

Improving Real-Time Train Dispatching: Models, Algorithms and Applications

Andrea D'Ariano

This publication has been supported by the research programs “Towards Reliable Mobility” of the Transport Research Centre Delft, the Dutch foundation “Next Generation Infrastructures”, the Dutch infrastructure manager ProRail and TRAIL Research School.



Cover illustration: Railway traffic control center of Pisa (www.rfi.it)

Improving Real-Time Train Dispatching: Models, Algorithms and Applications

Proefschrift

ter verkrijging van de graad van doctor

aan de Technische Universiteit Delft,

op gezag van de Rector Magnificus Prof.dr.ir. J.T. Fokkema,

voorzitter van het College voor Promoties,

in het openbaar te verdedigen op donderdag 7 april 2008 om 10:00 uur

door

Andrea D'Ariano

Computer Science and Automation Engineer

Roma Tre University

geboren te Rome, Italië

Dit proefschrift is goedgekeurd door de promotor:

Prof.Dr.-Ing. I.A. Hansen

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof.Dr.-Ing. I.A. Hansen,	Technische Universiteit Delft, promotor
Prof.dr.ir. P.H.L. Bovy,	Technische Universiteit Delft
Prof.dr.ir. B. De Schutter,	Technische Universiteit Delft
Prof.dr. L.G. Kroon,	Erasmus Universiteit Rotterdam
Prof.dr. M. Carey	Queen's University Belfast, Northern Ireland
Prof.Dr.-Ing. E. Wendler,	RWTH Aachen University, Germany
Prof.dr. D. Pacciarelli,	Roma Tre University, Italy

This thesis is the result of a Ph.D. study carried out from April 2004 to April 2008 at the Department of Transport & Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology.

TRAIL Thesis Series no. T2008/6, The Netherlands TRAIL Research School

TRAIL
P.O. Box 5017
2600 GA Delft
The Netherlands
Phone: +31 (0) 15 27 86046
Fax: +31 (0) 15 27 84333
E-mail: info@rsTRAIL.nl

ISBN-number: 978-90-5584-100-4

Keywords: Real-time Rail Traffic Management, Decision Support System, Alternative Graph, Scheduling, Routing, Blocking Time Theory, Train Speed Coordination

Copyright © 2008 by Andrea D'Ariano

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage and retrieval system, without written permission from the author.

Printed in The Netherlands

Preface

My interests in railway operations research started with the European project “COMBINE2 - enhanced CONTROL center for fixed and Moving Block signaling system 2”. I got this chance from Professor Dario Pacciarelli, when I was still a Master student at Roma Tre University. He heard about my strong curiosity on doing a PhD abroad and proposed me to stay in the Netherlands for a master thesis project on real-time railway traffic optimization. I therefore spent three months as a visiting student at ProRail, department of Strategies and Innovation, in order to get familiar with railway operations and traffic management problems, and to collect and elaborate data for my master thesis. This was mainly possible thanks to the effort done as mentor by the ProRail manager Robin Hemelrijk.

Thanks to Dario and Robin, I had the opportunity to be introduced to Professor Ingo A. Hansen at Delft University of Technology. Well, I loved the experience at ProRail but I got really impressed by the number of researchers working at this university by just walking in the corridors of the faculty of Civil Engineering. It was suddenly clear that moving to Delft would have opened a big door into the world of top level international scientific research.

Since the first meeting with Ingo, I had the feeling that he was curious about the potential use of the novel methodology based on a detailed mathematical formulation of railway traffic management problems. From that day on, his interests and trust in my research findings increased gradually, always guided by a constructive criticism. He has been my supervisor during the entire working period at Delft University of Technology, first as a young research assistant and then as a PhD student.

This PhD thesis is mainly the result of a fruitful cooperative research between Delft University of Technology, Roma Tre University and ProRail. The general purpose was to develop a decision support tool for traffic controllers. To this aim, I spent a lot of energy in order to combine the knowledge and expertise of the parties involved in this project.

