tightening*.

For each corridor $\varphi \in \Phi$, the existing conflict for a train pair $i, j \in T$, $c_{ij\varphi} > 0$, is resolved by updating the headway time as:

$$h_{ijs} \leftarrow h_{ijs} + c_{ij\varphi},$$

where s is a timetable point on corridor φ. Note that the process for detecting a conflict on a corridor and identifying the corresponding station headway to solve that conflict, is not trivial. Once a track conflict is detected, the algorithm identifies the location of the closest timetable point s to the conflict, and computes the minimum headway time which solves that conflict. In addition, the minimum time separation to avoid conflicts between two trains depends on whether the trains run in the same or in the opposite direction, so we determine the corresponding interaction between conflicting trains that may be arrival-arrival or departure-departure for trains running in the same direction, as well as arrival-departure or departure-arrival for trains running in opposite directions. Consequently, the minimum headway time h_{ijs} can be correctly updated.

We here give an example for updating the minimum headway time between two trains is updated if they were detected to be conflicting. Trains t_1 and t_2 run in the same direction along corridor φ, from timetable point s_1 to timetable point s_2 and have a track conflict with overlap time $c_{t_1t_2\varphi}$. If the conflict is geographically closer to location s_1 then we update the headway between the departures of our trains at location s_1, $h^{dd}_{t_1t_2s_1}$. Alternatively, if the conflict is closer to location s_2 then we update the headway between the arrivals of our trains at s_2, $h^{aa}_{t_1t_2s_2}$. The minimum time headway h_{ijs} is so increased by $c_{ij\varphi}$ and thereby tightening the relative constraint in the macroscopic model.

6.6.2 Constraints relaxation

A timetable point (e.g., station, junction) in a railway network is considered to be a potential bottleneck if the corresponding infrastructure occupation rate exceeds the thresholds recommended by the UIC on capacity usage. In this case, enlarging the time separation between trains at that timetable point is necessary to reduce the infrastructure occupation rate and introduce additional buffer times beneficial for the mitigation of delay propagation. We propose two types of constraint relaxations to reduce unacceptable occupation rate: 1) train homogenization, and 2) journey time extension.

First, it is commonly known that homogenised traffic consumes the least infrastructure capacity (Hansen & Pachl, 2014). Driven by this logic, we can guide trains to have more unified behaviour, i.e., more similar macroscopic running times. This can be obtained twofold: first, by allowing fast (intercity) trains to run slower, and second, by increasing the operational speed of slow (regional) trains through the bottleneck area.

This relaxation procedure is implemented as follows. Parameter μ is the threshold recommended by the UIC for a good usage of infrastructure capacity, while Δ is the percentage increment in running time supplement in the timetable (e.g., +0.5%). If the infrastructure occupation rate for corridor φ between two stations is $\mu(\varphi) > \nu$, then: *i)* the maximum running time supplement for fast trains t_F on corridor φ will be increased by Δ_1 as in (6.15) and *ii)* the nominal running time supplement for slow trains t_S on the same corridor will be decreased by Δ_2 as in (6.16).

$$\lambda_{\varphi}^{max}(T_I) \leftarrow (1+\Delta_1)\,\lambda_{\varphi}^{max}(t_F) \tag{6.15}$$

$$\lambda_{\varphi}^{nom}(T_R) \leftarrow (1-\Delta_2)\,\lambda_{\varphi}^{min}(t_S). \tag{6.16}$$

As a further measure to reduce infrastructure occupation, we also increase the maximum allowed journey time of fast trains. Note that this relaxation goes in line with (6.15) as a necessary correction. For example, let us assume a scheduled fast train that has assigned the maximum journey time and runs on a corridor where the total capacity is higher than $\mu(\varphi)$. If we increase the maximum running time for one train only locally λ_{φ}^{max}, while maximum journey time $\overline{J}_t$ stays unchanged, we might not experience the benefit of the λ_{φ}^{max} relaxation because $\overline{J}_t$ will be still the bounding constraint. Therefore, it is important to relax both $\lambda_{\varphi}^{max}$and $\overline{J}_t$.

6.7 Computational experiments

This section considers the computed timetable for the Dutch case study, including the computational results, the achieved values for the performance measures and plots illustrating the timetable and its performance measures. We have tested our iterative approach for timetable planning on a railway corridor in the Netherlands. The models and the framework are developed in Matlab and C++. Tests are made by using a dual core Intel E7 with 2.6 GHz processor and 8 GB RAM. The microscopic and macroscopic modules used only one processor core. As input to construct the timetable, we considered the Dutch train service specification for the year 2012.

6.7.1 Case study

We have designed the timetable for a relevant part of the Dutch network which includes the corridors between the stations of Utrecht (Ut) and Eindhoven (Ehv), 's Hertogenbosch (Ht) and Tilburg (Tb), and 's Hertogenbosch and Nijmegen (Nm). Figure 2a illustrates the geographical representation of the test case. The network in the microscopic model consists of 1500 homogenous behavioural sections, 950 block sections and 28 microscopic timetable points (e.g., stations, stops, junctions, bridges) of which five are IC stations: Ut, Ht, Ehv, Nm and Tb. Most of the corridors are double-track.

The timetable on this network is periodic with an hourly pattern composed of eight Intercity (IC) and twelve regional train lines all with two services per hour. So, a

Table 6.1: Timetable design norms

Maximum journey time extension	20%
Minimum running time supplement	5%
Maximum running time supplement	30%
Dwell time at stops	35 s
Dwell time at macro points	1-2 min
Minimum transfer times	1-3 min

total of 40 train runs per hour are scheduled over the whole area. The corresponding line plan is given in Figure 6.2b, where each colour represents a single train line. In particular, two IC lines serve the corridor Ehv-Ht-Ut, one serves Tb-Ht-Nm and one Ehv-Tb. Regional train lines operate on the corridors: Tiel (Tl)-Geldermalsen (Gdm)-Ut, Tb-Ht, Ht-Gdm-Ut, Ht-Nm, Ehv-Ht and Ehv-Tb. This line plan results in 16 operating trains between Ut and Gdm, 4 between Gdm and Tl, 12 between Gdm and Ht, 8 between Ht and Nm, as well as Ht and Tb, and 20 between Ht and Ehv.

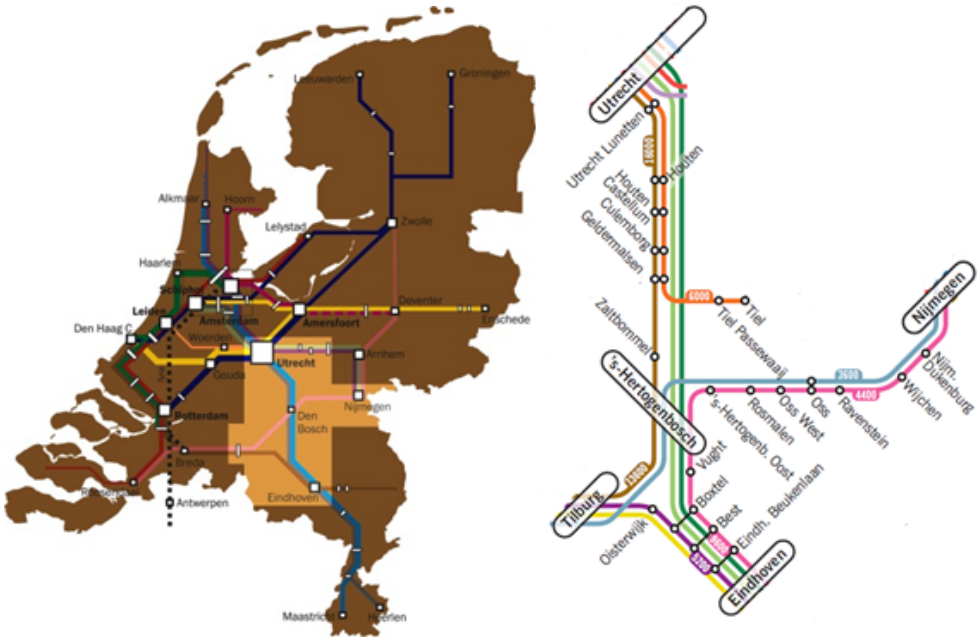

Figure 6.2: a) Dutch railway network with highlighted case study area and b) train line plan

Table 6.1 gives the timetable design norms that have been used as an input for the timetabling process. These values are provided by the railway planners.

The developed framework computes the necessary process times and sets up the macroscopic network by applying Algorithm 8. The macroscopic model is built by aggregating the microscopic infrastructure model. Specifically, this process has aggregated the 28 microscopic points into 15 timetable points in the macroscopic model with 15 corresponding arcs between them. The macroscopic infrastructure model is illustrated in Figure 3. Most of the lines are double-track unless the given number suggests differently. For example, the corridor between Boxtel (Btl) and Ehv consists of a four-track

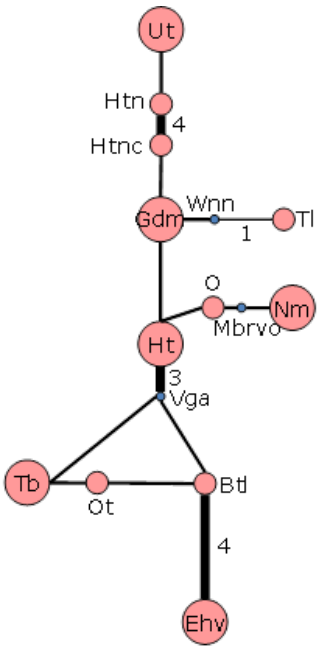

Figure 6.3: Macroscopic network

line. Note that not all microscopic timetable points are considered in the macroscopic model as explained in Section 6.3.1. In total, the function *AggregateProcessTimes* produces 76 macroscopic running times for any pair of consecutive timetable points for all of the 20 train lines, and 2027 minimum headway time computations. The computation time depends on the size of the network and the number of train runs. For the considered case study, the execution of Algorithm 8 took under 30 seconds to generate the macroscopic network model and compute the aggregated process times. The average computation time per iteration is about 40 seconds for the microscopic model, and about 80 seconds for the macroscopic one (with $\overline{iter}$ =1000 macroscopic iterations). Thus, the average time per micro-macro iteration was in total 120 seconds.

Figure 6.4 shows the computational results of the micro-macro iterations for obtaining a conflict-free, robust and stable timetable. After nine iterations the algorithm converged to a feasible solution which is both microscopically conflict-free and stable and macroscopically optimized. During the iterations a decreasing trend can be observed for the number of conflicts (blue solid line) and the total overlap time of conflicting blocking times (green dashed line), with some iterations leading to an increased number of conflicts and overlap time when the timetable structure (train orders) changes significantly from one iteration to the next in face of new minimum headway times provided to resolve the conflicts. The total computation time, from the microscopic input computation to the produced conflict-free, stable and robust timetable was about 20 min.

Figure 6.5 shows the time-distance diagram, while the associated blocking time diagram is given in Figure 6.6. The vertical axis shows time in minutes downwards. The horizontal axis shows distance with the station positions indicated. The blue lines are

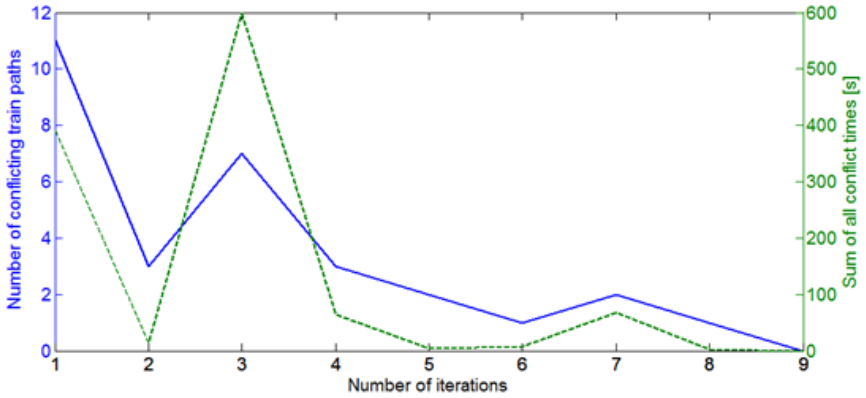

Figure 6.4: Evolution of the micro-macro interactions

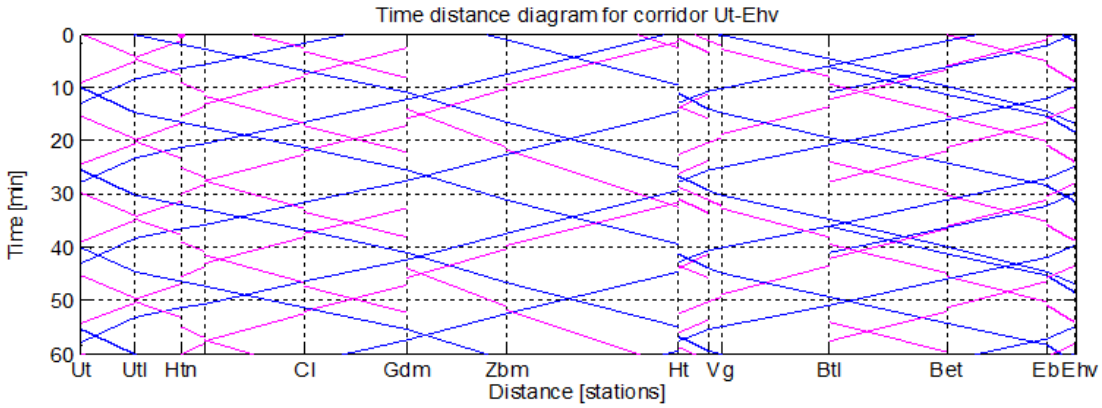

Figure 6.5: Time-distance diagram corridor Utrecht – Eindhoven

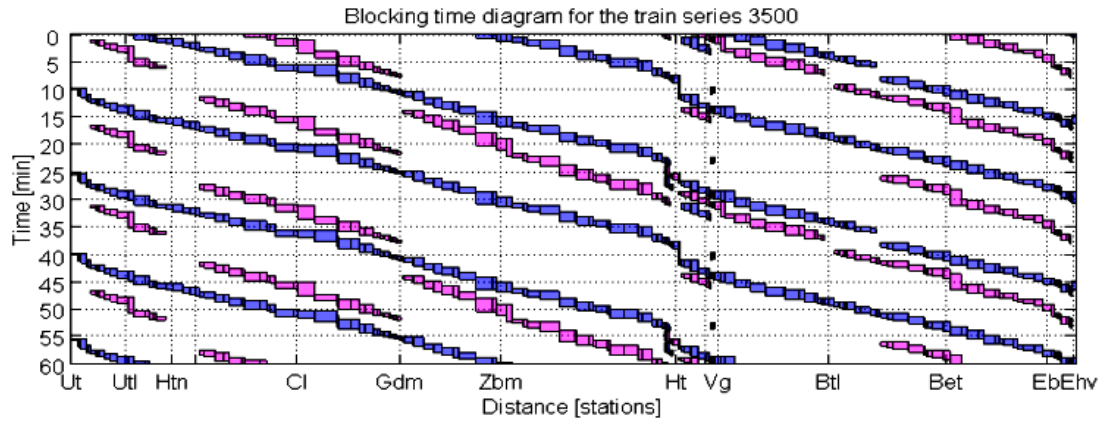

Figure 6.6: Blocking time diagram corridor Utrecht – Eindhoven

IC trains, while magenta lines are local trains. Note that Figure 6.6 considers only the infrastructure route used by the intercity train line 3500, running form Ut to Ehv. This means that blocking times of other trains will be represented in the picture only if these trains traverse block sections which are on the route of train line 3500.

Table 6.2: Infrastructure occupation at main corridors

Corridor	Time [s]	%
Ut-Ht	1968	54.7
Ht-Ut	1924	53.4
Ehv-Ht	1320	36.7
Ht-Ehv	1338	37.2

The optimized timetable shows periodic passenger trains with regular 15 minute services of both IC and local trains where two similar train lines follow the same route. Hence, effectively 15 minute train services are realised instead of two separate 30 minute train lines. On the main corridor Utrecht-Eindhoven, the ICs overtake local trains at Geldermalsen (Gdm), but this overtaking does not take place for trains in the opposite direction.

Table 6.7.1 shows the infrastructure occupation on the main corridors. All the infrastructure occupation rates are below the threshold recommended by the UIC of 75% defined for mixed traffic corridors at peak hours. The two heaviest used corridors are Utrecht – 's -Hertogenbosch in both directions, with a maximum infrastructure occupation rates of 54.7% for one direction and 53.4% for the opposite direction. The other corridors have an infrastructure occupation rate below 41%.

6.7.2 Additional computational analyses

To demonstrate the applicability of the micro-macro timetabling model, we utilized additional computations. We randomly generated line plans for a different number of train lines in the train service specification that ranged from 16 to 25. Table 6.3 shows the computational results for all given scenarios. For every scenario it is reported the number of train lines in the line plan, the number of macroscopic running arcs and headway arcs, the number of micro-macro iterations, the infrastructure occupation rate of the most used corridor, the average time supplements allocated in the corresponding timetable, the time for microscopic to macroscopic conversions in the initialization, and the total computation times. The scenario previously analysed in Section 6.7.2 is reported as scenario basic while the other scenarios were randomly generated from the set of lines of the basic scenario by varying the chosen lines.

In general, the number of micro-macro iterations grows when the number of requested train lines increases. The timetabling model needed at least three iterations to find a solution (*sc5*) when the number of train lines was 16. On the other hand, at least 10 iterations were needed to obtain the conflict-free solutions for scenarios with more than 20 lines. This could be expected as the larger the number of train lines, the less is the freedom to schedule trains on the given infrastructure. Consequently, the computation times ranged between 353 and 1885 s.

The number of macroscopic running arcs also depends on the number of train lines in the train service specification and we observed this number varied in the range from 56

Table 6.3: Computational results for all scenarios

Scen.	# lines	# trains	# macro running arcs	# headway arcs	# iter	Max infra occupation rate (%)	Average time supplements (%)	Init. (s)	CPU time (s)
basic	20	40	76	1712	9	53.44	8.80	29	1054
sc1	16	32	62	1304	6	42.46	8.79	23	611
sc2	16	32	56	966	3	46.89	8.78	23	353
sc3	17	34	66	1423	7	57.94	8.89	24	719
sc4	17	34	60	1049	5	46.89	8.88	24	539
sc5	18	36	70	1564	8	53.44	9.28	26	814
sc6	18	36	64	1154	4	51.33	9.27	26	502
sc7	19	38	70	1414	4	52.56	8.87	28	514
sc8	21	42	82	2030	11	59.06	8.93	30	1165
sc9	21	42	79	1846	5	57.94	8.84	30	640
sc10	22	44	88	2386	9	63.67	8.70	32	1089
sc11	22	44	82	2000	10	57.94	8.59	32	1207
sc12	22	44	84	2060	10	57.94	9.16	34	1304
sc13	23	46	91	2540	14	60.67	9.04	34	1795
sc14	23	46	85	2095	13	57.94	8.53	34	1694
sc15	24	48	90	2264	14	57.94	8.94	36	1809
sc16	24	48	90	2384	11	57.94	9.29	36	1456
sc17	25	50	96	2758	15	64.39	9.35	37	1885

to 96. A similar situation is observed for the number of headways which ranged from 996 to 2758 and the initialization time which varied between 23 and 37 s. The average computation time for calculating minimum and operational running times for a single train line was 1 s and 5 s, respectively. The average time supplement for all scenarios varied between 8.78 and 9.27%.

We also observed that the infrastructure occupation rate does not explicitly grow when increasing the number of train lines. Thus, changing the number of trains does not necessarily mean a change in infrastructure occupation. For example, this was observed between scenarios *basic* and *sc5*, where the maximum infrastructure occupation rate remained the same although the number of train lines was different, 20 and 18, respectively. Thus, we may say that the infrastructure occupation rate does not depend only on the number of train lines, but also on the characteristics of the line plan such as the type of line service (heterogeneity), the origin and destination stations, the line routes and the scheduled connections.

Figure 6.7 gives the time-distance diagram for the scenario with the biggest line plan *sc17* which included 25 train lines.

.

6.8 Conclusions

This paper presented an integrated automatic timetable planning framework that produces timetables that are microscopically feasible, stable and robust. The developed approach incorporated the strengths and advantages of microscopic and macroscopic

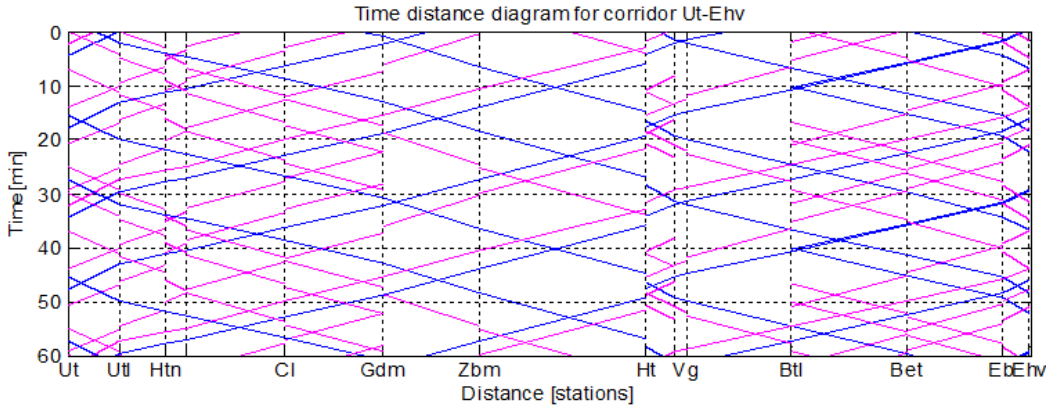

Figure 6.7: Time-distance diagram corridor Utrecht – Eindhoven for scenario *sc17*

algorithms to provide overall efficient and satisfactory solution. Network transformation algorithms were introduced to automatically convert data from the microscopic level to macroscopic one and vice versa. The macroscopic model is used for computing a robust network timetable which is afterwards converted and thoroughly analysed at the microscopic level. The analysis includes conflict detection and capacity assessment. If track conflicts are detected and/or capacity norms are violated, necessary adjustments to train process times were undertaken by applying a procedure of constraints tightening and relaxation. This iterative micro-macro process automatically terminates once the timetable is also microscopically feasible and stable.

A practical application to an area of the Dutch railway network showed the ability of this framework in ensuring the feasibility of the macroscopic timetables at the level of track detection sections. A high quality timetable was produced in 3 to 15 iterations, depending on the given number of train lines plan, while the computing times were between 353 and 1885 seconds. In addition, the UIC norms on infrastructure occupation rates were satisfied, so that for all the scenarios we obtain a maximum occupation rate below 65% [1].

The proposed framework and integrated models are suitable for developing both periodic and non-periodic timetables. Practitioners and timetable designers can use this framework for timetable design and for the evaluation of existing timetables. Future research will be addressed to generate and evaluate timetables that also include scheduling of short-term freight train paths. Also a specific study will be dedicated to define performance measures that evaluate the resilience of the timetable, i.e., the ability of restoring scheduled operations when real-time rescheduling is applied during perturbed traffic. The use of the proposed micro-macro approach is indicated for practitioners as a tool for generating timetables that are operationally feasible and robust to daily perturbations of the scheduled train operations.

[1] Only a subset of all freight trains was considered in the analysis.