In Delft, I built up my knowledge and experience in traffic engineering and railway operations and systems. A special thank clearly goes to Professor Ingo A. Hansen and his research group, especially to Rob Goverde and Jianxin Yuan. I also wish to express my gratefulness to the visiting researchers Professor Geoff Rose, Professor Lie Nie, Thomas Albrecht, Marco Luthi and Stefan Wegele.

During my numerous visits to Roma Tre University, I extended my knowledge in computer science and explored the fascinating world of operations research. A special thank goes to Professor Dario Pacciarelli and his research group, especially to Marco Pranzo.

On a regular basis, I had meetings with ProRail and NS managers in order to direct my research to address problems of actual practical interest. Their work has been really precious also when dealing with the data collection and elaboration for the test cases studied in this thesis. A special thank goes to Robin Hemelrijk, Leo Lodder, Dick Middelkoop, Tijs Huisman, Alfons Schaafsma, Michiel Deerenberg, Lesley Valies among others.

I am grateful to the members of my evaluation committee, as I had the chance to improve my thesis with their helpful comments and suggestions.

Of course, a big thank is for my kind colleagues, technical staff and secretaries of the department of Transport and Planning during these years, especially to Ciccio, Minwei, Mario, Francesco, Stella, Hans and Winnie. Another one is for my friends from other departments, faculties and universities collaborating within the TRAIL research school and from other universities and institutions.

In general, during my living in the Netherlands I had the pleasure to get in touch with a lot of interesting people from all around the world. Since it is hard to cite them one by one, I wish to thank them in the so-called groups of friends: “Koornmarkt46”, “Buitenlanders”, “Zuiderpark” and “SanMarco”.

Grazie anche a tutti i miei amici italiani che mi sono venuti a trovare in Olanda o che ho comunque il piacere di incontrare ogni volta che torno in Italia. Ma il ringraziamento più importante va alla mia famiglia, in particolare mia madre, mio padre, mio fratello, mia nonna, mio nonno, zio Alessandro, Stefania, zia Marina, zio Angelo, “sor” Alessio, Francesca, zio Massimo, zia Barbara e Gioia. Grazie di cuore!

Andrea D’Ariano

Contents

Preface	v
1 Introduction	1
1.1 Current railway operations practice	1
1.2 Research motivation	3
1.3 Research objectives	5
1.4 Research relevance	6
1.5 Thesis contributions	7
1.6 Thesis outline	11
2 An overview on rail traffic management	13
2.1 Off-line timetabling	14
2.1.1 Train routing models	18
2.1.2 Train scheduling models	21
2.1.3 Train routing and scheduling models	27
2.1.4 Stochastic models	31
2.2 Real-time traffic management	35
2.2.1 Fixed-speed models	39
2.2.2 Variable-speed models	44
2.3 Dynamic traffic management	49
2.4 Short discussion	51

3	Real-time train dispatching approach	53
3.1	Problem description	54
3.2	System architecture	57
3.2.1	Load information	59
3.2.2	Disruption recovery	61
3.2.3	Real-time traffic optimization	61
3.2.4	Train speed coordination	64
3.3	Conclusions	65
4	Fixed-speed traffic optimization model	67
4.1	Alternative graph model	69
4.2	CDRFR problem formulation	70
4.2.1	Running and setup time constraints	71
4.2.2	CDRFR solution description	72
4.2.3	Scheduled stop formulation	73
4.2.4	CDRFR illustrative example	74
4.2.5	Additional railway constraints	76
4.3	CDR problem formulation	78
4.3.1	CDR illustrative example	78
4.4	Conclusions	80
5	Train scheduling algorithms	81
5.1	Static and dynamic implications	82
5.2	Lower bound algorithm	85
5.3	Dispatching heuristics	86
5.4	Greedy heuristics	87
5.5	Branch and bound algorithm	88
5.6	Illustrative example of optimal CDRFR solutions	91
5.7	Computational experiments	94
5.7.1	Evaluation of CDRFR algorithms	95
5.7.2	Assessment of flexible timetables	101
5.8	Conclusions	105