Appendix A. Finding the least-cost path for a given line and timetable

The sub-routine for finding the least-cost path for a given line and a given partial timetable works as follows. The insertion of a line into the incumbent timetable corresponds to fixing a variable x into the model (6.5)-(6.14) and is performed by running an exact dynamic programming recursion, a simplified version of the one used in Cacchiani et al. (2010), that identifies a feasible min-cost path that is compatible with the paths assigned to the previously processed trains/lines, where the cost takes into account not only the cost of the paths but also the cost for connections. Such a dynamic programming recursion computes functions $f(t,\sigma,e)$ with three state-variables, where t is the instant of time at which the event e (be it an arrival, a departure, or a pass at a given station along the route of the train) takes places, and σ represents the total stretch of the path (computed as the running and dwell time excess over the nominal ones).

This sub-routine receives as input the index l of a line of the set $LeftLines$ that must be scheduled, and a timetable $CurrTT$ that contains at most a path for each of the lines of the set $L \backslash LeftLines$. The output of this sub-routine is either a path of the set P_l that minimizes the increase in the cost of timetable $CurrTT$ while maintaining its feasibility, or the indication that all of the paths of the set P_l are incompatible with the paths of the set $CurrTT$, which means that at least one of the constraints (6.10) is violated if such a path of line l is added to $CurrTT$.

The sub-routine is a dynamic programming recursion that computes functions $f(t,\sigma,e)$, for each instant of time $t = 0,\ldots,\bar{t}$ (where $\bar{t}$ is the length of the planning horizon over which the timetabling computation is performed), each possible total stretch $\sigma = 0,\ldots,\bar{s}_l$ with respect to nominal running and dwell times of line l, and each event e (being it a departure and/or an arrival to each of the timetable points belonging to the route of line l). Function $f(t,\sigma,e)$ represents the cost of the min-cost partial path from the departure from station s_1 of S_l up to event e among all partial paths scheduling event e at time t with a total stretch equal to σ.

Functions $f(t,\sigma,e)$ are computed according to the following recursive propagation rules:

- If event $e = dep_i$ corresponds to the departure from timetabling point $s_i \in S_l$, $i = 1,\ldots,|S_l|-1$, then

$$\begin{aligned} &f(t,\sigma,dep_i) = \\ &\min_{t'=\bar{t}+t-\overline{w}_{l,s_i},\ldots,\bar{t}+t-w_{l,s_i}} \Big\{ f\left(t'\%\bar{t}, \max\left\{\sigma+t'-\bar{t}-t+w_{l,s_i},0\right\}, arr_i\right) \\ &+\pi_l^{dwell}\left(\max\left\{t'-\bar{t}-t+w_{l,s_i},0\right\}\right)\Big\} \end{aligned} \tag{6.17}$$

for each instant of time $t = 0,\ldots,\bar{t}$ and each possible stretch $\sigma = 0,\ldots,\bar{s}_l$;

- If event $e = arr_i$ corresponds to the arrival at timetabling point $s_i \in S_l$, $i = 2, \ldots, |S_l|$, then

$$f(t,\sigma,arr_i) = \min_{t'=\bar{t}+t-\bar{r}_{l,(s_{i-1},s_i)},\ldots,\bar{t}+t-r_{l,(s_{i-1},s_i)}} \Big\{ f\left(t' \% \bar{t}, \max\{\sigma + t' - \bar{t} - t + r_{l,(s_{i-1},s_i)}, 0\}, dep_{i-1}\right) + \pi_l^{runn}\left(\max\{t' - \bar{t} - t + r_{l,(s_{i-1},s_i)}, 0\}\right) \Big\} \tag{6.18}$$

for each instant of time $t = 0, \ldots, \bar{t}$ and each possible stretch $\sigma = 0, \ldots, \bar{s}_l$.

Here, the term % represents the modulo operation, that is the remainder of the Euclidean division, $t' \% \bar{t} = t' - \bar{t} \cdot \left\lfloor t'/\bar{t} \right\rfloor$.

To explain the previous recursive propagation rules, consider the following examples:

- Given $\bar{t} = 100$, $t = 50$, and $\sigma = 10$, and considering the departure from station i (i.e., considering event dep_i), we want to compute functions $f(50, 10, dep_i)$, knowing that the nominal dwell time w_{l,s_i} is 5 and the maximum dwell time $\overline{w}_{l,s_i}$ is 10. All functions $f(50, 10, dep_i)$ recursively originate from functions $f(t', \sigma', arr_i)$, where t' is in between 40 $(= \left(\bar{t} + t - \overline{w}_{l,s_i}\right) \% \bar{t})$ and 45 $(= \left(\bar{t} + t - w_{l,s_i}\right) \% \bar{t})$ and $\sigma' = 10$ when $t' = 45$, $\sigma' = 9$ when $t' = 44$, …, $\sigma' = 5$ when $t' = 40$, where $\sigma' = \sigma + t' - \bar{t} - t + w_{l,s_i}$.
- Given $\bar{t} = 100$, $t = 50$, and $\sigma = 10$, and considering the arrival at station i (i.e., considering event arr_i), we want to compute functions $f(50, 10, arr_i)$, knowing that the nominal running time $r_{l,(s_{i-1},s_i)}$ from station s_{i-1} to station s_i is 5 and the maximum running time $\bar{r}_{l,(s_{i-1},s_i)}$ from station s_{i-1} to station s_i is 10. All functions $f(50, 10, arr_i)$ recursively originate from functions $f\left(t', \sigma', dep_{i-1}\right)$, where t' is in between 40 $(= \left(\bar{t} + t - \bar{r}_{l,(s_{i-1},s_i)}\right) \% \bar{t})$ and 45 $(= \left(\bar{t} + t - r_{l,(s_{i-1},s_i)}\right) \% \bar{t})$ and $\sigma' = 10$ when $t' = 45$, $\sigma' = 9$ when $t' = 44$, …, $\sigma = 5$ when $t' = 40$, where $\sigma' = \sigma + t' - \bar{t} - t + r_{l,(s_{i-1},s_i)}$.

In order to compute functions $f(t, \sigma, e)$ recursively, the following initialization is required:

$$f\left(t, 0, dep_{l,s_1}\right) = 0$$

for each instant of time $t = 0, \ldots, \bar{t}$, which means that the cost for departing from the first station of line l with no stretch is 0 no matter the departing time, and

$$f\left(t, \sigma, dep_{l,s_1}\right) = \infty$$

for each instant of time $t = 0, \ldots, \bar{t}$ and each possible stretch $\sigma = 1, \ldots, \bar{s}_l$.

Moreover, whenever event e cannot take place at time t because this corresponding path when added to $CurrTT$ would violate some constraints (e.g., headway times, capacity constraints, connections involving lines already scheduled in $CurrTT$, etc.), we set $f(t,s,e) = \infty$ for each possible stretch $\sigma = 0,\ldots,\bar{s}_l$.

It is clear that, by computing functions $f(t,\sigma,e)$ as described above, the only penalties that are taken into account are the ones for exceeding nominal dwell times (i.e., π_l^{dwell}) and for exceeding nominal running times (i.e., π_l^{runn}). So far, penalties related to connections (i.e., π_q^{conn} and $\overline{\pi}_q^{conn}$) have not been considered. Nonetheless, it is easy to observe that, given the subset of paths of the set $CurrTT$, the penalties that must be paid if a path for line l is selected depend uniquely on the time each of the events of line l take place. This means that before computing functions $f(t,\sigma,e)$, the penalty $\pi^{conn}(t,e)$ for adding to $CurrTT$ a path of line l where event e takes place at time t can be computed.

Penalties $\pi^{conn}(t,e)$, for each instant of time $t = 0,\ldots,\bar{t}$ and each event e are computed as follows:

- If event $e = dep_i$ corresponds to the departure from timetabling point $s_i \in S_l$, $i = 1,\ldots,|S_l|-1$, then

$$\pi^{conn}(t,dep_i) = \begin{cases} \infty & \text{if } \exists l' \in L\backslash LeftLines \ \wedge \ \exists q = \left(l',l,s_i\right) \in Q : \\ & \pi^A_{p(l')s_i} < t - \underline{u}_q \vee \pi^A_{p\left(l'\right)s_i} > t - \overline{u}_q \\ \sum_{q=\left(l',l,s_i\right)\in Q} \pi_q^{conn}\left(t - \pi^A_{p(l')s_i}\right) & \text{otherwise,} \end{cases}$$

  where $\pi^A_{p(l')s_i}$ indicates the arrival time of line l' at timetable point s_i in path $p(l')$, which is the path assigned to line l' in $CurrTT$. This means that, whenever there exists a line l' that has already been scheduled (i.e., $l' \in L\backslash LeftLines$) and a connection between the arrival of line l' and the departure of line l from timetable point s_i that cannot be met if the departure of line l from s_i is scheduled at time t, then any path of line l scheduling such a departure at time t is infeasible; otherwise (if such a line and such a connection do not exist), then the penalty for scheduling the departure of line l from timetable point s_i at time t is given by the sum of the differences between t and the arrival time at s_i of all lines l' for which a connection $\left(l',l,s_i\right) \in Q$ exists;

- If event $e = arr_i$ corresponds to the departure from timetabling point $s_i \in S_l$, $i = 2,\ldots,|S_l|$, then

$$\pi^{conn}(t, arr_i) = \begin{cases} \infty & \text{if } \exists l' \in L \backslash LeftLines \wedge \exists q = (l, l', s_i) \in Q : \\ & \pi^D_{p(l')s_i} < t + \underline{u}_q \vee \pi^D_{p(l')s_i} > t + \overline{u}_q \\ \sum_{q=(l,l',s_i)\in Q} \pi^{conn}_q \left(\pi^D_{p(l')s_i} - t \right) & \text{otherwise,} \end{cases}$$

where $\pi^D_{p(l')s_i}$ indicates the departure time of line l' at timetable point s_i in path $p(l')$, which is the path assigned to line l' in $CurrTT$. This means that, whenever there exists a line l' that has already been scheduled (i.e., $l' \in L \backslash LeftLines$) and a connection between the departure of line l' and the arrival of line l at timetable point s_i that cannot be met if the arrival of line l at s_i is scheduled at time t, then any path of line l scheduling such an arrival at time t is infeasible; otherwise (if such a line and such a connection do not exist), then the penalty for scheduling the arrival of line l at timetable point s_i at time t is given by the sum of the differences between the departure time from s_i of all lines l' for which a connection $(l, l', s_i) \in Q$ exists and t.

Therefore, in order to keep into account penalties related to connections, the recursive equations (6.15) and (6.16) have to be modified as follows:

$$f(t, \sigma, dep_i) = \min_{t' = \bar{t}+t-\overline{w}_{l,s_i}, \ldots, \bar{t}+t-w_{l,s_i}} \left\{ f\left(t' \% \bar{t}, \max\left\{ \sigma + t' - \bar{t} - t + w_{l,s_i}, 0 \right\}, arr_i \right) + \pi^{dwell}_l \left(\max\left\{ t' - \bar{t} - t + w_{l,s_i}, 0 \right\} \right) \right\} + \pi^{conn}(t, dep_i)$$

and

$$f(t, \sigma, arr_i) = \min_{t' = \bar{t}+t-\overline{r}_{l,(s_{i-1},s_i)}, \ldots, \bar{t}+t-r_{l,(s_{i-1},s_i)}} \left\{ f\left(t' \% \bar{t}, \max\left\{ s + t' - \bar{t} - t + r_{l,(s_{i-1},s_i)}, 0 \right\}, dep_{i-1} \right) + \pi^{runn}_l \left(\max\left\{ t' - \bar{t} - t + r_{l,(s_{i-1},s_i)}, 0 \right\} \right) \right\} + \pi^{conn}(t, arr_i).$$

Then, the path for line l that implies the minimum increase when added to the set $CurrTT$, corresponds to the one generating the function $f\left(t^*, \sigma^*, arr_{|S_l|}\right)$, where

$$(t^*, \sigma^*) = \underset{t=0,\ldots,\bar{t};\ \sigma=1,\ldots,\bar{s}_l}{\arg\min} \left\{ f\left(t, \sigma, arr_{|S_l|}\right) \right\}.$$

Appendix B. Robustness assessment of a given timetable

This procedure is aimed at assessing the robustness of a given timetable. The input of the procedure is timetable $CurrTT$ generated in a given iteration of MacroHeu, and the output is its robust cost cr^{CurrTT}, which is defined in the following. The main idea about assessing the robustness of the timetable is to generate a number of different scenarios, each one characterized by a random delay for each train, and run on each of these scenarios a local search procedure that tries to eliminate conflicts by retiming trains. The robust cost cr^{CurrTT} is then determined by the weighted sum of the unresolved conflicts plus the time to absorb the delays on each of the scenarios considered. A step-by-step description of the robustness assessment procedure is provided in Algorithm 11.

Algorithm 11 Step-by-step description of the robustness assessment procedure

Input: macroscopic timetable $CurrTT$ of cost $c^{C}urrTT$
Output: robust cost cr^{CurrTT} of timetable $CurrTT$
0. Initialize cr^{CurrTT}:=0
For $nscenario = 1, \cdots, \overline{nscenario}$ **do**
1. Initialize $TTDelay$ by starting from $CurrTT$
2. Generate a random delay for each train of timetable $TTDelay$
3. Run a local search procedure that retimes trains of $TTDelay$ to resolve conflicts
4. Update cr^{CurrTT}
End For

Step 1 consists of an initialization phase. Timetable $TTDelay$ is the timetable that will be used in the following steps. It is the same as CurrTT except that the trains of each line are replicated. This means that, for example, if in $CurrTT$ there is a line with periodicity 30 minutes that is scheduled to depart from station s at 10:00, then in $TTDelay$ such a line corresponds to n trains (where n is a parameter, e.g., $n = 4$); the first one scheduled to depart from s at 10:00, the second at 10:30, the third at 11:00, and the fourth at 11:30. In other words, from the periodic timetable $CurrTT$ the corresponding non-periodic timetable $TTDelay$ is generated.

Step 2 consists of the generation of the delays for each train, in particular, delays are generated for the first train of each line of the non-periodic timetable $TTDelay$ only. The delays are generated according to a standard normal distribution and normalized over the periodicity of the timetable. In particular, for each train t a random value *rand* is generated and an event e (i.e., a departure or an arrival at one of the timetable points traversed) is randomly selected. The time of event e in timetable $TTDelay$ is then postponed by $rand * \overline{\theta}/3$ units of time, where $\overline{\theta}$ is the maximum delay considered. Function *rand* generates a number randomly based on a truncated Normal distribution $N(0,1)$ that allows only positive values that are not greater than three.

Having generated a delay for the first train of each line, the resulting timetable $TTDelay$ may no longer be feasible from a macroscopic point of view: there may be headway

times, capacity constraints, and/or connection times violated. In order to recover such a feasibility, all conflicts are iteratively processed, one at a time, by considering first the conflict occurring the last in time (i.e., conflicts are resolved starting from the latest ones in the time horizon), and trains causing the conflict are retimed to resolve the conflict. In particular, the retiming operation consists of simply increasing the dwell times and/or the running times of the trains involved in the conflict; no reordering nor rerouting of the trains are allowed. The procedure ends as soon as all conflicts are resolved or the resolution of the conflicts would imply that more conflicts would be generated.

Step 4 consists of updating the robust cost of timetable $CurrTT$. Let γ be the number of unresolved conflicts in $TTDelay$ (i.e., the number of unsatisfied connections, headway times, capacity constraints, etc.). Moreover, let $\widehat{\pi}^D_{p(t)s_i}$ be the departure time from timetable point $s_i \in S_l$, $i = 1,\ldots,|S_l|-1$, in the path $p(t)$ selected for any of the trains t of line $l \in L$, and let $\widehat{\pi}^A_{p(t)s_i}$ be the arrival time at timetable point $s_i \in S_l$, $i = 2,\ldots,|S_l|$, in the path $p(t)$ selected for any of the trains t of line $l \in L$ in timetable $TTDelay$ obtained after Step 3. The robust cost cr^{CurrTT} is updated by adding $\pi^{uns}\gamma + \pi^{delay}\left(\sum_{l\in L}\sum_{t\in T(l)}\sum_{s\in S_l}\left(\widehat{\pi}^D_{p(t)s_i} - \pi^D_{p(t)s_i} + \widehat{\pi}^A_{p(t)s_i} - \pi^A_{p(t)s_i}\right)\right)$, where π^{uns} and π^{delay} are parameters representing the penalties for each unsolved conflict and each unit of delay, respectively.

Chapter 7

A simulation-based optimization approach for the calibration of dynamic train speed profiles

---

This chapter has been published as:

Bešinović, N., Quaglietta, E., & Goverde, R. M. P., (2013). A simulation-based optimization approach for the calibration of dynamic train speed profiles. *Journal of Rail Transport Planning & Management*, 3(4), 126-136.

7.1 Introduction

Recent demand growth for passenger and freight transportation in railway systems has raised the need for practitioners to increase the level of network capacity while keeping a high standard of service availability and quality. To achieve this aim railway traffic needs to be scheduled according to robust timetables that guarantee higher levels of capacity usage also in presence of stochastic disturbances. On the other hand, suitable control measures (e.g., train retiming, reordering and/or rerouting) must be applied in real-time by dispatchers to provide rescheduling plans that mitigate the effects of observed conflicts on network performances. Both robust timetabling and real-time management of railway traffic aim at supplying conflict-free train paths computed on the basis of off-line and on-line predictions of traffic behaviour. In the first step, train trajectories must be computed taking into account microscopic details of the infrastructure (e.g., lengths, gradients, curvatures of rail tracks, speed limits), signalling system (e.g., positions of signals, block section lengths, braking behaviour imposed by the automatic train protection), train composition (e.g., number of wagons, rolling stock characteristics), and current traffic information when the prediction is performed

on-line. Then, based on the estimated train trajectory a conflict-free schedule is constructed by solving a mathematical problem (e.g., optimization, heuristics), or by relying on rule-of-thumbs or experience of the operator (i.e., a planner in timetabling and a dispatcher within real-time operations). The effectiveness of these schedules depends on the reliability of the estimated train trajectories and the precise identification of potential track conflicts. Inaccurate forecasts can lead to wrong detection of possible conflicts and to traffic schedules that are ineffective or even infeasible when put into operation. In this context, accurate traffic prediction models must be used to confidently describe the real evolution of train behaviour. To this purpose a proper calibration phase is needed to estimate input parameters against train data (e.g., position, speed) collected from the field, so that the model can reproduce the real train trajectories as much as possible.

This paper presents an approach to derive the most probable speed profiles of train runs from observed track occupation/release data. The train behaviour is modelled according to the Newton dynamic motion equations which are numerically integrated over distance employing the Runge-Kutta method (Butcher, 2008). A simulation-based optimization approach is adopted to calibrate input parameters of the equations describing the tractive effort, the motion resistances, the braking effort, and the cruising phase. These parameters are fine-tuned for different classes of train composition (defined by the number of wagons, the type of traction unit, and the length of the train) by minimizing the gap between observed and simulated running times, using a genetic algorithm. Additionally, since the train composition is not known with certainty beforehand, a model for train length estimation is developed. For each composition the calibration experiment is performed over a significant set of observed train runs. This enabled estimating the probability distributions the different input parameters for each class of train compositions. This aspect gives also insight in different driving behaviours adopted during real operations. The proposed approach is applied to train runs operating along the corridor Rotterdam-Delft in the Netherlands. Results illustrate the effectiveness of this method in calibrating parameters of the Newton's dynamic equations versus track occupation/release data collected at the level of track sections.

With this paper the authors provide the following main contributions:

- A novel simulation-based method to calibrate the parameters of the train dynamic motion equations against observed track occupation data. This approach allows the derivation of train speed profiles from the real distance-time trajectory collected at discrete points from track-free detection sections.
- A procedure to assess the length of trains from time-distance data collected by track-free detection sections
- A statistical assessment of parameters relative to both physical-mechanical characteristics of trains (e.g., coefficients of resistance and traction equations) and the behaviour of train drivers (e.g., compliance to the max speed limit on the track, braking rate applied).

- A practical application to a real test case which proves the applicability of the proposed approach and the usefulness that results can have for both practitioners (e.g., more reliable predictions of train trajectories) and academics (e.g., distribution of parameters suitable for robust timetabling design).

Section 2 gives a literature review on the different approaches proposed to model train running times and calibrate model parameters. In section 3, the methodology proposed in this paper is described. Section 4 illustrates the case study considered for the application and provides the corresponding results. Conclusions and final comments are given in section 5.

7.2 Literature review

In the literature, several approaches are presented for estimating train running times taking into account microscopic features of both trains and the infrastructure (including the signalling system). In particular, models can be mainly divided in the ones using kinematic motion equations and others adopting a dynamic representation of the movement, basically by means of Newton's motion formula (Hansen & Pachl, 2008).

T. Albrecht, Gassel, Knijff, and van Luipen (2010); T. Albrecht, Goverde, Weeda, and Van Luipen (2006) described train motion based on the kinematic equations and calibrate their parameters (speed and acceleration) versus track occupation data collected by means of train describer systems (Daamen, Goverde, & Hansen, 2009; Goverde & Meng, 2011). T. Albrecht, Gassel, Binder, and van Luipen (2010) use calibrated kinematic models to understand the influence of the Dutch signalling and ATP system on train speed profile and energy consumption. The disadvantage of these models is that they calibrate only the parameters of the kinematic motion equations which are trajectory-dependent and cannot be used anymore when considering a different train run even if the rolling stock is the same.

Medeossi, Longo, and de Fabris (2011) use a dynamic equation for each phase of the train motion (i.e., acceleration, cruising, coasting and braking) and fine-tunes the respective performance parameters against GPS data collected on-board of the trains. A probability distribution is then estimated for these parameters to characterize stochastic variations of running times.

Hertel and Steckel (1992) proposed a model that computes running times based on theoretical stochastic distributions of train parameters (e.g., resistance coefficient, braking rate) instead of using typical deterministic parameters as commonly considered in practice. The parameter distributions adopted in this work are however not derived from any realised train run.

Kecman and Goverde (2012) adopt a method suitable for real-time predictions, that represents train trajectories by means of a weighted graph that evolves dynamically each time that new information is gathered from the field; weights of the arcs are train

running and dwell times and minimum headway times measured by means of detailed track occupation/release data from train describer records collected at the level of track sections (e.g., axle counters, track circuits).

During real operations stochastic variations to individual train runs are observed due to changes in the rolling stock condition, rail deterioration, as well as variations in the train driver behaviour and weather circumstances. These unpredictable variations induce an alteration of train characteristics such as the deceleration and the acceleration rates as well as motion resistances (e.g. due to gradient, air viscosity, rail curvatures) and consequently, a change in train trajectories Kecman and Goverde (2013). According to this, approximated parameters estimated by manufacturer or train operators should not be taken for granted (Radosavljević, 2006), but need to be computed for each train composition and railway corridor separately.

This work helps filling the gap between practice and theory under the following perspectives:

- So far, research approaches proposed in literature were mainly focussed on calibrating parameters of the kinematic train motion equations (T. Albrecht, Gassel, Binder, & van Luipen, 2010; T. Albrecht, Gassel, Knijff, & van Luipen, 2010; T. Albrecht et al., 2006) or only performance factors of the dynamic train motion equations[12]. This work instead has the objective to calibrate all the parameters of the dynamic train motion equation and not only performance factors as in Medeossi et al. (2011). The fact that we consider and calibrate all the parameters of the dynamic equation, gives to our model a higher flexibility than (Medeossi et al., 2011) since it can accurately describe every kind of observed trajectory. This means that it can reproduce every type of observed driving behaviour.