6	Train rerouting algorithms	107
6.1	Properties of the CDR problem	108
6.2	A local search algorithm	110
6.3	A tabu search algorithm	112
6.3.1	Upper and lower bounds to the CDRFR problem	113
6.3.2	Routing neighborhoods	113
6.3.3	The tabu search scheme	117
6.4	Computational experiments	119
6.4.1	Local search algorithm	120
6.4.2	Tabu search algorithm	126
6.5	Conclusions	131
7	Train speed coordination	133
7.1	Feasibility check	134
7.2	Speed updating	137
7.3	Modeling the adjustment of train speed profiles	138
7.4	Train speed coordination procedure	139
7.4.1	TSC illustrative example	142
7.5	Improvement of speed profiles at conflict points	146
7.6	Computational experiments	148
7.6.1	Applications of the train speed coordination procedure	149
7.6.2	Effects of optimal train speed profiles	155
7.7	Conclusions	158
8	Short-term traffic prediction	159
8.1	Temporal decomposition approach	160
8.1.1	STTP illustrative example	162
8.2	Computational experiments	164
8.2.1	Configurations analysis	166
8.2.2	Effects of timetable disturbances	168
8.3	Conclusions	171

9 Conclusions	173
9.1 Resume of the main achievements	173
9.2 Recommendations for future research	175
Bibliography	177
A Test cases description	195
B Glossary	199
Index	205
Summary	209
Samenvatting	215
About the author	221
TRAIL Thesis Series	223
NGInfra Theses	229

Chapter 1

Introduction

1.1 Current railway operations practice

A railway system is a complex system with many interacting processes that depend on technical devices, human behavior, and the external environment, and therefore contains many risks of disturbances. Once a delayed train deviates from its original time-distance path, it may hinder subsequent trains that are scheduled over the same railway infrastructure or it may conflict in passings or meetings with other trains. Hence, a delayed train may propagate its delay to other trains due to infrastructure, signaling or timing conflicts.

In the Netherlands, the railway network is heavily used with heterogeneous train traffic and short headway times (i.e., time intervals between successive trains). Every day approximately 5,000 passenger and 200 freight trains use the network. In 2003, all the trains jointly traveled more than 137 million kilometers. Moreover, the continuous growth of frequency of passenger and freight railway traffic (e.g. Dutch railway realized 10% passenger growth over the last two years) is increasing the pressure on European railway companies, facing the challenge of accommodating the expected growth of transport demand while improving train punctuality. As reported in Goverde (2005), in the autumn of 2001 the punctuality of the Dutch railway system was about 80% (delay < 3 min). The government demands a future punctuality level of at least 87%. On top of this, transport demand is expected to grow strongly. Possibilities for large infrastructure investments are limited and therefore a more efficient use of railway infrastructure is necessary.

The usual method how railways manage their traffic performance is through a carefully designed plan of operations, defining several months in advance routes, orders and timing for all trains. This process, called *off-line timetabling*, is followed by real-time measures to manage disturbances. Designing a timetable is a long-term and complex procedure, and sophisticated decision support tools, based on mathematical programming techniques (see, e.g. the Dutch timetable design system DONS (Designer Of Network Schedules) (Hooghiemstra et al., 1999)), as well as macroscopic simulation tools (see, e.g. the Dutch tool SIMONE (Simulation MOdel of NETworks) (Middelkoop & Bouwman, 2001)), are

applied to optimize the use of infrastructure capacity and to distribute suitable running time supplements and time margins that can absorb minor delays occurring in practice.