- Compared to the previous work by Medeossi et al. (2011), the main advantage of our approach is that we manage to accurately describe observed train trajectories on the basis of track occupation data and not GPS. Currently, only in rare cases it is possible to use GPS data, given that the most part of railway networks in Europe are not equipped with these systems. Most part of the railway networks are equipped with track-free detection systems that detects the occupation/release of a certain track from a train. This means that the model proposed in this paper can be used for all those networks having track-free detection systems since we use exactly these data to calibrate train parameters. Moreover, these data are automatically collected which provides a big amount of data for detailed analyses.

- The presented methodology provides probability distributions of train parameters fitted on data gathered from the real field, which can be used for more reliable robust timetabling (where train running times are generated from random distributions) or as more realistic input for the model of Hertel and Steckel (1992) to calculate train running times.

To the best of the authors' knowledge no efforts have been addressed in literature to the estimation of parameters relative to tractive effort and motion resistances based on actual track occupation data.

7.3 Methodology

7.3.1 A simulation-based framework to calibrate dynamic equations of train motion

To provide a reliable prediction model able to accurately reproduce real train trajectories it is necessary to calibrate model parameters against real data collected from the field. In this paper the calibration process is performed by developing a simulation-based framework that integrates a genetic algorithm with a microscopic running time model based on dynamic motion equations as given by Newton.

This framework has been developed in Matlab and consists of several components (Figure 7.1). The entire framework is based on data relative to the infrastructure (e.g., track length, gradient, speed limits, signal and station positions), the rolling stock features (e.g., train length) and the track occupation/release collected from the field. A pre-processing phase is necessary to convert the different input data into a suitable format and combine them in order to derive information needed to initialize the calibration model. In particular these data are combined to identify the exact route (i.e. the sequence of track sections, switches, signals, and stations crossed by the train during its run) and the train length (which is related to the composition) of each observed train run. Train length has been used to group the observed train runs in different classes of train compositions. Parameters of the running time model are estimated separately for each class.

Also, track occupation/release data are processed to derive discrete space-time trajectory data for each observed run that are used to evaluate the objective function at each iteration of the optimization algorithm. The calibration experiment is performed only against distance-time data relative to unhindered trains, thus train runs that are not disturbed by the presence of other trains on the network. This assumption consents to understand how the value of train parameters varies over different runs only due to the behaviour of the train driver and not to the interactions with other trains.

The proposed algorithm developed for the optimization problem is customised genetic algorithm which is implemented in Matlab. Output of the framework consists of: calibrated parameters of the dynamic equation for each train (i.e., braking rate, parameters of the tractive effort equation, coefficients of the resistance equation, speed adopted in cruising phases) and the corresponding train trajectories (i.e., distance-time diagrams, speed-time and speed-distance diagrams).

This framework has been applied to calibrate a significant set of train runs for each class of train compositions. By doing this, it has been possible to estimate the proba-

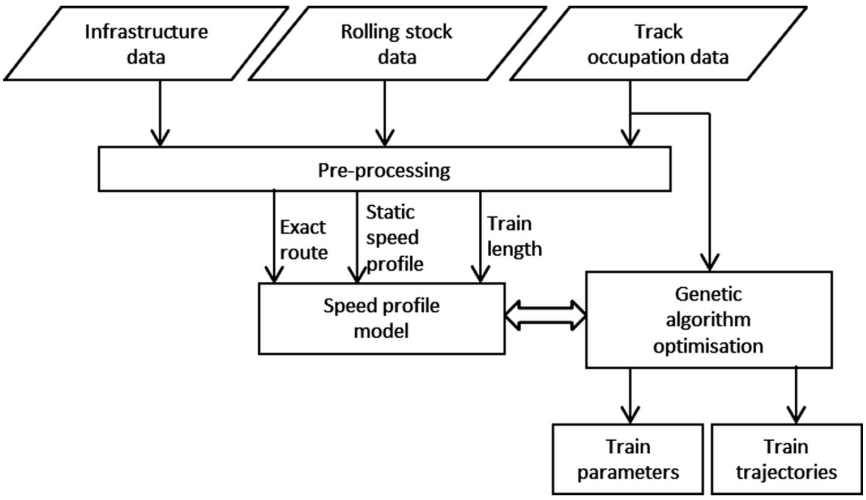

Figure 7.1: Functional scheme of the simulation-based optimization framework

bility distribution relative to the input parameters of the running time model for each train class.

7.3.2 Input data

Input data to the proposed framework are relative to the infrastructure characteristics, the rolling stock features and observed track occupation/release data collected at the level of track sections for a significant set of train runs. In this section a detailed description of each of these data is provided as follows:

- Infrastructure data contains detailed information about microscopic characteristics of railway network. These data describe lengths of track sections, curvature radii, gradients, static speed limits, positions of stations, signals and switches as well as the braking behaviour and the supervised speed codes by the Automatic Train Protection system (in this case represented by the Dutch ATB). All this information is derived from infrastructure maps and digital InfraAtlas data provided by the Dutch infrastructure manager (IM) ProRail.

- Rolling stock data specify the features regarding rail vehicles such as train compositions (number of wagons, type of traction unit), mass, parameters of the tractive effort-speed curve as well as coefficients of the resistance equations. This data have been supplied by the main Dutch railway Undertaking (RU) Netherlands Railways (NS).

- Track occupation/release data are gathered from field measurements that return the event time that a given train has occupied or released a certain section on the network. This information has been collected by means of the train describer system in the Netherlands called TROTS (ProRail, 2008). This system logs generated train number messages and incoming infrastructure messages (from

signals, switches, track sections) to provide a list of events in a chronological order. The advantage of the TROTS system is that it is able to record train number steps at the level of track sections. Measured occupation/release times are rounded down to the full second and are affected by an error (i.e., delay) of release of track circuits, that has been defined for safety reasons. This measurement inaccuracy has a big influence on very short sections when short occupation times are observed. These data are pre-processed by using the data mining tool developed by Kecman and Goverde [9].

7.3.3 Data pre-processing

The main function of the pre-processing phase is to: i) convert the different input data into a format that is usable by the developed framework, ii) combine these data in order to derive additional information which are needed to initialize and apply the calibration model. Specifically, the latter process is addressed to provide for each observed train run: the exact route and the train length.

The route of a train is defined as the sequence of infrastructure elements (i.e. track sections, switches, signals, station platforms) traversed during its run. To determine the route relative to a certain observed train run, it is necessary to combine track occupation/release data corresponding to that run together with the infrastructure data (InfraAtlas maps). The track occupation/release data is a chronological ordered list of the IDs (identification number) relative to the infrastructure elements crossed by the train during a certain run. By coupling this list of IDs with the infrastructure data it is possible to identify the route followed by that run in terms of length of track sections, gradients, static speed limits, curvature radii, the switches used, the signals approached, and the platforms at which it stopped.

In the Netherlands, different rolling stock is used in service and these variations may be observed even at a single train line. Despite the existing rolling stock plans for each day of operation, the realised rolling stock tends to differ due to both real-time situations that cannot be predicted in advance (i.e., train delays and track obstructions) and the actual rolling stock availability (e.g. due to breakdowns or unplanned maintenance). Moreover, both planned or realised rolling stock may be unavailable to the infrastructure manager. Hence, there is a need for detecting train compositions that have been actually used during real operations. This detection can be performed by estimating train lengths by means of track occupation/release data. To explain this procedure it is possible to refer to the example illustrated in Figure 7.2 where two track sections s_i are represented together with their respective section joints x_i and x_{i+1}. The average speed of train run j when traversing track section s_i can be calculated as:

$$\bar{v}_{i,j} = \frac{x_{i+1} - x_i}{t^{occupy}(s_{i+1}) - t^{occupy}(s_i)} \quad [m/s] \tag{7.1}$$

where $t^{occupy}(s_{i+1})$ and $t^{occupy}(s_i)$ represent the time in which the head of the train enters track section s_i and s_{i+1}, respectively. Also, $t^{release}(s_i)$ is the time instant in

which the tail of the train releases track section s_i. As said in Section 3.2, the release time $t^{release}(s_i)$is affected by an accuracy error δ that is the time delay between the perceived and the actual instants in which a train releases section s_i (i.e., $t^{release} \pm d$). Due to this delay we can only estimate an interval I_j for the length of train j. The width of interval I_j (expressed in m) is easily assessed as:

$$I_j = [\bar{v}_{i,j} \cdot ((t^{release}(s_i) - d) - t^{occupy}(s_{i+1})), \bar{v}_{i,j} \cdot ((t^{release}(s_i) + d) - t^{occupy}(s_{i+1}))]. \tag{7.2}$$

Consequently, we assume that the expected length l_j of train j coincides with the median of interval I_j.

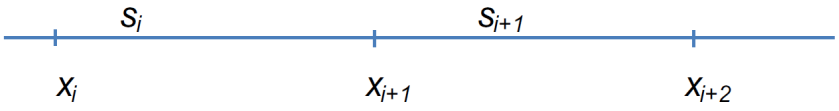

Figure 7.2: Track sections and respective joints

Assume that from the analysis of rolling stock data we observe different possible train compositions c_1, c_2, c_3, c_4 with associated lengths L_1, L_2, L_3, L_4, respectively. L_i of each possible composition c_i, where $i = (1,2,3,4)$. The composition assigned to j train will be therefore the one whose length L_i is the closest to the estimated length l_j, i.e., the one which minimizes the difference $|L_i - l_j|$. When the estimated interval I_j is too wide, it may happen that multiple composition lengths are covered. Also, it may happen that no train lengths are feasible. In these cases it is not possible to assign a specific composition to train run j.

7.3.4 Microscopic speed profile model based on dynamic motion equations

The developed running time model is based on Newton's dynamic motion equations, where the train is modelled as a mass point. This assumption is widely accepted and used in practice (Hansen & Pachl, 2008), since practical applications have shown satisfactory results. The train length is not neglected in the model since the trajectory of the tail of the train is obtained from the one of the head shifted back for the train length. Referring to the Newton's motion law, the force $f_s(v)$ (surplus force) that is used to accelerate a train is produced by the difference between the tractive effort $f_t(v)$, and the resistance forces $r(v)$. The tractive effort is generated by the traction unit and applied at the wheel's rim. The resistance forces are obtained as the sum of the resistances due to air viscosity and line characteristics (e.g., gradient and curves). This relation can be formally expressed as:

$$f_t(v) - r(v) = f_s(v) = m \cdot dv/dt. \tag{7.3}$$

The tractive effort is assumed a piecewise function of the train speed v consisting of a

linear and a hyperbolic part (Hansen & Pachl, 2008):

$$f_t(v) = \begin{cases} c_0 + c_1 v, & v \leq v_{overheat}, \\ c_2/v, & v > v_{overheat}. \end{cases} \tag{7.4}$$

The linear part of the function $(c_0 + c_1 v)$ is valid for values of the speed lower than the so called overheat speed limit $v_{overheat}$, while a hyperbolic characteristic is denoted for higher speeds and presents a limitation due to adhesion and tractive power. The resistance forces $r(v)$ acting against the train movement are modelled as a second-order polynomial of speed, expressing resistances on a flat and straight line $(r_0 + r_1 v + r_2 v^2)$, and constant resistances due to the topology of tracks, i.e., gradient (f_G) and curve alignment (f_C), respectively [7]:

$$r(v) = r_0 + r_1 \cdot v + r_2 \cdot v^2 + f_C + f_G. \tag{7.5}$$

The coefficients r_0, r_1 and r_2 depend on several variables such as type of the rolling stock, train composition and number and type of train axles. The constant and linear term with coefficients r_0 and r_1 represent the mechanical resistance of the rolling stock, while the quadratic term models the aerodynamic resistance. In this model extra resistances relative to the presence of tunnels are not considered. Figure 7.3 shows a typical trend for the tractive effort and the train resistances as described by 7.4 and 7.5, respectively.

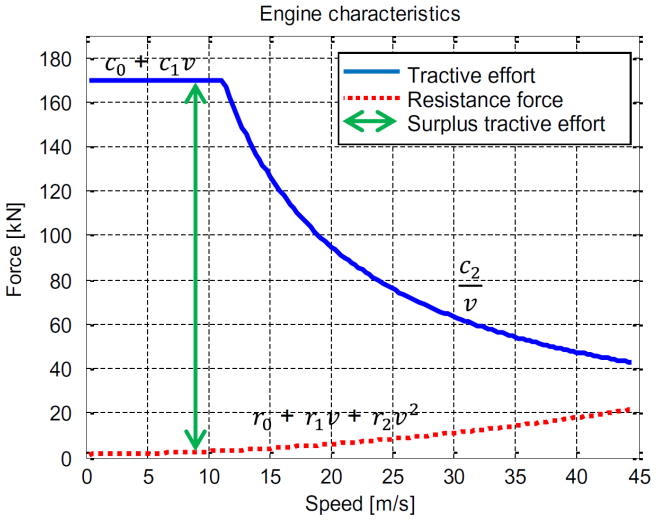

Figure 7.3: Train characteristics

It should be noted that the mass of the train is implicitly included within the coefficients of resistances and tractive effort equations. Indeed, these are specific coefficients since they are expressed per mass unit. Also, weather conditions such as wind speed, are embodied as part of the parameters. For example, if weather conditions are bad (i.e., wind against train movement direction) this will result in higher resistance parameters.

In order to estimate train trajectories, it is necessary to solve equation 7.3 for each

phase of the motion, that is to say: i) acceleration, ii) cruising and iii) braking. The analysed corridor Rotterdam-Delft is one of the densest corridors in the Dutch network, thus it is assumed that just small running time supplements had been allocated. Therefore, the coasting phase is not included in the presented running time model. The following characteristics are considered for each phase:

- In the acceleration phase the driver is supposed to accelerate the train by using the tractive effort described by 7.4 until the train reaches the maximum speed allowed for a given track, or the desired value of cruising speed.
- In the cruising phase the train moves with a constant speed. For a certain track this speed can be the static maximum speed, or a certain lower cruising speed deployed by the train driver. Therefore, the rate between a static speed limit and the cruising speed actually operated is represented by $\theta_{cruising}$. This cruising performance can vary from track to track and depends on the driver behaviour.
- In the braking phase train speed is reduced to accomplish speed restrictions imposed by the track (e.g. static speed limits, switches, stops at stations) or by the signalling system (e.g. red or yellow aspects). Experimental results presented in Medeossi et al. (2011) show that during service two different braking rates are used by trains when 1) slowing down to respect static or dynamic (e.g. given by the signalling system) speed limits and 2) coming to a standstill because of stopping in a station. This assumption has been made in the present model, whereby two braking rates are used for the former (b_{limit}) and the latter case (b_{stop}), respectively. Specifically, due to the specific allocation of track circuits it has been not possible to collect time data suitable for the determination of b_{stop}. That is why a default value of 0.66 m/s^2 has been used for this parameter as provided by NS.

A partial train trajectory is determined for each phase by computing the speed v assumed by the train at a certain distance s, and afterwards calculating the time t corresponding to obtained speed and distance. Particularly, a dynamic train speed profile is modelled as a function of speed depending on distance:

$$\frac{dv}{ds} = \frac{f_t(v) - r(v)}{v}, \tag{7.6}$$

where dv/ds is the derivative of speed with respect to distance. \f speed dependent on distance aerodynamic resistance. The corresponding running time is expressed as:

$$\frac{dt}{ds} = \frac{1}{v}, \tag{7.7}$$

where dt/ds is the derivative of time to distance. The given equations (7.6)-(7.7) are autonomous first-order ordinary differential equations for which several numerical solution methods have been tested in terms of speed and accuracy. As a result, the method

given by Dormand-Prince (Butcher, 2008) is adopted which is a particular application of the more general Runge-Kutta approach.

7.3.5 Formulation of the calibration model: a simulation-based optimization problem

The calibration process is formulated as an optimization problem that aims to minimize the error between simulated and real passage running times. As explained earlier, actual running times are derived by pre-processing TROTS data.

The decision variables (i.e., the parameters that need to be calibrated) of the problem are listed in Table 7.1.

Table 7.1: Decision variables

Parameter	Description
c_0	maximum starting tractive effort due to overheating limit [N/kg]
c_1	linear parameter of tractive effort equation [Ns/m/kg]
c_2	hyperbolic parameter of tractive effort function [Nm/s/kg]
r_0	constant resistance coefficient [N/kg]
r_1	linear resistance coefficient [Ns/m/kg]
r_2	quadratic resistance coefficient [Ns^2/m^2/kg]
b_{limit}	braking to speed limit characteristic [m/s^2]
$\theta_{cruising}$	cruising performance [%]

The optimization problem can now be formulated as:

$$\min \sum_{i \in N} |t_i^{observed} - t_i^{simulated}| \tag{7.8}$$

Subject to

$$\frac{dv}{ds} = \frac{f_t(v) - r(v)}{v}, \tag{7.9}$$

$$\frac{dt}{ds} = \frac{1}{v}, \tag{7.10}$$

$$c_0 \in [c_0^{lb},\ c_0^{ub}], \tag{7.11}$$

$$c_1 \in [c_1^{lb},\ c_1^{ub}], \tag{7.12}$$

$$c_2 \in [c_2^{lb},\ c_2^{ub}], \tag{7.13}$$

$$r_0 \in [r_0^{lb},\ r_0^{ub}], \tag{7.14}$$

$$r_1 \in [r_1^{lb},\ r_1^{ub}], \tag{7.15}$$

$$r_2 \in [r_2^{lb},\ r_2^{ub}], \tag{7.16}$$

$$b_{limit} \in [b_{limit}^{lb},\ b_{limit}^{ub}], \tag{7.17}$$

$$\theta_{cruising} \in \left[\theta_{cruising}^{lb}, \theta_{cruising}^{ub}\right], \tag{7.18}$$

$$v(0) = v_0 = 0, \qquad v(N) = v_{end} = 0, \tag{7.19}$$

where the objective function (7.8) is represented by the absolute error between the simulated and observed passage running times for all the N measurements provided by the TROTS data. It is clear that the evaluation of the objective function requires a preliminary computation of the speed profile and the running time. This means that a numerical integration of the speed and the running time as represented by equations (7.9) and (7.10) must be performed at each iteration of the optimization algorithm.

These parameters are relative to the tractive effort equation (c_0, c_1 and c_2), the resistance equation (r_0, r_1 and r_2), the braking rate used to slow down (b_{limit}) and the cruising performance adopted by train driver during cruising phases ($\theta_{cruising}$), respectively. Equations (7.11) - (7.18) define the optimization constraints for each of these variables imposing the lower (lb) and upper bounds (ub) of their domains. Finally, the equation (7.19) gives the initial and final speed conditions representing that a train starts the run from a standstill and stops at the end of route.

Therefore, a solution to the optimization problem is represented by the vector:

$$\beta = (c_0, c_1, c_2, r_0, r_1, r_2, b_{limit}, \theta_{cruising}), \tag{7.20}$$

which contains a set of values for the decision variables.

7.3.6 The optimization metaheuristics: a genetic algorithm

A genetic algorithm (GA) is developed to solve the optimization problem. GA is a well-known robust and adaptive method largely used in the scientific field to solve search and optimization problems. The algorithm works with a population of individuals, each representing a possible solution, in this case a set of train parameters β. Each individual produces a different value of the objective function. The population evolves towards better solutions (i.e. lower values of the objective function) by means of randomized processes of selection, crossover, and mutation (see [13] for more information on the topic). The GA used in this research has been developed in Matlab and customized to improve its performances according to the specific problem applied. Moreover, its execution has been parallelized by allocating different functions of the algorithm to different threads. This strongly reduces computing times of the optimization when adopting multi-core computers.

7.4 Case study: the Rotterdam-Delft corridor

The framework proposed in this research has been applied to calibrate a significant set of trains running along the corridor Rotterdam-Delft, which is one of the most densely operated lines in the Netherlands. The line has a length of 14.3 km with a double track layout. The Dutch signalling system NS'54 with ATB automatic train protection is implemented over the whole corridor. A detailed explanation of this system can be found in T. Albrecht, Gassel, Binder, and van Luipen (2010). Both regional and

Table 7.2: Input data of rolling stock

Train composition	Length [m]	v_{max} [km/h]
VIRM4	108	160
VIRM6	162	160
VIRM10	270	160
ICRm (Locomotive 1700 + 10 cars)	282	160

Intercity (IC) trains operate on this line, but for the sake of simplicity the analysis performed in this research is only demonstrated to the latter type of trains.

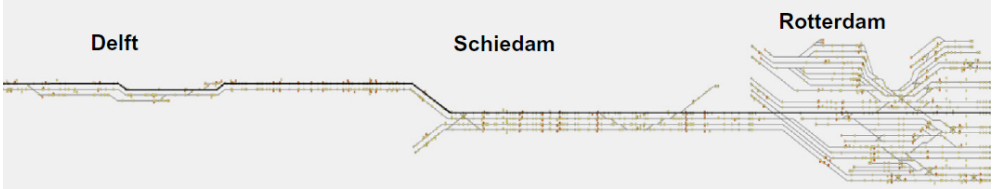

Figure 7.4: Schematic layout of the corridor Rotterdam – Delft

In particular, the intercity train line IC1900 is analysed. According to the timetable, the rolling stock used in service is reported in Table 7.2. Four different classes of train compositions have been observed: the electrical multiple units VIRM with four, six and ten units as well as the locomotive hauled trains ICRm with ten cars. All these trains use the same route and therefore the same platform tracks, in- and outbound interlocked routes, and block sections, with a slight difference in terms of the stop locations in stations.

The calibration of the running time model is performed for observed TROTS data collected over a 28-days period of operation in April 2010. In total 42 track sections have been considered. This means that the parameters of each train run have been calibrated versus 42 time-distance observations.

All the calibration experiments are carried out on an AMD Athlon 3300 GHz processor with six cores and 4GB of RAM. The integration of a single train trajectory takes about 0.02 seconds, while the computing time needed to complete a single calibration experiment is always lower than one minute.

7.4.1 Analysis of parameters and model performance

A preliminary sensitivity analysis has been performed to understand which input parameters is the more influential for the running time model. This has been carried out by evaluating variation of the running time model output by changing the value of one parameter while keeping fixed the other ones. This procedure has been repeated for all input elements. Only the parameters that produced a significant variation of the running time have been selected for the calibration, since the model is more sensitive to these ones. In particular, the linear parameter of the resistance equation, r_1, does not

Table 7.3: Model performance output

Parameter	Default value	Average value	Standard Deviation	
			Value	%
$c_0[10^{-4}\text{N/kg}]$	5.62	5.33	0.05	0.96
c_2[Nm/s/kg]	6.21	6.29	0.00	0.00
$r_0[10^{-2}\text{N/kg}]$	1.53	1.60	0.07	4.24
r_2 $[10^{-5}\text{Ns}^2/\text{m}^2/\text{kg}]$	4.08	3.55	0.10	2.79
b_{limit} [m/s^2]	0.66	0.24	0.00	0.00
$\theta_{cruising}$[%]	100	101	0.00	0.00
Objective function [s]	135.76	79.00	0.34	0.42
Running time [s]	595.8	572.64	0.18	0.03

have significant importance, given that this parameter produces a variation of running times less than 0.1%. Lukaszevicz (2001) came to the same conclusion for passenger trains. A small relevance is also identified for the linear parameter of the tractive force equation, c_1. Hence, fixed values have been assumed for these two parameters and the calibration process has been reduced to the following factors:

$$\beta = (c_0, c_2, r_0, r_2, b_{limit}, v_{cruising}). \tag{7.21}$$

As a result, the value of c_1 has been set to zero, while r_1 is fixed to the default value used by the RU (given by the rolling stock input data) and dependent on the train length.

A robustness analysis has been carried out to evaluate the robustness and performance of the optimization algorithm. In particular, 30 calibration experiments have been executed for a fixed realised train trajectory. This gives insight in whether the algorithm is able to return consistent results for the same calibration problem (with the same observed data). If the value returned for each parameter is not the same over the different experiments then the algorithm is not robust enough and/or the optimization problem is not well-defined. Results obtained from this test are reported in Table 3 which shows the average and the standard deviation of the values determined for each parameter over the 30 calibration experiments. It can be seen that parameters c_2, b_{limit} and $\theta_{cruising}$ converge to the same value for all the calibration experiments. Relatively low values of the standard deviation are observed for r_0 and r_2, 4.24% and 2.79%, respectively. However, variations of these two parameters produce just slight changes in the objective function value and the total running time of 0.4% and 0.03%, respectively. This outcome confirms the robustness of the algorithm used and the validity of the formulated optimization problem.

The parameters are compared with the corresponding default values provided by the RU (second column of Table 7.3). As can be seen the calibrated values vary around the default ones for all the parameters but the braking rate b_{limit}. The latter is due to the fact that during the observed train run a train driver adopted a braking rate that was on average lower than the one assumed by the operating company. Therefore, such aspect

highlights the ability of the proposed model to estimate also the driving behaviour of the train driver.

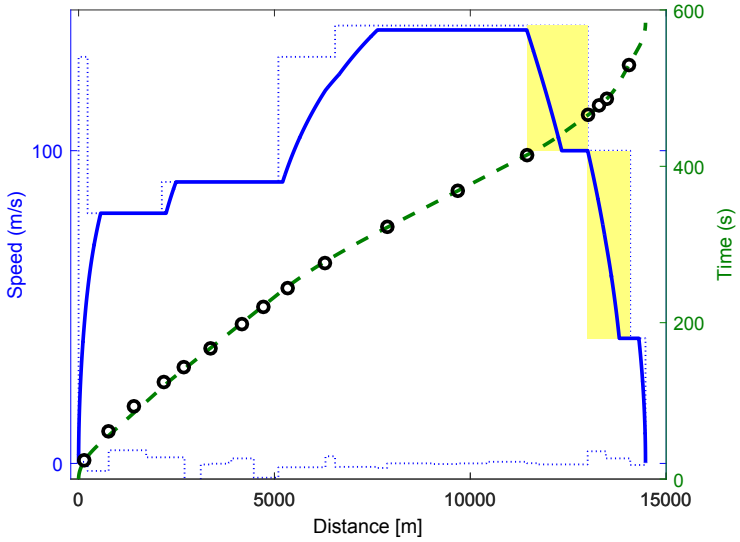

Figure 7.5: Estimated speed profile and time-distance diagram for a single train run

Figure 7.5 illustrates the output of a single calibration experiment: the calibrated distance-speed diagram (solid line) and the corresponding time-distance trajectory (dashed line). The circles depict measured time-distance as given by TROTS. The effectiveness of the calibration performed is immediately visible since the simulated time-distance trajectory practically overlaps observed data. This means a very high accuracy of the model. The gradient profile of the track is reported with the blue line at the bottom while the static speed limit is depicted with the dashed blue line. Yellow blocks represent the approach indication corresponding to those block sections in which trains has to start braking because of a restricted aspect imposed by the NS'54/ATB system.

7.4.2 Train length estimation

The train lengths are estimated by means of the process explained in Section 3.3. Figure 7.6 shows the obtained intervals for the train lengths of the observed trains. Horizontal lines show the width of this interval for each train run, while vertical lines indicate the four lengths associated to each of the four compositions considered. A different line style has been used to represent the estimated length. A dash-dotted line is adopted for the class VIRM4, dotted for VIRM6, dashed for VIRM10, and solid for ICRm, while solid grey is employed to represent cases in which it was not possible to have a correct estimation of train length (i.e. when no composition length falls inside the interval).

As can be seen these intervals of train lengths have different ranges. This depends on the value of the measurement error d that affects release times of track circuits.

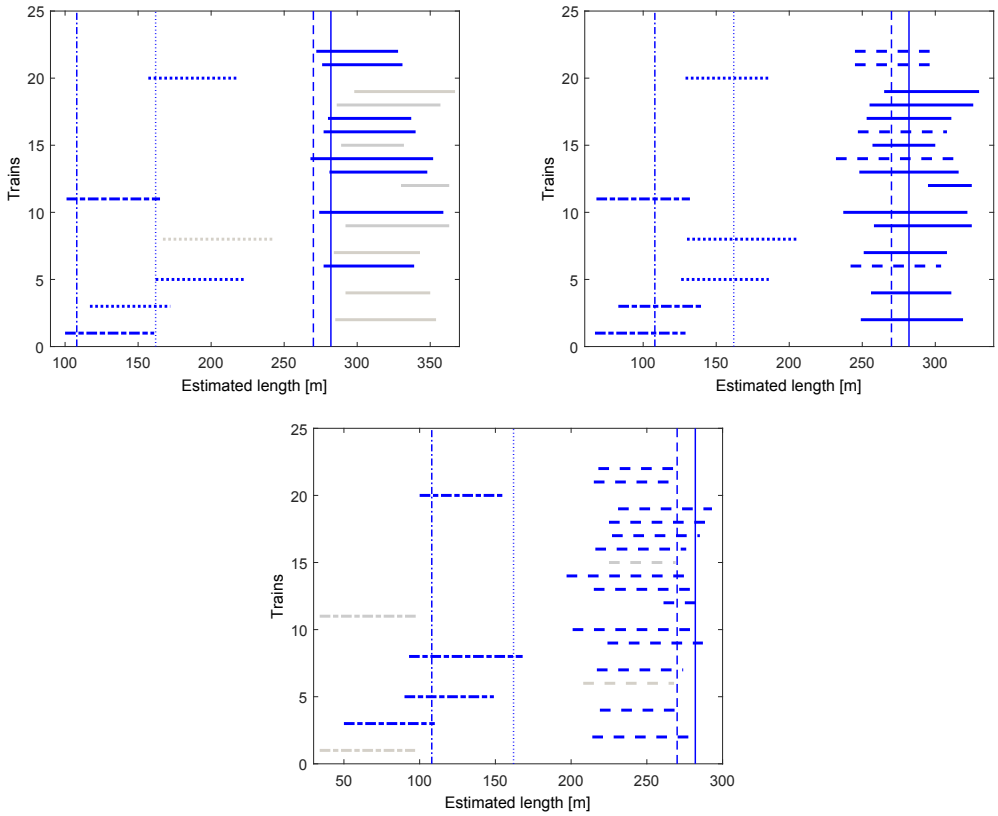

Figure 7.6: Estimation of trains lengths for: a) actual measured release times, b) measured release times delayed by one second and c) measured release times delay by two seconds

Specifically d is the time delay between the time in which the train actually releases a track circuit and the time that instead this track circuit perceives this release. It is easy to understand that as effect of this inaccuracy the average train speed and consequently the intervals of train lengths are estimated with some tolerance as can be observed in Figure 7.6. In order to understand how the value of error d affects the accuracy of the estimated intervals of train lengths, we assessed these interval against three different values of d, namely 0, 1 and 2 seconds. Such an assessment exposed that by assuming a value of $d = 1$ s (7.6b) it was possible to estimate the lengths of the largest amount of observed trains. Therefore we adopted this as the value of the time delay during the whole analysis.

7.4.3 Calibration results

Calibration of parameters is undertaken for the four classes of train compositions. For each class 70 train runs have been examined. Parameters of the running time model have been calibrated for each of the train runs. This means that 70 sets of calibrated parameters β is provided for each train composition. This consents to estimate variations of a certain parameter over different train runs for a given composition. A probability distribution has been assessed for each parameter by applying the method of the max-

imum likelihood estimation (MLE). The goodness-of-fit of the distribution to the data has been tested using the Kolmogorov-Smirnov (KS) test. The probability distributions that we obtained for a certain parameter are identified as the distributions having the best fit with the observed data, i.e. the lowest value of the KS-statistic. The KS-statistic assumes indeed low values for a good distribution fit, while high values for a bad fit. On the other hand, the P-value ranges between 0 and 1, and it is close to 1 for a good fit while close to 0 for bad fit. Figures 7.7-7.9 shows the results obtained for the train class VIRM4. In particular, for each train parameter the figures report the corresponding probability distribution, the related distribution parameters, and the corresponding values of the KS-statistic and the P-value. It should be noted that similar distributions are obtained for other compositions that for brevity are not explicitly reported.

Figure 7.7a gives the distribution of the constant parameter of the tractive effort. It shows that this parameter fits best to a Weibull distribution. As expected, not all the observed train runs use the maximum tractive effort while accelerating from a standstill. Nevertheless, some runs exceeded the theoretical maximum tractive force given by the RU. Figure 7.7b shows that parameter c_2 fits best to the generalised extreme value (GEV) distribution. It is observed that in a certain number of train runs c_2 was higher than the experimental maximum.

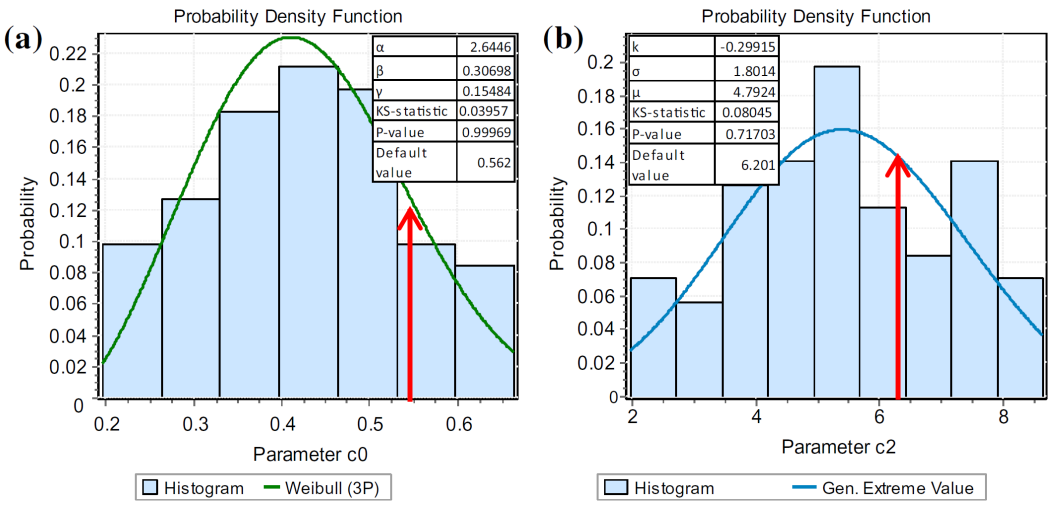

Figure 7.7: Distributions of tractive effort parameters

The constant parameter of the resistance equation r_0 (Figure 7.8a) shows a fit to a uniform distribution. It can be observed that calibrated estimates tend to undervalue the theoretical value. The distribution of r_0 can be explained by recent developments in rail-wheel contact and consequently, expected reduction of mechanical resistance. On the other hand, higher values may be an effect of deteriorated rolling stock or a train occupancy. The quadratic parameter r_2 shows the best fitting to a Pareto distribution. From Figure 7.8b can be distinguished the variance of the aerodynamic resistance, which may be considerable while taking into account adverse weather conditions. Thereby, it may be assumed that the default value is slightly overestimated.

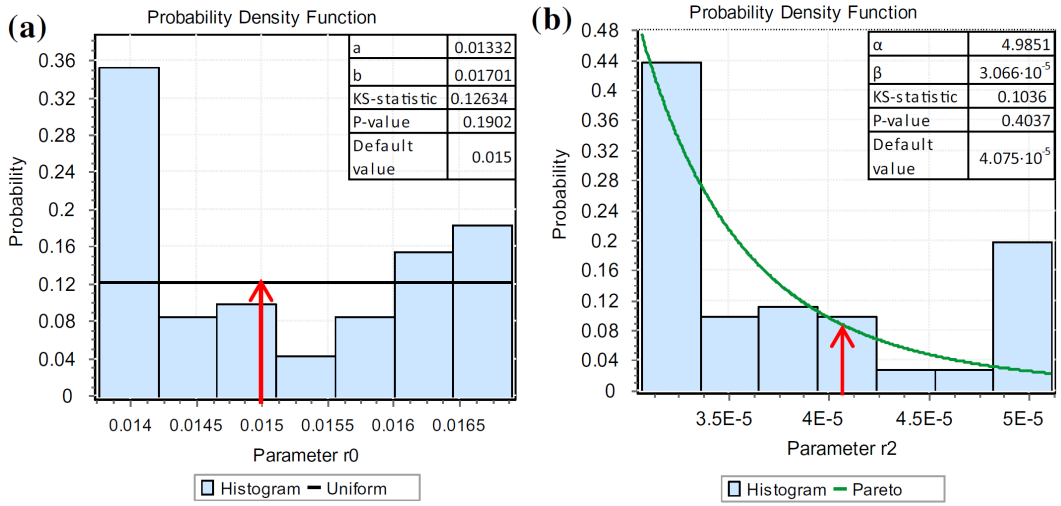

Figure 7.8: Distributions of resistance parameters

The distribution of the braking rate (due to speed restriction) is shown in Figure 7.9a. It can be observed that the most probable rate is significantly less than the default value used by the RU which is 0.66 m/s^2. Some of the higher values of the parameter can be evaluated as an error in calibrated parameters regarding the inability of the current model to detect and reconstruct coasting phases. For example, in case of coasting, a simulated train speed profile tends to have a higher speed at the approach indication than the realised train behaviour with coasting and it would consequently assume a higher braking rate. The braking parameter shows the best goodness of fit with the log-logistic distribution. This parameter shows a relevant variation over the different train runs. An explanation to this can be the consistent difference in the driving behaviour for different train drivers.

Finally, the cruising performance is depicted in Figure 7.9b. It is shown that the most part of the trains tend to run at the maximum allowed speed given by the static speed limit while some of them even overrun this limit for 1-2%. However, it has been observed that some trains run only at 80% of the maximum speed, which presents a significant diversity in the driver behaviour. This parameter fits best to the Johnson bounded distribution (Johnson, 1949).

Table 7.4 present the ranges of calibrated parameters for all the train compositions. Parameter r_1 is not given as an interval since it was not part of the calibration and set to a fixed default value, while the parameter c_1 equals zero, as provided by the RU.

Figure 7.10 illustrates results from Table 7.4 and gives a comparison with the default values of parameters provided by the RU. As can be observed the default values (yellow dots) represent neither the upper bound nor the average value of the distributions of the calibrated input parameters. This aspect can be clearly seen for the factors relative to the tractive effort, c_0 and c_2. The default values given by the RU for these parameters are usually employed for the calculation of the minimum running time, and therefore should represent the upper bound of these intervals since it is assumed that

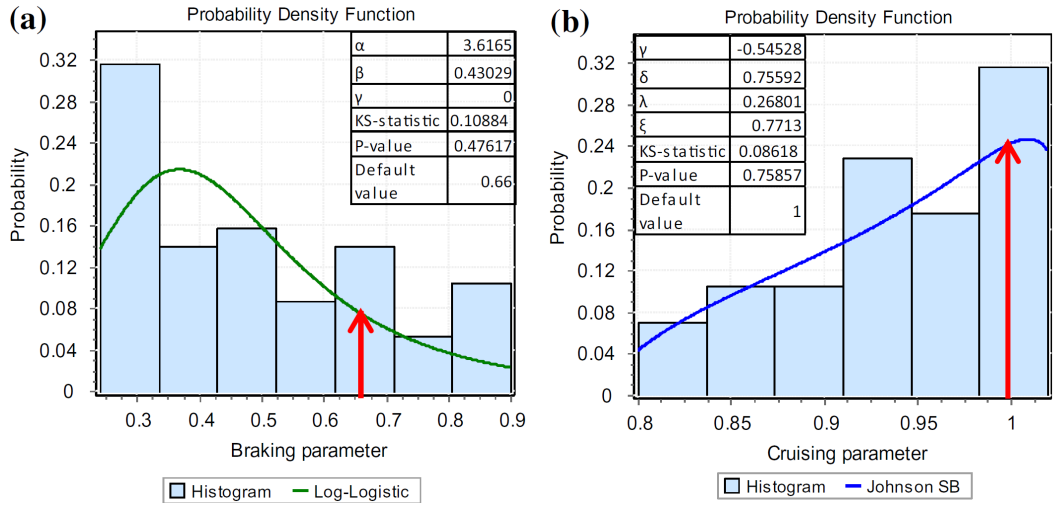

Figure 7.9: Parameter distributions for: a) braking rate, b) cruising performance

Table 7.4: Calibrated parameters for the four train compositions

Parameter	VIRM4	VIRM6	VIRM10	ICRm
c_0 [10^{-3}N/kg]	[0.251, 0.621]	[0.440, 0.600]	[0.200, 0.511]	[0.283, 0.503]
c_2[Nm/s/kg]	[3.144, 8.648]	[3.669, 7.075]	[2.555, 7.355]	[3.341, 11.792]
r_0[N/kg]	[0.014, 0.016]	[0.014, 0.016]	[0.010, 0.020]	[0.019, 0.022]
r_1 [10^{-4}Ns/m/kg]	2.162	1.939	3.341	3.342
r_2 [10^{-5}Ns^2/m^2/kg]	[3.499, 4.678]	[2.910, 3.904]	[1.774, 3.597]	[2.672, 3.616]
b_{limit}[m/s^2]	[0.24, 0.9]	[0.24, 0.9]	[0.24, 0.9]	[0.24, 0.9]
$v_{cruising}$[m/s]	[0.89, 1.02]	[0.81, 1.02]	[0.89, 0.98]	[0.81, 1.02]

the train accelerates with the maximum power of the engine. Instead, the results of the calibration experiment show the presence of train runs that overcome these values in the reality. Furthermore, parameters of the resistance equation, r_0 and r_2, supplied by the RU are within the estimated distributions, for all the train compositions. For r_0 was expected to be the lower bound of these intervals. Parameter r_2 describes the aerodynamic resistances and takes into account the effect of the wind in the same or the opposite direction of the train run. The expectations were that the default values supplied for these parameters would correspond to the means of the corresponding distributions. Nevertheless, it has been observed that the default values tend to represent slightly overestimated values comparing with the observed distributions. On the other hand, large variation intervals are revealed for the braking rate. This denotes a consistent variation in the driving behaviour of train drivers. The default value for braking rate cannot describe this aspect. Moreover, this value does not coincide with any representative value of the distribution (i.e., mean, lower or upper bound). For the cruising performance the same conclusions can be drawn as the braking rate.

7.5 Conclusions

Predictions of railway traffic are needed by designers and dispatchers respectively for the design of robust timetables and the real-time management of perturbed conditions.

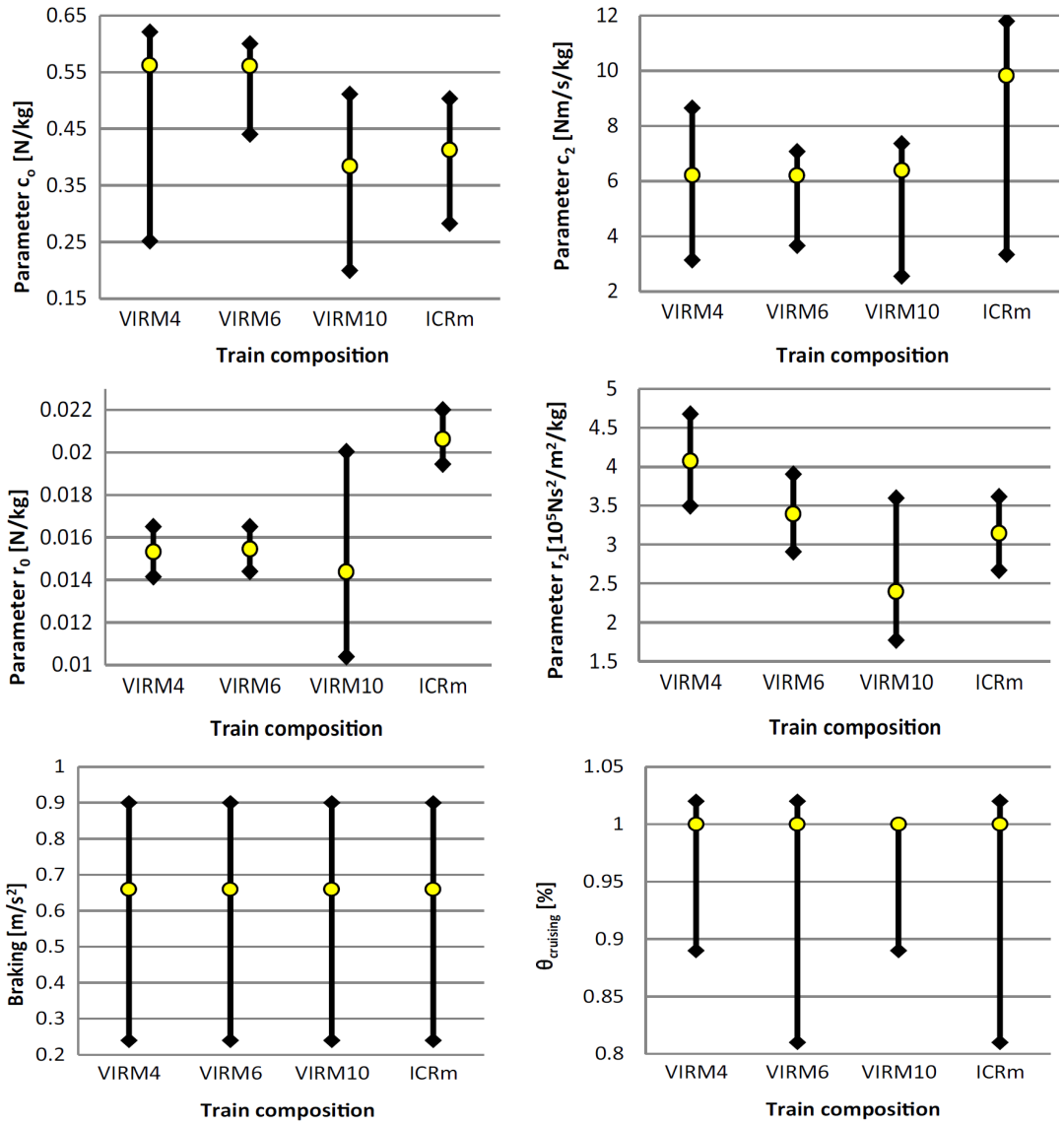

Figure 7.10: Calibrated parameters for the four train composition

These tasks can be performed effectively only when using train running time models which reliably describe actual trajectories. To this purpose the calibration of model parameters against field data is necessary.

This paper presented an approach to derive the most probable speed profiles of train runs from observed track occupation/release data. The train behaviour is modelled according to the Newton dynamic motion equations, which are numerically integrated over distance employing the Runge-Kutta method. A simulation-based optimization approach is adopted to calibrate input parameters of the equations describing the tractive effort, the motion resistances, the braking effort, and the cruising phase. These parameters are fine-tuned for different classes of train composition (defined by the number of wagons, the type of traction unit, and the length of the train) by minimizing the error between observed and simulated running times, using a genetic algo-

rithm. For each composition the calibration experiment is performed on a significant set of observed trains running along the Rotterdam-Delft corridor in the Netherlands. A probability distribution has been estimated for the input parameters of each class of composition. This aspect gives also insights in different driving behaviour adopted during real operations.