The tactical planning process of timetabling already starts a year before a new annual timetable becomes operational. During that year the Dutch infrastructure manager (ProRail) allocates the available infrastructure capacity to the various train operators and resolves conflicting train path requests by means of a planning tool (VPT-planning) linked to a database of infrastructure and rolling stock characteristics. The timetable prescribes the infrastructure allocation in time and space for all regular passenger trains including reservations for (freight) train paths for short-term capacity requests which may be allocated a few days before operation. The train paths are designed at a precision of minutes and contain the routes through stations, platform track usage, and train sequences at open tracks. Rolling stock circulations and crew schedules are worked out in advance according to the timetable. The extensive tactical planning process cumulates into integrated feasible day plans containing the infrastructure allocation of the national Dutch railway network on each day and, in case of the passenger train operator NS Reizigers (NSR), also the definitive rolling stock circulations and crew schedules. A day before operations the day plan for the next day is distributed from the planning system to the traffic control system (VPT-VKL) and the dispatching systems (VPT-PRL). Detailed descriptions of VPT functionalities (the automated system whereby signals and point switches are operated) can be found e.g. in Goverde (2005), Goverde and D'Ariano (2005) and Daamen et al. (2007).

Operational traffic management is currently mainly directed towards maintaining the tactical day plan and recovering from disruptive events as quickly as possible back to the original timetable. Traffic control is responsible for the train traffic operations in a wider control area. In agreement with the concerned transport operator a traffic controller may reschedule delayed trains from the next station onwards. Network traffic managers are responsible for general supervision of their part of the network and communications with neighboring control centers. Dispatchers supervise all train traffic in their control area based on real-time monitoring of track occupation and clearance, as well as train-number records. They are responsible for route setting within the interlocking limits of the station, supported by automatic train regulation integrated in VPT-PRL. The actual Dutch automatic route setting is usually done by the ARI system (i.e., Automatische Rijweg Instelling, described in Berends and Ouburg (2005)) based on the actual (adjusted) route plan and the train positions received from the train describer system (TNV) (Goverde & Hansen, 2000). The dispatcher may also set routes manually when e.g. last-minute disruptions occur. If an approaching train is delayed and the dispatcher expects a route conflict then s/he may decide to adjust the local route plan and change the route setting orders. In case of large delays, the dispatchers apply predetermined if-then measures in order to determine an alternative train path until the next station and keep the (network) traffic control center informed.

In practice, major disturbances may influence the off-line plan of operations, thus causing primary (initial) delays, such as train delays or temporary unavailability of some routes, that propagate as secondary (consecutive or knock-on) delays to other trains in the network. In such situations, the timetable no longer provides an optimal use of infrastructure capacity,

and short-term adjustments are worthwhile in order to minimize the negative effects of the disturbances. This process is called *real-time traffic management*, which consists of modifying the timetable such that it becomes compatible with the real traffic situation. Possible traffic control actions include changing dwell times at scheduled stops, changing train speeds along lines, or adjusting train orders at junctions, stations and passing points. Other control actions involve major modifications such as changing train routes or even canceling scheduled train journeys.

1.2 Research motivation

The (off-line) design of timetables is a complicated and recurrent problem, and typically requires many months. During operations, however, unforeseen events may disrupt the timetable and cause conflicts between train paths which must be resolved in real-time. Moreover, the railway infrastructure may become more saturated by which local initial delays are more difficult to manage and easily generate knock-on delays. Hence, improving the reliability of dense train traffic requires an advanced railway traffic management system that accurately monitors the current train positions, predicts the potential conflicts and reschedules trains in real-time such that consecutive delays are minimized.

Current operational traffic management is reactive: a train driver tries to adhere to the original schedule and responds to the actual signaling aspects when the route ahead happens to be occupied. Dispatchers only reschedule the route setting plan when trains have a considerable delay, and network traffic controllers become active only when train traffic is already highly disrupted.