The results show that the train length estimation model obtained good computation accuracy. To this aim the error due to the delay of the release time has been distinguished. Further, the results illustrate the effectiveness of the proposed optimization method in calibrating parameters of the Newton's dynamic equations versus track occupation/release data collected at the level of track sections. It has been observed that some of the parameters of tractive effort and resistance do not affect the train behaviour significantly, i.e., the linear parameter of tractive effort as well as the linear parameter of resistance force. Furthermore, the comparison with the default parameters provided by the RU highlights that some of the default values tend to be inadequate for the calculation of the technical running time for which they are generally used. Tractive effort parameters seldom overreach the corresponding default values, therefore showing that the latter are not the absolute maximum values, but a train has an extra power reserve that can be used for faster running. On the other hand, the parameters of the resistance equation tend to be slightly overestimated based on the received distributions. The realised braking rate is significantly smoother than the default one; therefore trains traverse the braking distance faster than computed in the minimum running time. Also, train drivers do not always follow the maximum static speed limit. Instead, it has been observed that in some cases the cruising performance is just 80% of the maximum. Finally, it has been shown that a specific calibration process should be performed to understand the variation in the coefficients of the dynamic motion equations over different train runs. In this way it is possible to set more reliable values to generate stochastic running times during robust timetabling.

Instead, in a real time context the model can be used to predict train trajectories for the detection of track conflicts. In particular, the implementation may be considered in two different ways: i) By applying parameters only for a given train, we can perform a deterministic and accurate prediction of its trajectory over a certain time period ahead and consequently for a set of trains anticipate the future conflicts, ii) By using distributions for a category of trains we can identify a set of probable trajectories that a train can have over a certain period ahead. In this case we can develop a statistical conflict detection model that can derive probabilities of possible conflicts.

The current work can be extended in several ways. First, the calibration model could be performed on different lines to evaluate possible different behaviour of train drivers as well as to distinguish parameters for different train compositions. Second, it would be noteworthy to compare realised and simulated running times based on achieved stochastic parameters as well as analyse the dependency of running time and distributions of dynamic train parameters. Third, analyses to understand the train parameters variation between delayed and on-time trains can be undertaken. Moreover, the com-

putation time of the proposed simulation-based model can be enhanced by adjusting parameters of the implemented GA. Finally, proper validation of the speed profiles obtained by this model will be realized against GPS data.

Chapter 8

Conclusions and future developments

This thesis has developed automated capacity assessment and timetabling models using data analysis, simulation and optimisation to successfully generate feasible, efficient, stable and robust timetables. This chapter gives a summary of our main findings, main conclusions, the recommendations to practice and future scientific research.

8.1 Main findings

Capacity assessment. In Chapter 2, we first reviewed methods for railway capacity assessment, with the focus on methods based on timetable compression, such as UIC 406 and CUI. We then described the existing models for assessing the whole networks as well as corridors and proposed a new max-plus model for the capacity occupation assessment. The model follows the compression method and enables computation of capacity occupation in stations while also being applicable for corridors. A capacity occupation rate is considered as a stability measure, where a remaining available capacity identifies if delays could mitigate in limited time. If the capacity occupation satisfies predefined thresholds the timetable is considered to be stable. The concept of timetable stability was afterwards elaborated and applied in Chapters 3 to 6. This chapter answers research question 1.

Conceptual timetabling framework. In Chapter 3, we proposed a concept of a three-level modular performance-based timetabling framework that integrates timetable construction and evaluation. We showed the importance of including performance indicators, such as feasibility, stability and robustness, during the timetable construction. In addition, multiple models need to be considered to represent each particular indicator either at microscopic or macroscopic levels. This chapter answers research question 2.

Microscopic models. In Chapter 4, microscopic models were developed to compute reliable running and minimum headway times for a macroscopic timetabling model, as well as to check the microscopic feasibility and stability of the macroscopic timetables. Train running times are computed by integrating the Newton's motion formulae, while the accurate headway computation is based on the blocking time theory. In this

way, we obtain a fast computation of train process times and acceptable speed profiles even for very dense railway traffic. The feasibility of the timetable is checked by an efficient conflict detection and resolution model based on blocking time theory. In case of conflicts, new running and minimum headway times are automatically computed. Microscopic timetabling models are necessary to guarantee timetable feasibility and stability and thus, create an added value to the timetable planning, which answers research question 3.

Macroscopic model. In Chapter 5, a two-stage stability-to-robustness model is proposed, which is the first timetable optimization model that incorporates three important performance indicators of timetable design: efficiency, stability and robustness. Our approach incorporates a network capacity assessment model in Stage 1 with a PESP-variant of a timetabling model in Stage 2. The first stage focuses on stability, the second stage on robustness, while efficiency is considered in both stages. Five objective functions were defined to generate alternative timetables, which were evaluated a posteriori and compared with existing (single stage) PESP-based models. This chapter answers research question 4.

The two-stage stability-to-robustness model was tested on a real-life Dutch railway network. The produced timetables were, in most cases, better than the ones computed by existing models, i.e., they generated a smaller a amount of average delays. Objective functions that focus only on increasing robustness, MaxMin and HalfBuffer, tend to generate solutions that incorporate an excessive amount of time supplements and as such may be inefficient for passengers. The multi-objective models MaxBuffer, MaxMin+ and HalfBuffer+ usually created the most robust solutions that were also efficient, meaning that only limited time supplements were allocated. The results also showed that objective function MaxBuffer seems too sensitive to changing weight factors, while HalfBuffer+ tends to allow more flexibility to generate significantly different solutions. Therefore, HalfBuffer+ can be considered as a promising choice for future implementations.

Micro-macro timetabling framework. In Chapter 6, we integrated microscopic models from Chapter 4 with a macroscopic timetable model into a micro-macro timetabling framework that incorporates performance measures from Chapter 3. The resulting timetable is computed together with all measures which are either satisfied or optimized depending on the required criteria. This alleviate the time-consuming task of ex-ante simulations to test the constructed timetable on, for example, conflicts, stability and robustness. This unprecedented integrated approach that guarantees feasible, efficient, stable and robust solutions has been made possible by the advances in both microscopic and macroscopic timetable models, and also by efficient and consistent data transformations between the various levels. This enables an effective framework in which microscopic details can be combined with macroscopic optimization over large networks, including stochastic models for robustness evaluation. This micro-macro approach provides the answer to research question 5.

The modular micro-macro framework was tested on a case study from the Netherlands,

showing good results on all performance indicators. In particular, a practical application to an area of the Dutch railway network showed the ability of this framework to ensure the feasibility of the timetables at the level of track detection sections. High quality timetables were produced in a limited number of iterations (up to 15), depending on the given line plan. In addition, the UIC norms on infrastructure occupation rates were satisfied, so that for all the scenarios, we obtained a maximum occupation rate below 65%.

Data analysis. In Chapter 7, we developed a simulation-based optimization model to derive the most probable train speed profiles from observed track occupation data. The train behaviour, which includes driving parameters like tractive effort, motion resistances, braking effort, and cruising, was modelled according to the Newton's dynamic motion formulae. Train driving parameters were calibrated using a genetic algorithm by minimizing the error between observed and simulated running times. This chapter answers research question 6.

The calibration experiments were performed on a set of observed trains running along the Rotterdam-Delft corridor in the Netherlands for four train compositions (defined by the number of train units, the type of traction unit, and the length of the train). Probability distributions were estimated for the input parameters of each train composition, which gave also insight in different driving behaviour adopted during operations.

Motivated by the successful results of this thesis, we developed a railway planning toolbox with a micro-macro timetabling tool and a robustness evaluation tool. The former allows testing different timetabling scenarios by varying given line plan, timetable design parameters, and selecting different objective functions. The latter provides a basis for evaluating robustness by varying initial disturbances. These two tools constitute a safe playfield to practitioners and students to experience design and evaluation of timetables and understand effects of changing timetable design parameters. These tools are described in Appendix A.

8.2 Main conclusions

Reliable and high quality railway timetables with better customer satisfaction and reduced train delays can be achieved by: *a*. capacity assessment, *b*. macroscopic network timetable optimisation, and *c*. microscopic timetable evaluation. The thesis demonstrated that these models have to be used together in the planning process to satisfy all predefined performance indicators. The proposed performance-based timetabling approach can design efficient, conflict-free, stable and robust timetables. Stability tests assure that a timetable has sufficient buffer time to prevent or reduce delays. Moreover, the settling time of delays is explicitly incorporated in the timetable optimization in a trade-off with running, dwell and transfer times, to provide robust timetables. Microscopic conflict detection and updating guarantees that the created timetable is conflict-free. These models integrated together result in a more efficient use of capacity, higher punctuality and increased customers' satisfaction.

Second, timetable stability should be considered in the planning phase and the corresponding models should be applied. This will prevent the exhaustive a posteriori analysis of created timetables. What is more, as shown in Chapter 5, modelling stability in a macroscopic timetabling model can result in more robust timetables. Still, a single dominant objective that generates the best solutions for all timetabling instances does not exist. Instead, significant tests are necessary to determine the best objective and appropriate weight factors for objective functions. Based on performed analysis, HalfBuffer+ can be considered as a promising choice for future implementations.

Third, we can further gain on on-time performances by accounting for driver behaviour in the planning phase. On one hand, tractive effort parameters tend to overreach the corresponding default values, thus showing that a train may have an extra power reserve that can be used to accelerate faster. This would allow trains running faster than theoretical minimum running times. On the other hand, the realised braking rate is significantly lower than the default one; therefore, trains traverse the braking distance slower than computed in the minimum running time. In addition, train drivers do not always follow the maximum speed limit, but may cruise with just 80% of the maximum speed instead. This extra understanding of actual train behaviour can lead toward better punctuality and more accurate scheduled train services.

Most importantly, by guaranteeing satisfaction of all performance indicators, a created timetable can be applied directly in practice. Our integrated approach can save significant time for computing a single timetable solution, and makes possible generating multiple solution alternatives that can be tested in consecutive planning steps such as rolling stock and crew scheduling and also weighted on other (non-quantitative) criteria such as travel comfort, crowdedness, and accessibility. Decision makers can then choose the best ones that would provide the best quality of service to passengers and freight operators. Increased punctuality would also support overall better capacity use and more train services to be scheduled. This would eventually make up room for achieving goals of Better and More, High-Frequency Rail Transport Programme and projects alike.

8.3 Recommendations for practice

The integrated timetabling approach highlighted nine main recommendations that need to be considered explicitly in the design of a stable, robust, conflict-free timetable with optimal journey times:

- Microscopic calculations of running and blocking times taking into account all running route details at section level (gradients, speed restrictions, signalling),
- Microscopic conflict detection guaranteeing a conflict-free timetable,
- Timetable precision of at most 6 s to minimize capacity waste,

- Incorporation of infrastructure occupation and stability norms on corridor, node and network level,
- Macroscopic network optimization, with respect to running, dwell and transfer times, that exploits the most stable timetable structure,
- Timetable stability optimization directly in a macroscopic timetabling model to produce more robust timetables,
- Macroscopic robustness analysis using stochastic simulation to obtain a robust network timetable,
- Acceptable speed profiles computation for all trains,
- Reliable running time computation based on calibrated train driving parameters.

8.4 Future research developments

This thesis has been motivated by the general need for better mathematical models and algorithms for timetable planning in railway networks and provides high quality solutions. The potential of capacity assessment and timetabling models for improved railway timetabling stimulates further research in this field. This section points out promising research directions, which may be organized along the four main aspects: modelling, behavioural, organisational and technological.

Modelling. The future development of capacity assessment models should stay in line with the existing compression method. To make it a standard evaluation tool and apply it internationally, additional research on capacity saturation rates and measures that provide reliable services is essential. The network capacity assessment models should gain more attention, as only these are able to incorporate all interactions (headways, transfers, turnarounds) present in a timetable.

The developed timetabling framework is general and may be applied to other railway planning problems like short-term planning and maintenance planning. For example, for scheduling additional ad-hoc freight or passenger trains, the main structure of the timetable may be fixed and the proposed framework can be applied to insert extra trains. The current framework makes a first step in railway planning. As such, it assumes fixed train routes and focuses on a single planning stage. Thus, the framework could be further integrated with line planning on one side, and rolling stock and crew scheduling on the other. In addition, integrating with routing models is required to further improve the quality of timetable solutions.

More generally, the modular performance-based framework can be applied to planning and scheduling resources in other industries, such as air and road transport systems (Jacquillat, 2015; Van de Weg, Hegyi, & Hoogendoorn, 2014), and telecommunication systems (Cohen & Katzir, 2010).

In order to better understand and evaluate timetable robustness, future research could introduce a microscopic robustness analysis of computed timetables. These models would include detailed infrastructure data and realistic train movement characteristics and driver behaviour. For this, we need to develop stochastic running time models based on train parameter distributions. Consequently, stochastic buffer times could be used to propose new and more sophisticated robustness measures.

Most of the proposed optimization models for timetabling are deterministic so far. In future, more emphasis should be addressed on developing stochastic optimization models which will directly consider variability of running and dwell times. What is more, we should be able to integrate the actual (stochastic) passenger demand directly in the timetabling process.

The ability of recovering as quickly as possible after a disruption determines the system resilience. However, it is often overshadowed or even disregarded in the planning process, which means that current timetables may involve a structural lack of operational resilience to cope with disruptions. The consequence is that current planning measures do not fully meet customers' requirements allowing strong propagation of delays and service cancellations in case of disturbances and disruptions of the network. To prevent this, we would need to aim also at resilience already in the planning phase.

Behavioural. The calibration of train parameters should be performed on different lines to determine possible different behaviour of train drivers related to the surrounding traffic and the geographical part of the network. In addition, rising availability and use of advanced train positioning systems could be used to provide more detailed data and allow more accurate calibrations. We could use this additional knowledge to understand and estimate driver behaviour in various conditions, such as congested versus non-congested networks, and during regular (on-time) services versus disturbance scenarios. Such calibrated train parameters should be used for timetable planning and real-time models.

Organizational. Differences in objectives severely limit actors' (infrastructure manager - IM and railway undertakings - RUs) ability to jointly plan and operate the rail system in a well-coordinated manner (Steenhuisen, 2009). For example, the IM focuses firstly on availability of infrastructure, quality of capacity allocation, safety and non-discriminatory access to RUs. Passenger RUs, instead, value mostly punctuality, growth of passenger numbers and safety, while freight RUs prefer availability of paths and flexibility in choosing from multiple options. Also, when looking at the planning time-scale, the needs of freight and passenger RUs differ as the former wants flexibility for short-term scheduling, while the latter expect planning to be done well in advance. Therefore, new organization and planning concepts are needed for improving communication between actors in addressing their needs more transparently and integrating them in the operational planning.

Technological. Current models consider always the existing fixed-block signalling systems. However, with increasing level of traffic automation (e.g., automated trains) and

new moving-block signalling systems (e.g., ETCS Level 3), the railway traffic behaviour may become more flexible and possibly more similar to road traffic today. This may incur different concepts of train interactions, as well as computing running and headway times, which could not be represented in the current timetabling models. Therefore, new timetabling models should consider such new technologies and be able to exploit the existing resources in the best possible way.

The current framework together with these future developments would eventually allow complete integration of automated planning models, from long-term to operational, in future railway decision support systems. This would result in better railway operations that use infrastructure capacity most efficiently, provide conflict-free and efficient services and reduce overall delays to both passenger and freight trains.

Appendices

Appendix A

Railway planning toolbox STAFER

This appendix demonstrates the railway planning toolbox STAFER (Scheduling for Today's Advanced, Flexible and Effective Railways) for designing and evaluating railway timetables. STAFER incorporates two main tools: Micro-macro timetabling for designing improved railway timetables and RobEval for evaluating timetable robustness. The modular structure of STAFER allows alternative optimization and evaluation algorithms and further planning steps to be easily added.

Micro-macro timetabling tool. The core of the Micro-macro timetabling tool is the performance-based framework from Chapter 3. The tool integrates microscopic models from Chapter 4 for computing running and headway times and evaluating feasibility and stability. For network optimization, we implemented the macroscopic timetabling model PESP-N from Chapter 5.3. The modular structure of the micro-macro framework allows using different timetabling models. During development, we also tested the framework with the stability-to-robustness model from Chapter 5 and reported it in Bešinović, Goverde, and Quaglietta (2016).

Figure A.1 shows the graphical user interface (GUI) of the micro-macro timetabling tool. The GUI consists of three building blocks: Input, Graphical output and Statistics.

Input includes selection of timetable design parameters, minimum and maximum running time rates (in %), minimum buffer time (in seconds), and a preferable objective function. Objective functions can be minimizing efficiency (i.e., train journey times), maximizing robustness (i.e., total buffer between train events) or a combination of both.

Graphical output consists of three types of diagrams (see Figure A.2). First, the time-distance diagram is shown for a selected corridor. Second, by selecting a train line (solid green line) in the time-distance diagram, the corresponding blocking time diagram appears. Note that only infrastructure occupation related to the selected train line is pictured to explicitly represent conflict-freeness of a generated timetable. Third, the computed minimum and operational speed profiles for the selected line are shown. By analysing these diagrams, you could assure that a designed timetable includes acceptable speed profiles and is conflict-free.

Statistics report the characteristic values (minimum, mean and maximum) for scheduled running time supplements and train journey times over all lines. The button Line statistics opens more detailed statistics defined per train line (see Figure A.3). In addition, the number of iterations needed to design a feasible, stable and robust solution is produced by pressing the Iterations button (see Figure A.4). Clicking the Evaluate robustness button, activates the RobEval tool.

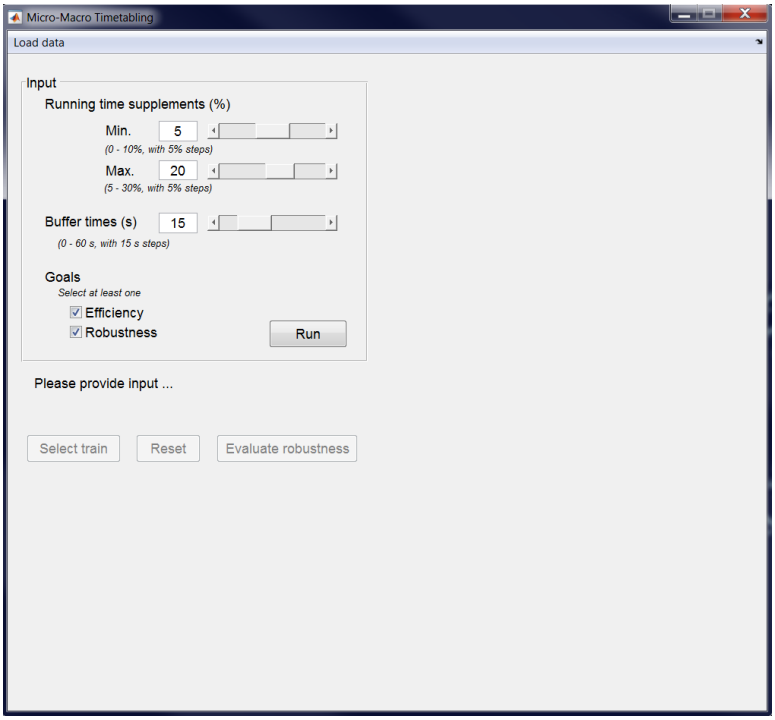

Figure A.1: Graphical interface of Micro-Macro timetabling tool

Robustness evaluation tool. The robustness evaluation tool RobEval integrates the delay propagation model from Chapter 5.4.4. The input timetable to RobEval is one generated by the Micro-macro timetabling tool, or it can be loaded directly within the GUI.

Figure A.5 shows the graphical user interface of the developed RobEval tool. The GUI consists of three building blocks: Settings, Graphical output and Statistics.

Input provides defining an average input delay (in %), the number of simulations and the considered time horizon. Graphical output gives a selection of three possible plots: the time-distance diagram with all computed scenarios (Figure A.6 top left), the time-distance bandwidth diagram with the corresponding mean running times (Figure A.6 top right) and network layout with visualised total delays in stations (Figure A.6 bottom). Statistics report gives the total delay per hour and detailed total delays per train line and per station (Figure A.7).

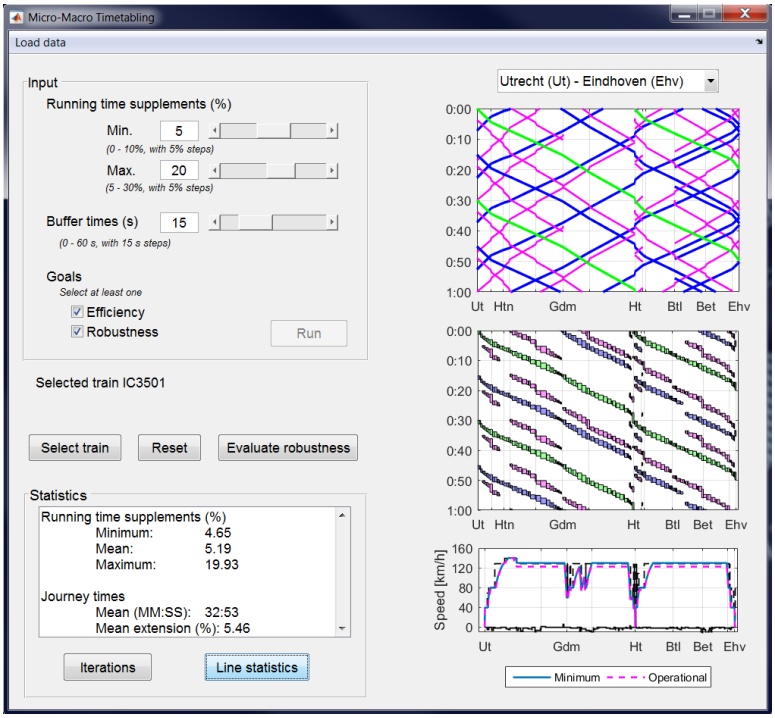

Figure A.2: Graphical output of Micro-macro timetabling tool

Statistics per line

Line	Min. running time supplement (%)	Mean running time supplement (%)	Max. running time supplement (%)	Journey time (MM:SS)
5230	4.98	5.05	5.12	27:12
3500	4.91	4.98	5.08	47:03
800	4.91	4.98	5.08	47:03
9600	4.9	4.97	5.04	23:34
1900	4.86	4.94	5.04	21:29
4400	4.9	4.94	4.96	34:12
16000	5.04	5.07	5.1	35:49
13601	4.65	4.86	5.07	13:59
9601	4.98	5.06	5.12	24:37
3601	4.83	5.01	5.16	43:26
4401	4.98	5.02	5.08	35:12
1901	4.93	9.98	19.93	23:00
5201	4.94	4.96	4.96	27:48
13600	4.69	4.82	4.95	13:11
3600	4.96	5.05	5.15	41:00
6000	4.9	5.05	5.15	31:27
3501	4.96	5.05	5.23	49:44
801	4.83	4.99	5.09	50:09
16001	4.95	4.97	4.99	37:08
6001	4.95	5.01	5.08	30:40

Figure A.3: Line statistics for the designed timetable

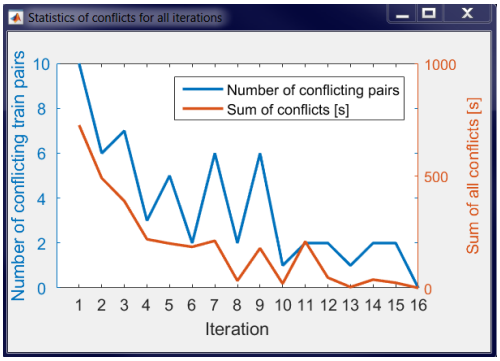

Figure A.4: Number of iterations needed

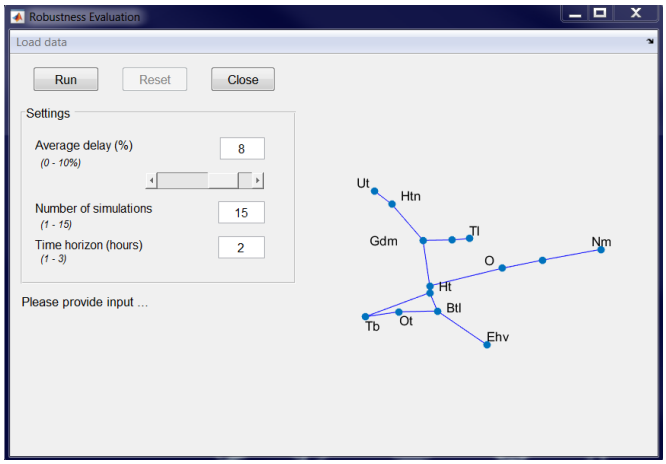

Figure A.5: Graphical interface of robustness evaluation tool

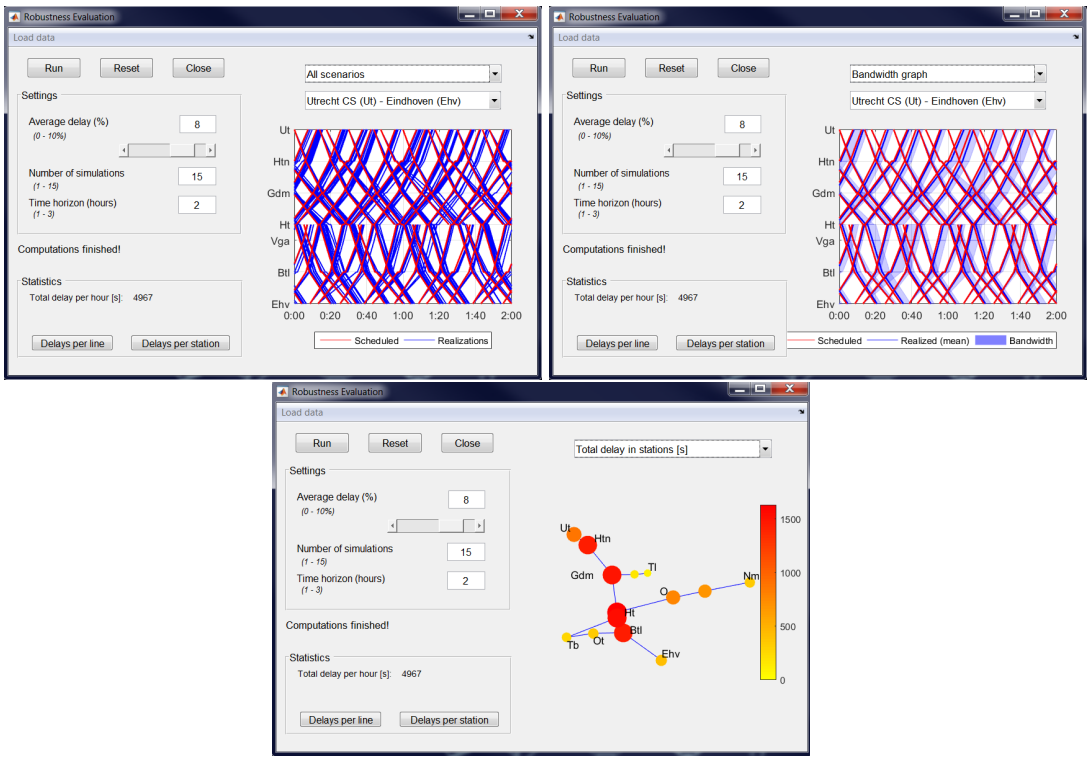

Figure A.6: Graphical output of robustness evaluation tool

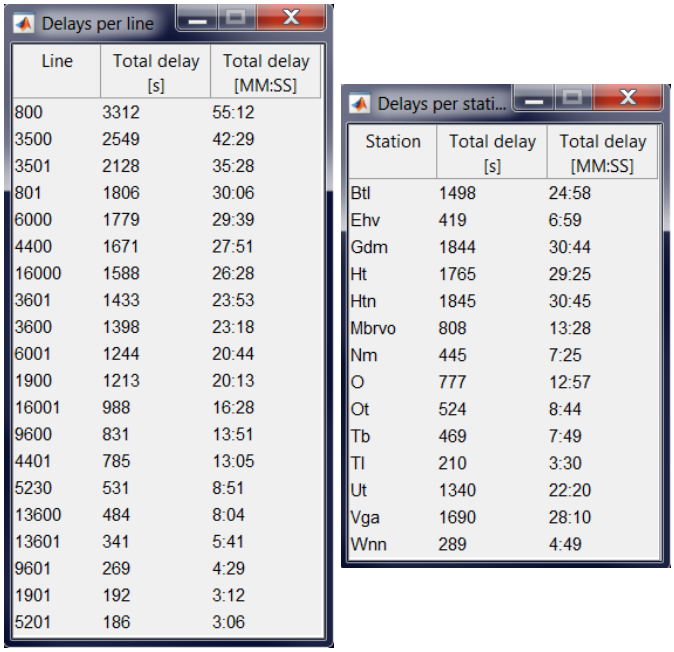

Figure A.7: Statistics reports of robustness evaluation

Bibliography

Abril, M., Barber, F., Ingolotti, L., Salido, M. A., Tormos, M. P., & Lova, A. (2008). An assessment of railway capacity. *Transportation Research Part E: Logistics and Transportation Review*, *44*(5), 774–806.
(Cited on pages 13, 16, and 18.)

Abril, M., Salido, M. A., Barber, F., Ingolotti, L., Lova, A., & Tormos, M. P. (2005). A heuristic technique for the capacity assessment of periodic trains. *Frontiers in Artificial Intelligence and Applications*, *131*, 339-346.
(Cited on pages 13 and 18.)

Albrecht, A. R., Howlett, P. G., Pudney, P. J., & Vu, X. (2013). Energy-efficient train control: From local convexity to global optimization and uniqueness. *Automatica*, *49*(10), 3072-3078.
(Cited on page 65.)

Albrecht, T. (2005). Energy-efficient train control in suburban railways: Experiences gained from onboard tests of a driver assistance system. In *Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis (RailDelft 2005).*
(Cited on pages 48, 49, 50, and 52.)

Albrecht, T. (2014). Energy-efficient train operation. In I. Hansen & J. Pachl (Eds.), *Railway timetabling & operations* (p. 91-116). Hamburg: Eurailpress.
(Cited on pages xv and 50.)

Albrecht, T., Gassel, C., Binder, A., & van Luipen, J. (2010). Dealing with operational constraints in energy efficient driving. In *Proceedings of the 4th IET International Conference on Railway Traction Systems (RTS 2010).* Birmingham.
(Cited on pages 161, 162, and 170.)

Albrecht, T., Gassel, C., Knijff, J., & van Luipen, J. (2010). Analysis of energy consumption and traffic flow by means of track occupation data. In *Proceedings of the 4th IET International Conference on Railway Traction Systems (RTS 2010).* Birmingham.
(Cited on pages 161 and 162.)

Albrecht, T., Goverde, R. M. P., Weeda, V. A., & Van Luipen, J. (2006). Reconstruction of train trajectories from track occupation data to determine the effects of a driver information system. In J. Allan, C. A. Brebbia, A. F. Rumsey, G. Sciutto, & S. a. Sone (Eds.), *Computers in Railways X* (p. 207-216). 88, WIT Press, Southampton: WIT Transactions on The Built Environment.

(Cited on pages 161 and 162.)

Armstrong, J., Preston, J., & Hood, I. (2015). Evaluating capacity utilisation and its upper limits at railway nodes. In *Proceedings of the 13th Conference on Advanced Systems in Public Transport (CASPT2015).* Rotterdam, The Netherlands, 19-23 July 2015.
(Cited on pages 17 and 65.)

Barrena, E., Canca, D., Coelho, L. C., & Laporte, G. (2014a). Exact formulations and algorithm for the train timetabling problem with dynamic demand. *Computers & Operations Research*, *44*, 66-74.
(Cited on page 98.)

Barrena, E., Canca, D., Coelho, L. C., & Laporte, G. (2014b). Single-line rail rapid transit timetabling under dynamic passenger demand. *Transportation Research Part B: Methodological*, *70*, 134-150.
(Cited on pages 37 and 98.)

Bellman, R. (1957). *Dynamic programming*. Princeton, USA: Princeton University Press.
(Cited on page 53.)

Bergmann, D. (1975). Integer programming formulation for deriving minimum dispatch intervals on a guideway accommodating through and local public transportation services. *Transportation Planning and Technology*, *3*(1), 27-30.
(Cited on page 96.)

Bešinović, N., Goverde, R. M. P., & Quaglietta, E. (2016). A novel approach for automated timetable planning. In *Proceedings of the 11th World Congress on Railway Research (WCRR) 2016.* Milan, Italy, 29 May-2 June 2016.
(Cited on page 191.)

Bešinović, N., Goverde, R. M. P., & Quaglietta, E. (2017). Microscopic models and network transformations for automated railway traffic planning. *Computer-Aided Civil and Infrastructure Engineering*, *32*(2), 89–106.
(Cited on pages 14, 38, 42, 99, 104, 137, and 139.)

Bešinović, N., Goverde, R. M. P., Quaglietta, E., & Roberti, R. (2016). An integrated micro-macro approach to robust railway timetabling. *Transportation Research Part B: Methodological*, *87*, 14-32.
(Cited on pages 35, 38, 42, 45, 46, 47, 85, 86, 88, 96, and 114.)

Bešinović, N., Quaglietta, E., & Goverde, R. M. P. (2013). A simulation-based optimization approach for the calibration of dynamic train speed profiles. *Journal of Rail Transport Planning and Management*, *3*(4), 126-136.
(Cited on pages 75 and 137.)

Bešinović, N., Quaglietta, E., & Goverde, R. M. P. (2014). Supporting tools for automated timetable planning, in: C. In A. Brebbia, N. Tomii, P. Tzieropoulos, & J. M. Mera (Eds.), *Computers in Railways XIV* (p. 565-576). Southampton: WIT Press.
(Cited on pages 38 and 66.)

Binder, A., & Albrecht, T. (2013). Timetable evaluation and optimization under con-

sideration of the stochastical influence of the dwell times. In *Proceedings of the 3rd International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS) 2013* (p. 471-481). Dresden, Germany, 2-4 December 2013.
(Cited on pages 51 and 55.)

Borndörfer, R., Hoppmann, H., & Karbstein, M. (2016a). Passenger routing for periodic timetable optimization. *Public Transport*, 1–21.
(Cited on page 95.)

Borndörfer, R., Hoppmann, H., & Karbstein, M. (2016b). Separation of cycle inequalities for the periodic timetabling problem. In *Lipics-leibniz international proceedings in informatics* (Vol. 57).
(Cited on page 95.)

Borndörfer, R., & Schlechte, T. (2007). Models for railway track allocation. In *7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization and Systems (ATMOS'07)*. OpenAccess Series in Informatics (OASIcs).
(Cited on page 44.)

Bosschaart, M., Quaglietta, E., Janssen, B., & Goverde, R. M. P. (2015). Efficient formalization of railway interlocking data in railml. *Information Systems*, *49*, 126-141.
(Cited on pages 34 and 130.)

Brännlund, U., Lindberg, P. O., Nou, A., & Nilsson, J. E. (1998). Railway timetabling using lagrangian relaxation. *Transportation Science*, *32*(4), 358-369.
(Cited on page 30.)

Brünger, O., & Dahlhaus, E. (2014). Running time estimation. In I. A. Hansen & J. Pachl (Eds.), *Railway Timetabling and Operations*. Hamburg: Eurailpress.
(Cited on page 75.)

Büker, T., & Seybold, B. (2012). Stochastic modelling of delay propagation in large networks. *Journal of Rail Transport Planning & Management*, *2*(12), 34–50.
(Cited on page 12.)

Burdett, R. L. (2015). Multi-objective models and techniques for analysing the absolute capacity of railway networks. *European Journal of Operational Research*, *245*(2), 489 - 505.
(Cited on page 12.)

Burdett, R. L., & Kozan, E. (2006). Techniques for absolute capacity determination in railways. *Transportation Research Part B: Methodological*, *40*(8), 616–632.
(Cited on pages 12 and 13.)

Bussieck, M. R., Winter, T., & Zimmermann, U. T. (1997). Discrete optimization in public rail transport. *Mathematical Programming*, *79*, 415-444.
(Cited on page 30.)

Butcher, J. C. (2008). *Numerical methods for ordinary differential equations, second edition*. London: Wiley.
(Cited on pages 75, 160, and 169.)

Cacchiani, V., Caprara, A., & Fischetti, M. (2012). A lagrangian heuristic for robust-

ness, with an application to train timetabling. *Transportation Science*, *46*(1), 124-133.
(Cited on pages 45 and 98.)

Cacchiani, V., Caprara, A., & Toth, P. (2008). A column generation approach to train timetabling on a corridor. *4OR*, *6*, 125-142.
(Cited on page 44.)

Cacchiani, V., Caprara, A., & Toth, P. (2010). Scheduling extra freight trains on railway networks. *Transportation Research Part B: Methodological*, *44*, 215-231.
(Cited on pages 9, 43, 44, and 152.)

Cacchiani, V., Huisman, D., Kidd, M., Kroon, L., Toth, P., Veelenturf, L., & Wagenaar, J. (2014). An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, *63*, 15-37.
(Cited on page 65.)

Cacchiani, V., & Toth, P. (2012). Nominal and robust train timetabling problems. *European Journal of Operational Research*, *219*, 727-737.
(Cited on pages 3, 30, 43, 64, 96, 98, and 128.)

Caimi, G. (2009). *Algorithmic decision support for train scheduling in a large and highly utilised railway network*. PhD Thesis, ETH Zurich.
(Cited on pages 3 and 4.)

Caimi, G., Chudak, F., Fuchsberger, M., Laumanns, M., & Zenklusen, R. (2011). A new resource-constrained multicommodity flow model for conflict-free train routing and scheduling. *Transportation Science*, *45*(2), 212-227.
(Cited on page 30.)

Caimi, G., Fuchsberger, M., Laumanns, M., & Schüpbach, K. (2011). A multi-level framework for generating train schedules in highly utilised networks. *Public Transport*, *3*(1), 3-24.
(Cited on page 64.)

Caimi, G., Fuchsberger, M., Laumanns, M., & Schüpbach, K. (2011). Periodic railway timetabling with event flexibility. *Networks*, *57*(1), 3-18.
(Cited on pages 98, 112, and 128.)

Caprara, A., Fischetti, M., & Toth, P. (2002). Modeling and solving the train timetabling problem. *Operations Research*, *50*, 851–861.
(Cited on pages 43, 44, and 127.)

Caprara, A., Kroon, L., Monaci, M., Peeters, M., & Toth, P. (2007). Passenger railway optimization. In C. Barnhart & G. Laporte (Eds.), *Handbooks in operations research and management science* (p. 129-187). Amsterdam: Transportation, Elsevier.
(Cited on page 30.)

Castillo, E., Gallego, I., Sánchez-Cambronero, S., Menéndez, J. M., Rivas, A., Nogal, M., & Grande, Z. (2015). An alternate double-single track proposal for high speed peripheral railway lines. *Computer-Aided Civil And Infrastructure Engineering*, *30*, 181-201.
(Cited on page 64.)

Cerreto, F., Nielsen, O. A., Harrod, S., & Nielsen, B. F. (2016). Causal analysis of railway running delays. In *Proceedings of the 11th World Congress on Railway Research (WCRR) 2016.* Milan, Italy, 29 May-2 June 2016.
(Cited on page 103.)

Čičak, M., Mlinarić, T. J., & Abramović, B. (2012). Methods for determining throughput capacity of railway lines using coefficients of elimination. *PROMET - Traffic & Transportation*, *16*(2), 63–69.
(Cited on pages 13 and 18.)

Cochet-Terrasson, J., Cohen, G., Gaubert, S., McGettrick, M., & Quadrat, J.-P. (1998). Numerical computation of spectral elements in max-plus algebra. In *Proceedings of the ifac conference on system structure and control.*
(Cited on page 26.)

Cohen, R., & Katzir, L. (2010). Computational analysis and efficient algorithms for micro and macro ofdma downlink scheduling. *IEEE/ACM Transactions on Networking*, *18*, 15-26.
(Cited on page 185.)

Cordeau, J. F., Toth, P., & Vigo, D. (1998). A survey of optimization models for train routing and scheduling. *Transportation Science*, *32*(4), 380-404.
(Cited on page 30.)

Corman, F., & Meng, L. (2015). A review of online dynamic models and algorithms for railway traffic management. *IEEE Transactions on Intelligent Transportation Systems*, *16*(3), 1274-1284.
(Cited on page 30.)

Corman, F., & Quaglietta, E. (2015). Closing the loop in real-time railway control: Framework design and impacts on operations. *Transportation Research Part C: Emerging Technologies*, *54*, 15-39.
(Cited on page 30.)

Cucala, A., Fernandez, A., & Sicre, C. (2012). Fuzzy optimal schedule of high speed train operation to minimize energy consumption with uncertein delays and driver's behavioral response. *Engineering Applications of Artificial Intelligence*, *25*(8), 1548-1557.
(Cited on page 49.)

Daamen, W., Goverde, R. M. P., & Hansen, I. A. (2009). Non-discriminatory automatic registration of knock-on train delays. *Networks and Spatial Economics*, *9*, 47-61.
(Cited on page 161.)

D'Ariano, A., Pacciarelli, D., & Pranzo, M. (2007). A branch and bound algorithm for scheduling trains on a railway network. *European Journal of Operational Research*, *183*(2), 643-657.
(Cited on page 43.)

D'Ariano, A., Pranzo, M., & Hansen, I. A. (2007). Conflict resolution and train speed co- ordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems*, *8*(2), 208-222.

(Cited on page 30.)

De Fabris, S., Longo, G., Medeossi, G., & Pesenti, R. (2013). Application and validation of a timetabling algorithm to a large italian network. In *Proceedings of the 5th International Seminar on Railway Operations Modelling and Analysis (RailCopenhagen)* (p. 13-15). Copenhagen.
(Cited on pages 64 and 128.)

De Goffau, W. (2013). *Rood seinnaderingen (Red signal approaches)*. ProRail.
(Cited on pages 2 and 103.)

Delorme, X., Gandibleux, X., & Rodriguez, J. (2009). Stability evaluation of a railway timetable at station level. *European Journal of Operational Research*, *195*(3), 780–790.
(Cited on pages 12 and 16.)

Egmond, R. J. (2000). An algebraic approach for scheduling train movements. In *Computer-Aided Scheduling of Public Transport (CASPT) 2000.* Berlin, 21-23 June.
(Cited on page 83.)

Ekman, J. (2011). Kaban–a tool for analysis of railway capacity. *Computational Methods and Experimental Measurements XV*, 693–702.
(Cited on page 23.)

Eliasson, J., & Börjesson, M. (2014). On timetable assumptions in railway investment appraisal. *Transport Policy*, *36*, 118–126.
(Cited on page 12.)

Fischetti, M., & Monaci, M. (2009). Light robustness. In R. Ahuja, R. Möhring, & C. Zaroliagis (Eds.), *Robust and Online Large-Scale Optimization, Lecture Notes in Computer Science, 5868* (p. 61-84). Berlin: Springer.
(Cited on pages 45 and 98.)

Gassel, C., & Albrecht, T. (2009). The impact of request stops on railway operations. In *Proceedings of 3rd International Seminar on Railway Operations Modelling and Analysis (RailZurich).*
(Cited on page 48.)

Gattermann, P., Großmann, P., Nachtigall, K., & Schöbel, A. (2016). Integrating passengers' routes in periodic timetabling: A sat approach. In *Oasics-openaccess series in informatics* (Vol. 54).
(Cited on page 95.)

Gaubert, S., & Mairesse, J. (1999). Modeling and analysis of Timed Petri Nets using heaps of pieces. *IEEE Transactions on Automatic Control*, *44*, 683-697.
(Cited on pages 18, 42, 82, 83, and 139.)

Ghoseiri, K., Szidarovszky, F., & Asgharpour, M. (2004). A multi-objective train scheduling model and solution. *Transportation Research Part B: Methodological*, *38*(10), 927-952.
(Cited on page 48.)

Gibson, S., Cooper, G., & Ball, B. (2002). Developments in transport policy: The evolution of capacity charges on the uk rail network. *Journal of Transport Eco-*

nomics and Policy, *36*(2), 341-354.
(Cited on page 16.)

Gille, A., Klemenz, M., & Siefer, T. (2008). Applying multiscaling analysis to detect capacity resources in railway networks. *Computers in Railways XI*, 595-604.
(Cited on pages 30 and 64.)

Goerigk, M. (2015). Exact and heuristic approaches to the robust periodic event scheduling problem. *Public Transport*, *7*, 101-119.
(Cited on pages 97, 98, and 99.)

Goerigk, M., & Schöbel, A. (2013). Improving the modulo simplex algorithm for large-scale periodic timetabling. *Computers & Operations Research*, *40*, 1363-1370.
(Cited on page 95.)

Goerigk, M., & Schöbel, A. (2014). Recovery-to-optimality: A new two-stage approach to robustness with an application to aperiodic timetabling. *Computers & Operations Research*, *52*, 1–15.
(Cited on page 97.)

Goverde, R. M. P. (2005). *Punctuality of railway operations and timetable stability analysis*. PhD Thesis, Delft University of Technology.
(Cited on pages 25 and 26.)

Goverde, R. M. P. (2007). Railway timetable stability analysis using max-plus system theory. *Transportation Research Part B: Methodological*, *41*(2), 179–201.
(Cited on pages 13, 18, 23, 25, 65, 82, 95, and 104.)

Goverde, R. M. P. (2010). A delay propagation algorithm for large-scale railway traffic networks. *Transportation Research Part C: Emerging Technologies*, *18*(3), 269–287.
(Cited on pages 23, 41, 61, 103, and 104.)

Goverde, R. M. P., Bešinović, N., Binder, A., Cacchiani, V., Quaglietta, E., Roberti, R., & Toth, P. (2016). A three-level framework for performance-based railway timetabling. *Transportation Research Part C: Emerging Technologies*, *67*, 62-83.
(Cited on pages 66, 67, 96, 128, and 129.)

Goverde, R. M. P., Corman, F., & D'Ariano, A. (2013). Railway line capacity consumption of different railway signalling systems under scheduled and disturbed conditions. *Journal of Rail Transport Planning & Management*, *3*(3), 78–94.
(Cited on pages 13 and 40.)

Goverde, R. M. P., & Hansen, I. A. (2013). Performance indicators for railway timetables. In *IEEE International Conference on Intelligent Rail Transportation (ICIRT) 2013* (pp. 301–306).
(Cited on pages 13, 30, 31, 33, 66, and 95.)

Goverde, R. M. P., & Meng, L. (2011). Advanced monitoring and management information of railway operations. *Journal of Rail Transport Planning and Management*, *1*, 69–79.
(Cited on page 161.)

Hansen, I. A., & Pachl, J. (2008). *Railway Timetable and Traffic*. Hamburg: Eurailpress.
(Cited on pages 161, 166, and 167.)

Hansen, I. A., & Pachl, J. (2014). *Railway Timetabling & Operations: Analysis, Modelling, Optimisation, Simulation, Performance Evaluation*. Hamburg: Eurailpress.
(Cited on pages 6, 13, 30, 32, 39, 40, 47, 77, 79, 102, 137, and 144.)

Harrod, S. (2009). Capacity factors of a mixed speed railway network. *Transportation Research Part E: Logistics and Transportation Review*, *45*(5), 830 - 841.
(Cited on page 13.)

Heidergott, B., & de Vries, R. (2001). Towards a (max,+) control theory for public transportation networks. *Discrete Event Dynamic Systems*, *11*(4), 371–398.
(Cited on page 23.)

Heidergott, B., Olsder, G. J., & van der Woude, J. (2014). *Max-Plus at Work: Modeling and Analysis of Synchronized Systems: A Course on Max-Plus Algebra and its Applications*. Princeton University Press, Princeton, New Jersey, USA.
(Cited on pages 18, 23, 25, 26, and 95.)

Hertel, G., & Steckel, J. (1992). Fahrzeitberechnung unter stochastischem Aspekt. *Eisenbahningenieur*, *43*, 304-306.
(Cited on pages 161 and 162.)

Heydar, M., Petering, M. E. H., & Bergmann, D. R. (2013). Mixed-integer programming for minimizing the period of a cyclic railway timetable for a single track with two train types. *Computers & Industrial Engineering*, *66*(1), 171-185.
(Cited on pages 96, 98, and 99.)

Howlett, P., & Pudney, P. J. (1995). *Energy-Efficient Train Control*. London: Springer.
(Cited on page 50.)

Hu, J., Li, H., Meng, L., & Xu, X. (2013). Modeling capacity of urban rail transit network based on bi-level programming. In *2013 Joint Rail Conference.*
(Cited on page 13.)

Huisman, T., Boucherie, R. J., & van Dijk, N. M. (2002). A solvable queueing network model for railway networks and its validation and applications for the netherlands. *European Journal of Operational Research*, *142*(1), 30 - 51.
(Cited on page 12.)

Jacquillat, A. (2015). *Integrated allocation and utilization of airport capacity to mitigate air traffic congestion*. PhD thesis, Massachusetts Institute of Technology.
(Cited on page 185.)

Jensen, L. W., Landex, A., Nielsen, O. A., Kroon, L. G., & Schmidt, M. (2017). Strategic assessment of capacity consumption in railway networks: Framework and model. *Transportation Research Part C: Emerging Technologies*, *74*, 126 - 149.
(Cited on pages 11 and 16.)

Johnson, N. L. (1949). Systems of frequency curves generated by methods of translation. *Biometrika*, *36*, 149-176.

(Cited on page 176.)

Kecman, P., Corman, F., D'Ariano, A., & Goverde, R. M. P. (2013). Rescheduling models for railway traffic management in large-scale networks. *Public Transport*, *5*(1-2), 95-123.
(Cited on page 128.)

Kecman, P., & Goverde, R. M. P. (2012). Process mining of train describer event data and automatic conflict identification. In C. A. Brebbia, N. Tomii, P. Tzieropoulos, & J. M. Mera (Eds.), *Computers in Railways XIII* (p. 227-238). 127, WIT Press, Southampton: WIT Transactions on The Built Environment.
(Cited on page 161.)

Kecman, P., & Goverde, R. M. P. (2013). Process mining approach for recovery of realized train paths and route conflict identification. In *The 92nd Annual Meeting of Transportation Research Board.* Washington, DC, USA.
(Cited on page 162.)

Kraay, D., Harker, P. T., & Chen, B. (1991). Optimal pacing of trains in freight railroads: model formulation and solution. *Operations Research*, *39*(1), 82-99.
(Cited on page 48.)

Kroon, L. G., Huisman, D., Abbink, E., Fioole, P. J., Fischetti, M., Maróti, G., ... Ybema, R. (2009). The New Dutch Timetable: The OR revolution. *Interfaces*, *39*, 6-17.
(Cited on pages 3, 64, and 128.)

Kroon, L. G., Huisman, D., & Maróti, G. (2014). Optimisation models for railway timetabling. In I. A. Hansen & J. Pachl (Eds.), *Railway timetabling and operations, second edition,* (pp. 155–174). Hamburg: Eurailpress.
(Cited on page 114.)

Kroon, L. G., Maróti, G., Retel Helmrich, M. J., Vromans, M., & Dekker, R. (2008). Stochastic improvement of cyclic railway timetables. *Transportation Research Part B: Methodological*, *42*(6), 553-570.
(Cited on pages 45, 95, 97, 98, and 99.)

Kroon, L. G., & Peeters, L. W. (2003). A variable trip time model for cyclic railway timetabling. *Transportation Science*, *37*(2), 198-212.
(Cited on page 95.)

Kroon, L. G., Peeters, L. W., Wagenaar, J. C., & Zuidwijk, R. (2013). Flexible connections in PESP models for cyclic passenger railway timetabling. *Transportation Science*, *48*(1), 136-154.
(Cited on page 95.)

Krueger, H. (1999). Parametric modeling in rail capacity planning. In *Winter Simulation Conference Proceedings 1999* (Vol. 2, pp. 1194–1200). Phoenix, Arizona.
(Cited on page 12.)

Kümmling, M., Großmann, P., Nachtigall, K., Opitz, J., & Weiß, R. (2015). A state-of-the-art realization of cyclic railway timetable computation. *Public Transport*, *7*(3), 281-293.
(Cited on page 95.)

Lai, Y.-C., & Barkan, C. (2009). Enhanced parametric railway capacity evaluation tool. *Transportation Research Record: Journal of the Transportation Research Board*, *2117*, 33–40.
(Cited on page 12.)

Landex, A. (2009). Evaluation of railway networks with single track operation using the UIC 406 capacity method. *Network and Spatial Economics*, *9*(1), 7-23.
(Cited on pages 13 and 18.)

Landex, A., & Jensen, L. W. (2013). Measures for track complexity and robustness of operation at stations. *Journal of Rail Transport Planning & Management*, *3*(1-2), 22-35.
(Cited on page 65.)

Liebchen, C. (2008). The first optimized railway timetable in practice. *Transportation Science*, *42*(4), 420-435.
(Cited on pages 95 and 113.)

Liebchen, C., Lübbecke, M., Möhring, R., & Stiller, S. (2009). The concept of recoverable robustness, linear programming recovery, and railway applications. In R. Ahuja, R. Möhring, & C. Zaroliagis (Eds.), *Robust and Online Large-Scale Optimization* (p. 1-27). Lecture Notes in Computer Science, 5868 Springer-Verlag, Berlin Heidelberg.
(Cited on pages 45 and 98.)

Liebchen, C., & Peeters, L. (2009). Integral cycle bases for cyclic timetabling. *Discrete Optimization*, *6*(1), 98-109.
(Cited on page 105.)

Liebchen, C., Proksch, M., & Wagner, F. (2008). Performance of algorithms for periodic timetable optimization, In Hickman M, Mirchandani P, Voss S (Eds.), Computer-Aided Systems in Public Transport (CASPT 2004). *Lecture Notes in Economics and Mathematical Systems*, *600*, 151–180.
(Cited on page 105.)

Liebchen, C., Schachtebeck, M., Schöbel, A., Stiller, S., & Prigge, A. (2010). Computing delay resistant railway timetables. *Computers & Operations Research*, *37*(5), 857–868.
(Cited on page 98.)

Lindfeldt, A. (2015). *Railway capacity analysis: Methods for simulation and evaluation of timetables, delays and infrastructure*. PhD Thesis, KTH Royal Institute of Technology.
(Cited on page 13.)

Lindner, T. (2011). Applicability of the analytical UIC code 406 compression method for evaluating line and station capacity. *Journal of Rail Transport Planning and Management*, *1*(1), 49–57.
(Cited on pages 4, 17, and 65.)

Löfberg, J. (2004). YALMIP: A toolbox for modeling and optimization in MATLAB. In *IEEE International Symposium on Computer Aided Control Systems Design 2004* (p. 284-289).

(Cited on page 115.)

Lucchini, L., Rivier, R., & Emery, D. (2000). CAPRES network capacity assessment for Swiss North-South rail freight traffic. *Computers in Railways VII*, 221-230. (Cited on page 23.)

Lukaszevicz, P. (2001). *Energy consumption and running time for trains.* PhD Thesis, Royal Institute of Technology, Stockholm. (Cited on page 172.)

Lusby, R., Larsen, J., Ehrgott, M., & Ryan, D. (2011). Railway track allocation: models and methods. *OR Spectrum*, *33*(4), 843-883. (Cited on page 30.)

Mackie, P., & Preston, J. (1998). Twenty-one sources of error and bias in transport project appraisal. *Transport Policy*, *5*(1), 1-7. (Cited on page 12.)

Maróti, G. (2016). A convex programming approach for robust railway timetabling. In *Proceedings of 9th Triennal Symposium on Transportation Analysis (TRISTAN IX).* Aruba, 13-17 June 2016. (Cited on pages 97, 98, 99, and 115.)

Medeossi, G., Longo, G., & de Fabris, S. (2011). A method for using stochastic blocking times to improve timetable planning. *Journal of Rail Transport Planning & Management*, *1*, 1-13. (Cited on pages 161, 162, and 168.)

Melody, K. S. (2012). *Railway track capacity: Measuring and managing*. PhD Thesis, University of Southampton. (Cited on page 17.)

Meng, L., & Zhou, X. (2014). Simultaneous train rerouting and rescheduling on an n-track network: A model reformulation with network-based cumulative flow variables. *Transportation Research Part B: Methodological*, *67*, 208-234. (Cited on page 30.)

Middelkoop, A. D. (2010). Headway generation with roberto. *WIT Transactions on the Built Environment*, *114*, 431-439. (Cited on page 128.)

Ministerie van Infrastructuur en Milieu. (2013). *Beter en meer: Concept operationele uitwerking van de lange termijn spooragenda.* Author. Retrieved from `https://www.rijksoverheid.nl/ministeries/ministerie-van-infrastructuur-en-milieu/documenten/rapporten/2014/02/11/beter-en-meer` (Cited on page 2.)

Ministerie van Verkeer en Waterstaat. (2010). *High-Frequency Rail Transport Programme - Priority Decision*. Retrieved from `https://www.rijksoverheid.nl/onderwerpen/spoor/documenten/brochures/2010/09/20/priority-decision-high-frequency-rail-transport-programme` (Cited on page 2.)

Mussone, L., & Calvo, R. W. (2013). An analytical approach to calculate the capacity

of a railway system. *European Journal of Operational Research*, *228*(1), 11-23.
(Cited on page 12.)

Nachtigall, K. (1993). *Exact solution methods for periodic programs*. Technical Report, 14/93 Hildesheimer Informatik-Berichte.
(Cited on page 95.)

Nachtigall, K. (1999). *Periodic network optimizationi and fixed interval timetables*.
(Cited on page 108.)

Nachtigall, K., & Opitz, J. (2008). Solving periodic timetable optimisation problems by modulo simplex calculations. In *OASIcs-OpenAccess Series in Informatics*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.
(Cited on pages 3 and 95.)

Nash, A., & Huerlimann, D. (2004). Railroad simulation using opentrack. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, & S. Sone (Eds.), *Computers in Railways IX* (p. 45-54). Southampton: WIT Press.
(Cited on page 65.)

Network Rail. (2015). *Timetable Planning Rules*. `www.networkrail.co.uk/aspx/3741.aspx`. (Accessed 15.10.2015)
(Cited on page 17.)

Niu, H., & Zhou, X. (2013). Optimizing urban rail timetable under time-dependent demand and oversaturated conditions. *Transportation Research Part C*, *36*, 212-230.
(Cited on page 37.)

NS. (2015). *Annual report 2015*. Technical Report.
(Cited on pages xv, 1, and 2.)

Odijk, M. A., Romeijn, H. E., & van Maaren, H. (2006). Generation of classes of robust periodic railway timetables. *Computers & Operations Research*, *33*(8), 2283-2299.
(Cited on page 12.)

Oettich, S., Albrecht, T., & Scholz, S. (2004). Improvements of energy efficiency of urban rapid rail systems. In C. A. Brebbia & L. C. Wadhwa (Eds.), *Urban transport and the environment* (pp. 573–582).
(Cited on page 48.)

Oliveira, E., & Smith, B. M. (2000). *A job-shop scheduling model for the single-track railway scheduling problem*. Technical Report, 2000.21, School of Computing Research Report, University of Leeds.
(Cited on page 43.)

ON-TIME. (2012). *Methods for capacity assessment in Europe*. Report ONT-WP03-I-UDB-009-03.
(Cited on pages 11 and 17.)

ON-TIME. (2013). *Assessment of state-of-art of train timetabling*. Report ONT-WP03-I-EPF-008-03.
(Cited on page 30.)

ON-TIME. (2014). *Benchmark analysis, test and integration of timetable tools*. Report

ONT-WP03-D-TUT-037-02.
(Cited on page 17.)

ON-TIME. (2016). Retrieved from www.ontime-project.eu (23.01.2016.)
(Cited on pages 6, 29, 66, and 128.)

Opitz, J. (2009). Untersuchungsgebiete–reale verkehrsbeispiele. In *Automatische Erzeugung und Optimierung von Taktfahrplänen in Schienenverkehrsnetzen* (pp. 35–44). Springer.
(Cited on page 3.)

Pachl, J. (2014). *Railway Operation and Control, Third Edition*. Mountlake Terrace: VTD rail publishing.
(Cited on page 14.)

Pätzold, J., & Schöbel, A. (2016). A matching approach for periodic timetabling. In *Oasics-openaccess series in informatics* (Vol. 54).
(Cited on page 95.)

Peeters, L. (2003). *Cyclic railway timetable optimization*. PhD thesis, Erasmus Universiteit Rotterdam, The Netherlands.
(Cited on pages 94, 95, 97, 99, 100, 105, 108, 111, and 113.)

Peng, F., Kang, S., Li, X., Ouyang, Y., Somani, K., & Acharya, D. (2011). A heuristic approach to the railroad track maintenance scheduling problem. *Computer-Aided Civil and Infrastructure Engineering*, *26*, 129-145.
(Cited on page 64.)

Petering, M. E., Heydar, M., & Bergmann, D. R. (2015). Mixed-integer programming for railway capacity analysis and cyclic, combined train timetabling and platforming. *Transportation Science*, *50*(3), 892–909.
(Cited on pages 96, 98, and 99.)

Planting, T. (2016). *Ontwerpmethoden van dienstregelingen (in Dutch)*.
(Cited on page 2.)

Pouryousef, H., Lautala, P., & White, T. (2015). Railroad capacity tools and methodologies in the us and europe. *Journal of Modern Transportation*, *23*(1), 30–42.
(Cited on pages 16 and 18.)

ProRail. (2008). *TROTS protocol – interface design description (in Dutch)*. Technical report, Utrecht.
(Cited on page 164.)

ProRail. (2016). *Network statement 2017*.
(Cited on page 2.)

Quaglietta, E. (2014). A simulation-based approach for the optimal design of signalling block layout in railway networks. *Simulation Modelling Practice and Theory*, *46*, 4–24.
(Cited on pages 18 and 65.)

Quaglietta, E., Pellegrini, P., Goverde, R. M. P., Albrecht, T., Jaekel, B., Marlire, G., ... Nicholson, G. (2016). The ON-TIME real-time railway traffic management framework: A proof-of-concept using a scalable standardised data communication architecture. *Transportation Research Part C: Emerging Technologies*, *63*,

23-50.
(Cited on page 50.)

Radosavljević, A. (2006). Measurement of train traction characteristics. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, *220*, 283-291.
(Cited on page 162.)

RailML. (2015). Retrieved from `www.railml.org` (last accessed 30.06.2015.)
(Cited on pages 68 and 130.)

RMCon. (2012). *Railsys 8 Enterprise, network wide timetable and infrastructure management*.
(Cited on pages 11 and 18.)

Robenek, T., Maknoon, Y., Azadeh, S. S., Chen, J., & Bierlaire, M. (2016). Passenger centric train timetabling problem. *Transportation Research Part B: Methodological*, *89*, 107–126.
(Cited on page 95.)

Rotoli, F., Navajas Cawood, E., & Soria, A. (2016). *Capacity assessment of railway infrastructure: Tools, methodologies and policy relevance in the EU context*. EUR 27835 EN, Joint Research Center, European Union.
(Cited on pages 14 and 17.)

SBB. (2016). *Network statement 2017*.
(Cited on page 2.)

Scheepmaker, G. M., & Goverde, R. M. P. (2015). The interplay between energy-efficient train control and scheduled running time supplements. *Journal of Rail Transport Planning & Management*, *5*(4), 225-239.
(Cited on pages 48 and 65.)

Schlechte, T., Borndörfer, R., Erol, B., Graffagnino, T., & Swarat, E. (2011). Micro-macro transformation of railway networks. *Journal of Rail Transport Planning and Management*, *1*(1), 38-48.
(Cited on pages 3, 4, 30, 36, 64, 66, 71, 128, and 134.)

Schmidt, M. (2014). *Integrating routing decisions in public transportation problems*. Springer.
(Cited on page 95.)

Schmidt, M., & Schöbel, A. (2015). Timetabling with passenger routing. *OR spectrum*, *37*(1), 75–97.
(Cited on page 95.)

Schmidt, M. J. (2014). *The effect of infrastructure changes on railway operations*. PhD Thesis, University of Louisville.
(Cited on page 13.)

Schrijver, A., & Steenbeek, A. (1994). *Dienstregelingontwikkeling voor Railned (Timetable construction for Railned)*. Technical report, Center for Mathematics and Computer Science. Amsterdam, The Netherlands.
(Cited on pages 94 and 95.)

Schwanhäußer, W. (1978). Die Ermittlung der Leistungsfähigkeit von großen

Fahrstraßenknoten und von Teilen des Eisenbahnnetzes. *AET*, *33*, 7–18.
(Cited on page 11.)

Schwanhäußer, W. (1994). The status of German railway operations management in research and practice. *Transportation Research Part A: Policy and Practice*, *28*(6), 495 - 500.
(Cited on page 11.)

Sels, P., Dewilde, T., Cattrysse, D., & Vansteenwegen, P. (2016). Reducing the passenger travel time in practice by the automated construction of a robust railway timetable. *Transportation Research Part B: Methodological*, *84*, 124-156.
(Cited on pages 64 and 98.)

Serafini, P., & Ukovich, W. (1989). A mathematical model for periodic event scheduling problems. *SIAM Journal on Discrete Mathematics*, *2*, 550–581.
(Cited on pages 43, 94, and 127.)

Shih, M.-C., Dick, C., Sogin, S., & Barkan, C. (2014). Comparison of capacity expansion strategies for single-track railway lines with sparse sidings. *Transportation Research Record: Journal of the Transportation Research Board*(2448), 53–61.
(Cited on page 13.)

Sicre, C., Cucala, P., & Fernandez, A. (2010). A method to optimize train energy consumption combining manual energy efficient driving and scheduling. *In: Computers in Railways XII,WIT Transactions on The Built Environment*, *114*, 549-560.
(Cited on page 49.)

Siebert, M., & Goerigk, M. (2013). An experimental comparison of periodic timetabling models. *Computers & Operations Research*, *40*(10), 2251-2259.
(Cited on pages 85 and 99.)

Siefer, T., & Radtke, A. (2006). Evaluation of delay propagation. In *the Proceedings of 7th World Congress on Railway Research (WCTR).* Montreal.
(Cited on page 65.)

Sogin, S., Lai, Y.-C., Dick, C. T., & Barkan, C. (2013). Comparison of capacity of single and double-track rail lines. *Transportation Research Record: Journal of the Transportation Research Board*, *2374*, 111–118.
(Cited on page 13.)

Sparing, D. (2016). *Reliable timetable design for railways and connecting public transport services*. PhD thesis, Delft University of Technology.
(Cited on page 96.)

Sparing, D., & Goverde, R. (2013). An optimization model for periodic timetable generation with dynamic frequencies. In *Proceedings of the 16th international IEEE annual conference on intelligent transportation systems (ITSC 2013)* (p. 785-790). IEEE.
(Cited on pages 96, 97, 99, and 104.)

Steenhuisen, B. (2009). *Competing public values: Coping strategies in heavily regulated utility industries*. PhD thesis, Delft University of Technology.
(Cited on page 186.)

Strategic Rail Authority. (2014). Capacity utilisation policy: Network utilisation strategy. *London, SRA*.
(Cited on page 13.)

Szpigel, B. (1973). Optimal train scheduling on a single track railway. In M. Ross (Ed.), *OR'72*. North-Holland, Amsterdam.
(Cited on page 43.)

TRB. (2013). *Transit Capacity and Quality of Service Manual. Third Edition. Transit Cooperative Research Program (TCRP) Report 165*. Transportation Research Board.
(Cited on page 13.)

Tsiflakos, K., & Owen, D. B. (1993). A decision support tool for the railway industry based on computer graphics and intuitive modelling techniques. *Computer-Aided Civil and Infrastructure Engineering*, *8*, 105-118.
(Cited on pages 63 and 69.)

UIC. (2004). *Code 406: Capacity, first edition*. Paris: International Union of Railways.
(Cited on pages 12, 13, and 16.)

UIC. (2013). *Code 406: Capacity, second edition*. Paris: International Union of Railways.
(Cited on pages 13, 16, 17, 32, 33, 34, 35, 42, 65, 66, and 132.)

UNECE. (2015). *Number of railway passengers by country, passengers and time.* `http://w3.unece.org/PXWeb2015/pxweb/en/STAT/STAT__40-TRTRANS__03-TRRAIL`. (Accessed: 30.09.2015)
(Cited on page 11.)

Van de Weg, G. S., Hegyi, A., & Hoogendoorn, S. P. (2014). Robust, optimal, predictive, and integrated road traffic control: Research proposal. (Cited on page 185.)

Weik, N., Niebel, N., & Nießen, N. (2016). Capacity analysis of railway lines in germany a rigorous discussion of the queueing based approach. *Journal of Rail Transport Planning & Management*, *6*(2), 99 - 115.
(Cited on page 12.)

Wendler, E. (2007). The scheduled waiting time on railway lines. *Transportation Research Part B: Methodological*, *41*(2), 148–158.
(Cited on page 12.)

Xie, W., Ouyang, Y., & Somani, K. (2016). Optimizing location and capacity for multiple types of locomotive maintenance shops. *Computer-Aided Civil and Infrastructure Engineering*, *31*(3), 163–175. doi: 10.1111/mice.12114
(Cited on page 64.)

Yuan, J., & Hansen, I. A. (2007). Optimizing capacity utilization of stations by estimating knock-on train delays. *Transportation Research Part B: Methodological*, *41*(2), 202 - 217.
(Cited on page 12.)

Zhang, X., & Nie, L. (2016). Integrating capacity analysis with high-speed railway

timetabling: A minimum cycle time calculation model with flexible overtaking constraints and intelligent enumeration. *Transportation Research Part C: Emerging Technologies*, *68*, 509-531.
(Cited on page 96.)

Summary

Mainline railways in Europe are experiencing more and more intensive use of their train services, particularly in urban areas, as the worldwide demand for passenger and freight transport is increasing across all transport modes. At the same time, much of the existing mainline railway network is reaching its capacity and has become susceptible to delays and disturbances. On one hand, a solution to the problem of saturated networks and growing demand would be to build more railway infrastructure; however, constructing new railways is expensive, takes considerable time and faces a number of environmental constraints. On the other hand, mathematical models and algorithms for capacity assessment and timetabling should be used to produce better timetable solutions and to speed up the planning process.

This thesis creates, optimizes, and evaluates railway timetables to promote more reliable, more attractive, and more sustainable railway transport systems. We develop an integrated approach for improved railway timetabling that combines capacity assessment and scheduling models in order to design timetables that are *efficient*, i.e., have as short as possible journey times, *feasible*, i.e., all trains operate undisturbed by other traffic, *stable*, i.e., do not have excessive infrastructure capacity occupation, and *robust*, i.e., are able to mitigate certain everyday operational disturbances.

We first reviewed methods for railway capacity assessment and described the existing models for assessing the whole networks as well as corridors. Also, a new max-plus model for capacity occupation assessment was defined. The model follows the compression method indicated by the UIC 406. A capacity occupation rate is considered as a stability measure, where the remaining available capacity indicates whether delays could recover in limited time. If the capacity occupation satisfies predefined thresholds the timetable is considered to be stable.

Second, we proposed a conceptual three-level performance-based timetabling framework that integrates timetable construction and evaluation. The advantage of this approach is that performance indicators, such as efficiency, feasibility, stability and robustness, are already taken into account during the timetable construction. This alleviates the time-consuming task of ex-ante simulations to test the constructed timetable on, for example, conflicts, stability and robustness.

Third, microscopic models were developed to compute reliable running and minimum headway times for a macroscopic timetabling model, as well as check the mi-

croscopic feasibility and stability of the macroscopic timetables. Train running times are computed by integrating the Newton's motion formulae, while the accurate headway computation is based on the blocking time theory. The feasibility of the timetable is checked by an efficient conflict detection and resolution model based on blocking time theory, and in case of conflicts, new running and minimum headway times are automatically computed.

Fourth, a two-stage stability-to-robustness model is proposed, which is the first timetable optimization model that incorporates three important performance indicators of timetable design: efficiency, stability and robustness. The first stage focuses on stability, the second stage on robustness, while efficiency is considered in both stages. Five objective functions were defined to generate alternative timetables. Results showed that the approach is capable of computing both stable and robust solutions, which are in most cases better than when applying existing timetabling models. It also showed a computational dominance over existing models.

Fifth, we integrated microscopic models with a macroscopic timetable model into a performance-based micro-macro timetabling framework. The resulting timetable is computed together with all performance measures that are either satisfied or optimized depending on the required criteria. This unprecedented integrated approach that guarantees feasible, efficient, stable and robust solutions has been made possible by developed microscopic and macroscopic timetable models, and also by efficient and consistent data transformations between the various levels. This enables an effective framework in which microscopic details can be combined with macroscopic optimization over large networks, including stochastic models for robustness evaluation. The application of the proposed framework showed that high quality timetables were produced in a limited number of iterations, while the capacity occupation rate was always below an acceptable level.

Sixth, we developed a simulation-based optimization model to derive the most probable speed profiles of train runs from traffic realization data. The train behaviour, which includes driving parameters like tractive effort, motion resistances, braking effort, and cruising, was calibrated using a genetic algorithm by minimizing the error between realized and simulated running times. Results showed differences in provided rolling stock parameters and the actual driver behaviour. This gained info can be used in planning and operations by tweaking inputs to the timetabling framework.

In summary, this thesis demonstrates that optimization, simulation and data analysis can be successfully applied to improving railway planning and account for better infrastructure capacity use and increased level of service.

Samenvatting

Het spoor in Europa wordt steeds intensiever benut, vooral in stedelijke gebieden. Dit komt doordat de vraag naar reizigers- en goederenvervoer wereldwijd toeneemt. Tegelijkertijd bereikt een groot deel van de spoorinfrastructuur haar capaciteit, waardoor het netwerk kwetsbaar is voor verstoringen en vertragingen. Een mogelijkheid om dit op te lossen is om meer spoorinfrastructuur aan te leggen. Het nadeel hiervan is dat dit veel tijd en geld kost en er veel ruimtelijke beperkingen zijn. Een andere mogelijkheid is het gebruik van wiskundige modellen en algoritmes voor capaciteitsbenutting en treindienstregelingsontwerp, waarmee betere dienstregelingen bepaald kunnen worden en het planproces versneld kan worden.

Dit proefschrift ontwikkelt, optimaliseert en evalueert treindienstregelingen en bevordert een betrouwbaarder, aantrekkelijker en duurzamer spoorvervoer. Een integrale methodiek is ontwikkeld om een dienstregeling te verbeteren. Deze methodiek combineert capaciteitsbenutting en planningsmodellen om een dienstregeling te ontwikkelen die efficiënt (zo kort mogelijke rijtijden), haalbaar (conflictvrije treindiensten), stabiel (voldoende speling in de capaciteitsbenutting) en robuust (verminderen van bepaalde dagelijkse realistische verstoringen) is.

Ten eerste zijn de methodieken voor capaciteitsbenutting beoordeeld en zijn de bestaande modellen voor het bepalen van deze benutting op baanvak en netwerk niveau besproken. Daarnaast is een nieuw max-plus model voor capaciteitsbenutting ontwikkeld. Dit model volgt de compressiemethodiek die beschreven staat in de UIC 406 standaard. Als stabiliteitsmaat wordt capaciteitsbenutting gehanteerd, waarbij de resterende beschikbare capaciteit een indicatie geeft hoeveel vertragingen opgevangen kan worden binnen redelijke tijd. Als de capaciteitsbenutting voldoet aan de voorgedefinieerde grenswaarden dan, is de dienstregeling stabiel.

Ten tweede is een conceptueel drie-fase model voor het dienstregelingsontwerpproces voorgesteld waarin dienstregelingsontwerp en evaluatie geïntegreerd zijn. Een voordeel van deze aanpak is dat er tijdens het ontwerpproces rekening wordt gehouden met de prestatie-indicatoren als efficiëntie, haalbaarheid, stabiliteit en robuustheid. Hierdoor zijn geen tijdrovende simulaties nodig om een ontwikkelde dienstregeling te toetsen op bijvoorbeeld conflicten, stabiliteit en robuustheid.

Ten derde zijn er microscopische modellen voor de berekening van betrouwbare rijtijden en minimale opvolgtijden als invoer voor macroscopische dienstregelingsmod-

ellen. Daarnaast is er microscopisch modellen ontworpen die de haalbaarheid en stabiliteit van macroscopische dienstregelingen toetst. De rijtijden worden berekend door integratie van de bewegingsvergelijkingen van Newton en de opvolgtijden worden berekend met behulp van de bloktijdtheorie. De haalbaarheid van de dienstregeling wordt getoetst door een efficiënt conflictdetectie- en oplossingsmodel gebaseerd op de bloktijdtheorie. Indien er conflicten zijn, worden er automatisch nieuwe rijtijden en minimale opvolgtijden berekend.

Ten vierde is een twee-fase model gepresenteerd die de stabiliteit en robuustheid van de dienstregeling optimaliseerd. Dit is het eerste dienstregelingsoptimalisatie model dat rekening houdt met de belangrijkste prestatie-indicatoren voor dienstregelingsontwerp, te weten: efficiëntie, stabiliteit en robuustheid. De eerste fase richt zich op stabiliteit, terwijl de tweede fase gericht is op robuustheid. Bovendien wordt in beide fases rekening gehouden met de efficiëntie van de dienstregeling. Vijf doelfuncties zijn opgesteld om alternatieve dienstregelingen te kunnen genereren. De resultaten laten zien dat de methodiek in staat is om stabiele en robuuste dienstregelingen te berekenen, die veelal beter zijn dan de bestaande dienstregelingsmodellen. Bovendien is de rekensnelheid van het model sneller dan de bestaande modellen.

Ten vijfde is het microscopische model met het macroscopische model geïntegreerd in een prestatie-gericht micro-macro dienstregelingsraamwerk. De ontworpen dienstregeling wordt opgesteld aan de hand van de prestatie-indicatoren, waaraan voldaan dient te worden of die geoptimaliseerd worden, wat afhankelijk is van de vereiste ontwerpcriteria. Deze geïntegreerde macro-micro ontwerpmethodiek zorgt ervoor dat de resulterende dienstregeling haalbaar, efficiënt, stabiel en robuust is, , waarbij gebruik wordt gemaakt van consistente data transformaties tussen de verschillende niveaus. Dit zorgt voor een efficiënt raamwerk waarin microscopische details worden gecombineerde met macroscopische optimalisatie, inclusief stochastische modellen voor robuustheidevaluatie. De toepassing van de methodiek toont aan dat hoge kwaliteit dienstregelingen worden berekend in een beperkt aantal iteratieslagen, waarbij de capaciteitsbenutting beneden een acceptabel niveau blijft.

Ten zesde is een simulatie gebaseerd optimalisatiemodel ontwikkeld dat op basis van realisatiedata de meest waarschijnlijke snelheidsprofielen van de gerealiseerde treinritten geneert. Het treingedrag (zoals trekkracht, treinweerstand en remkracht) is geijkt met behulp van een genetisch algoritme dat de fout tussen de rijtijden van de simulatie en realisatie minimaliseert. De resultaten laten een verschil zien tussen de gegeven materieel parameters en het echte rijgedrag. Dit kan dienen als verbeterde input voor de ontworpen methodiek om tot een betere planning en uitvoering te komen.

Samengevat laat dit proefschrift zien dat optimalisatie, simulatie en data-analyse succesvol toegepast kunnen worden om de planning van dienstregelingen te verbeteren, met als gevolg een betere capaciteitsbenutting van de infrastructuur en serviceniveau voor reizigers.

About the author

Nikola Bešinović was born in Zaječar, Serbia in 1985. After graduating from the Gimnasium in Bor, Serbia, he moved to Belgrade and obtained his BSc in Railway Traffic and Transport Engineering at the Faculty of Traffic and Transport Engineering, University of Belgrade in 2009. Consequently, he received the MSc in Operations Research in Transport from the University of Belgrade, Serbia in 2011 with a project on a location problem in transportation networks. After graduation, Nikola worked as a Research Assistant at the same Faculty.

In June 2012, Nikola started his PhD research on developing models and algorithms for improved railway timetabling at the Department of Transport and Planning, Delft University of Technology. This PhD project was part of the European FP7 project Optimal Networks for Train Integration Management Across Europe (ON-TIME). His main focus is on advanced models and algorithms for improving of railway and public transport timetable planning and traffic management, evaluating operations performances and appraisal of infrastructure projects. During his PhD research, Nikola was a visiting researcher at the British infrastructure manager Network Rail and a visiting lecturer at Beijing Jiaotong University, China.

After completing his PhD thesis in March 2017, he started as a post-doctoral researcher at the Department of Transport and Planning, Delft University of Technology. Currently, Nikola is also working on maintenance scheduling and has a rising interest in resilient public transport planning and operations as well as ad-hoc freight scheduling.

Curriculum Vitæ

Nikola Bešinović
n.besinovic@tudelft.nl

Education

2012–2017	Delft University of Technology, The Netherlands	
	PhD in Transport	
	Thesis:	Integrated capacity assessment and timetabling models for dense railway networks
	Promotors:	Prof. Dr. Serge Hoogendoorn and Dr. Rob Goverde
2009–2011	University of Belgrade, Serbia	
	MSc in Operations Research in Transport	
	Thesis:	Locating weigh-in-motion sensors in transportation networks (GPA: 10/10)
	Promotor:	Prof. Dr. Dušan Teodorović
2004–2009	University of Belgrade, Serbia	
	BSc in Railway Transport and Traffic Engineering	
	Thesis:	Simulation model for computing capacity consumption: Case study on railway line Novi Beograd Batajnica (GPA: 8.7/10)
	Promotor:	Prof. Dr. Slavko Vesković

Experience

2017-(now)	Post-doctoral researcher	Delft, NL
	Delft University of Technology	
2016	Guest researcher	London, UK
	Network Rail,	
2012–2014	Lead algorithm and model developer	Delft, NL
	Project Optimal Networks for Train Integration Management Across Europe (ON-TIME)	
2011–2012	Research Assistant	Belgrade, Serbia
	Faculty of Transport and Traffic Engineering, University of Belgrade	

Teaching activities

2016	Co-lecturer, Course Advanced Methods for Railway Timetabling and Capacity Assessment at Beijing Jiaotong University	Beijing, China
2013–(now)	Teaching Assistant, Course Railway Traffic Management at Delft University of Technology,	Delft, NL
2013–(now)	Supervising MSc students	Delft, NL

Conference activities

- Organizing a railway stream at EURO Working Group on Transportation (EWGT) 2015 held in Delft
- Chairing conference sessions: CASPT, TRISTAN, INFORMS, EWGT
- Referee for INFORMS RAS Problem Solving Competition 2016

Invited talks

Nov 2015	TransitLab seminar, Massachusetts Institute of Technology	Boston, USA
Nov 2015	Massachusetts Bay Transportation Authority (MBTA)	Boston, USA
Dec 2014	Railway Operations Research Seminar	Leuven, Belgium
May 2014	Workshop Capacity Research in Urban Railbound Transportation with Special Consideration of Mixed Traffic	Stuttgart, Germany
Nov 2013	Plenary at the 3rd Conference on Advanced Railway Technologies	Warsaw, Poland
Sept 2013	Plenary at the 6th Conference on Logistics in Academy and Industry	Lidzbark Varminski, Poland

Awards

- Young Railway Operations Research Award 2017 with the paper 'Resolving instability in railway timetabling problems' at the 7th International seminar on railway operations modelling and analysis (RailLille 2017)
- The 3rd Best paper titled 'Resolving instability in railway timetabling problems' at the 7th International seminar on railway operations modelling and analysis (RailLille 2017)

- The 8th Best paper titled 'Solving large-scale timetable adjustment problem under infrastructure maintenance possessions' at the 7th International seminar on railway operations modelling and analysis (RailLille 2017)
- Grant for EURO Winter Institute on Methods and models in transportation problems in 2017
- Erasmus+ Mobility Grant in 2016
- Best student paper award of INFORMS Railway Applications Section with the paper 'A novel two-stage approach for robust timetabling' at the 2015 INFORMS Annual Meeting
- Top 10 Best papers with the paper 'Micro-macro approach for robust railway timetabling' at the 6th International seminar on railway operations modelling and analysis (RailTokyo 2015)
- Nominated (Top 5) for Young Railway Operations Research Award with the paper 'A simulation-based optimization approach for the calibration of dynamic train speed profiles' at the 5th International seminar on railway operations modelling and analysis (RailCopenhagen 2013)

Publications

Journal articles

1. Van Aken, S., Bešinović, N., Goverde, R.M.P. (2017). Designing alternative railway timetables under infrastructure maintenance possessions, *Transportation Research Part B: Methodological*, 98, 224–238.
2. Bešinović, N., Goverde, R.M.P., Quaglietta, E. (2017). Microscopic models and network transformations for automated railway traffic planning. *Computer-Aided Civil and Infrastructure Engineering*, 32 (2), 89–106.
3. Bešinović, N., Goverde, R.M.P., Quaglietta, E., Roberti, R. (2016). An integrated micromacro approach to robust railway timetabling. *Transportation Research Part B: Methodological*, 87, 14–32.
4. Goverde, R.M.P., Bešinović, N., Binder, A., Cacchiani, V., Quaglietta, E., Roberti, R., Toth, P. (2016). A three-level framework for performance-based railway timetabling. *Transportation Research Part C: Emerging Technologies*, 67, 62–83.
5. Bešinović, N, Quaglietta, E., Goverde, R.M.P., (2013) A simulation-based optimization approach for the calibration of dynamic train speed profiles. *Journal of Rail Transport Planning & Management*, 3(4), 126–136.

Book chapter

1. Bešinović, N., Goverde, R.M.P., Capacity assessment in railway networks, In Borndörfer, R., Klug, T., Lamorgese, L., Mannino, C., Reuther, M., Schlechte, T. (Eds.), *Handbook on Operations Research in Railway Industry*, Springer, accepted.

Working articles

1. Bešinović, N., Goverde, R.M.P., A two-stage stability-to-robustness approach for robust railway timetabling, submitted.

2. Bešinović, N., Goverde, R.M.P., Robust train routing in station areas with reduced infrastructure capacity occupation, submitted.

3. Bešinović, N., Quaglietta, E., & Goverde, R.M.P., Resolving instability in railway timetabling problems, submitted.

4. Van Aken, S., Bešinović, N., Goverde, R.M.P., Solving large-scale timetable adjustment problem under infrastructure maintenance possessions, submitted.

Other relevant publications and reports

1. Bešinović, N., Goverde R.M.P., (2016). A novel approach for automated timetable planning, The *11th World Congress on Railway Research* (*WCRR 2016*), Milan, Italy, 29 May-1 June 2016.

2. Bešinović, N., (2016). Railway science: Toward reliable railway timetabling. *ExtraCT*, 18(4), January 2016.

3. Yan, F., Goverde, R.M.P., Bešinović, N., (2015). Line planning problem in a dense high-speed rail corridor, In *Conference on Advanced Systems in Public Transport* (*CASPT 2015*), Rotterdam, The Netherlands, 19-23 July 2015.

4. Bešinović, N., Cacchiani, V., Dollevoet, T., Goverde, R.M.P., Huisman, D., Kidd, M.P., Kroon, L.G., Quaglietta, E., Rodriguez, J., Toth, P., (2015). Integrated Decision Support Tools for Disruption Management, In *6th International conference on Railway Operations Modelling and Analysis* (*RailTokyo2015*), Narashino, Japan, 23-26 March 2015.

5. ON-TIME. (2014). *Benchmark analysis, test and integration of timetable tools*. Report ONT-WP03-D-TUT-037-02.

6. ON-TIME. (2014). *Methods and algorithms for the development of robust and resilient timetables*. Report ONT-WP03-D-TUT-034-01.

7. ON-TIME. (2013). *Functional design of robust and resilient timetable models*. Report ONT-WP03-I-UDB-010-03.

8. Bešinović, N., Quaglietta, E., Goverde, R.M.P., (2013) Calibrating and validating train dynamics characteristics against realisation data, In *Proceedings of the Second Methods and Technologies of Intelligent Transportation Systems* (*MT-ITS 2013*), Dresden, Germany, 2-4 December 2013.

TRAIL Thesis Series

The following list contains the most recent dissertations in the TRAIL Thesis Series. For a complete overview of more than 150 titles see the TRAIL website: www.rsTRAIL.nl. The TRAIL Thesis Series is a series of the Netherlands TRAIL Research School on transport, infrastructure and logistics.

Bešinović, N., *Integrated Capacity Assessment and Timetabling Models for Dense Railway Networks*, T2017/9, July 2017, TRAIL Thesis Series, the Netherlands

Chen, G., S*urface Wear Reduction of Bulk Solids Handling Equipment Using Bionic Design*, T2017/8, June 2017, TRAIL Thesis Series, the Netherlands

Kurapati, S., *Situation Awareness for Socio Technical Systems: A simulation gaming study in intermodal transport operations*, T2017/7, June 2017, TRAIL Thesis Series, the Netherlands

Jamshidnejad, A., *Efficient Predictive Model-Based and Fuzzy Control for Green Urban Mobility*, T2017/6, June 2017, TRAIL Thesis Series, the Netherlands

Araghi, Y., *Consumer Heterogeneity, Transport and the Environment*, T2017/5, May 2017, TRAIL Thesis Series, the Netherlands

Kasraian Moghaddam, D., *Transport Networks, Land Use and Travel Behaviour: A long term investigation*, T2017/4, May 2017, TRAIL Thesis Series, the Netherlands

Smits, E.-S., *Strategic Network Modelling for Passenger Transport Pricing*, T2017/3, May 2017, TRAIL Thesis Series, the Netherlands

Tasseron, G., *Bottom-Up Information Provision in Urban Parking: An in-depth analysis of impacts on parking dynamics*, T2017/2, March 2017, TRAIL Thesis Series, the Netherlands

Halim, R.A., *Strategic Modeling of Global Container Transport Networks: Exploring the future of port-hinterland and maritime container transport networks*, T2017/1, March 2017, TRAIL Thesis Series, the Netherlands

Olde Keizer, M.C.A., *Condition-Based Maintenance for Complex Systems: Coordinating maintenance and logistics planning for the process industries*, T2016/26, December 2016, TRAIL Thesis Series, the Netherlands

Zheng, H., *Coordination of Waterborn AGVs*, T2016/25, December 2016, TRAIL Thesis Series, the Netherlands

Yuan, K., *Capacity Drop on Freeways: Traffic dynamics, theory and Modeling*, T2016/24, December 2016, TRAIL Thesis Series, the Netherlands

Li, S., *Coordinated Planning of Inland Vessels for Large Seaports*, T2016/23, December 2016, TRAIL Thesis Series, the Netherlands

Berg, M. van den, *The Influence of Herding on Departure Choice in Case of Evacuation: Design and analysis of a serious gaming experimental set-up*, T2016/22, December 2016, TRAIL Thesis Series, the Netherlands

Luo, R., *Multi-Agent Control of urban Transportation Networks and of Hybrid Systems with Limited Information Sharing*, T2016/21, November 2016, TRAIL Thesis Series, the Netherlands

Campanella, M., *Microscopic Modelling of Walking Behavior*, T2016/20, November 2016, TRAIL Thesis Series, the Netherlands

Horst, M. van der, *Coordination in Hinterland Chains: An institutional analysis of port-related transport*, T2016/19, November 2016, TRAIL Thesis Series, the Netherlands

Beukenkamp, W., *Securing Safety: Resilience time as a hidden critical factor*, T2016/18, October 2016, TRAIL Thesis Series, the Netherlands

Mingardo, G., *Articles on Parking Policy*, T2016/17, October 2016, TRAIL Thesis Series, the Netherlands

Duives, D.C., *Analysis and Modelling of Pedestrian Movement Dynamics at Large-scale Events*, T2016/16, October 2016, TRAIL Thesis Series, the Netherlands

Wan Ahmad, W.N.K., *Contextual Factors of Sustainable Supply Chain Management Practices in the Oil and Gas Industry*, T2016/15, September 2016, TRAIL Thesis Series, the Netherlands

Liu, X., *Prediction of Belt Conveyor Idler Performance*, T2016/14, September 2016, TRAIL Thesis Series, the Netherlands

Gaast, J.P. van der, *Stochastic Models for Order Picking Systems*, T2016/13, September 2016, TRAIL Thesis Series, the Netherlands

Wagenaar, J.C., *Practice Oriented Algorithmic Disruption Management in Passenger Railways*, T2016/12, September 2016, TRAIL Thesis Series, the Netherlands

Psarra, I., *A Bounded Rationality Model of Short and Long-Term Dynamics of Activity-Travel Behavior*, T2016/11, June 2016, TRAIL Thesis Series, the Netherlands

Ma, Y., *The Use of Advanced Transportation Monitoring Data for Official Statistics*, T2016/10, June 2016, TRAIL Thesis Series, the Netherlands