Many knock-on delays, however, could be prevented if traffic was pro-actively managed (Isaai & Cassaigne, 2001), i.e., dispatchers can spend their time on preventing traffic disturbances instead of only solving them when they have already happened. Based on an accurate monitoring of the actual train positions and speeds the potential conflicting routes can be predicted in advance and might be resolved in real-time. The adjusted targets (location-time-speed) would then be communicated to the relevant trains by which the informed drivers could be able to anticipate better on the changed traffic conditions. Such a traffic management system would be effective in coordinating the speed of successive trains on open tracks, securing time windows at junctions/crossings, or synchronizing the arrival of the trains at stations in case of delays and expected route conflicts.

Network dispatchers regulate the railway traffic by sequencing the train movements and setting the routes with the aim of ensuring smooth train movements and limiting as much as possible existing delays. Due to the strict time limit available for computing a new timetable, which so far is rather infeasible by using existing tools, operators usually restrict themselves mostly to a few manual modifications of the timetable. Experienced dispatchers have developed strategies allowing them simply to foresee possible disruptions well in advance and to take compensatory control actions based on local information. In general,

minor changes are preferred instead of extensive rescheduling in order to alter as little as possible the original timetable.

Basic dispatching systems have been developed to aid traffic controllers (e.g. see Figure 1.1), whose computer support is often limited to graphical interfaces and simple automatic route setting systems (a system which provides the automatic setting of a given route when a train approaches a signal). The usual policy still consists of scheduling trains following the order in the timetable or according to pre-determined dispatching rules. A practical application of simple local measures, e.g. train reordering at crossing points on a first come first served basis, has recently been introduced in the Dutch railway network (see, e.g. Hemelrijk et al. (2003) and Schaafsma (2005)). Another example is the ARI system, which includes some limited conflict detection and resolution measures (Berends & Ouburg, 2005). However, such systems do not provide an effective support when dealing with heavy disturbances in complicated networks and extensive rescheduling would be necessary to obtain feasibility. Furthermore, there is a need for more effective conflict resolution systems which are able to exploit more information about the actual status of the network.

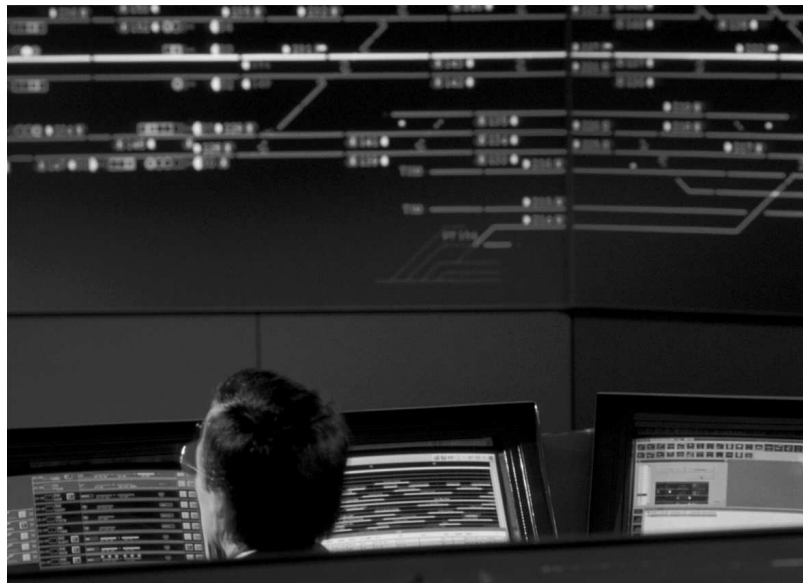


Figure 1.1: A dispatcher working at the traffic control center

Real-time train traffic management, so far, received rather limited attention in the literature. For an overview of the related approaches, the reader is addressed to the comprehensive surveys of Assad (1980), Cordeau et al. (1998), Oh et al. (2004), Törnquist (2005), to the contributions in Hansen et al. (2005) and Hansen et al. (2007), and to Chapter 2 of this thesis. Possible reasons are the inherent complexity of the real-time process and the strict time limits for taking decisions which leave small margins to computerized decision support systems. Another reason is that the real-time schedule adjustments are often limited by short time horizons of traffic prediction, and depend on the timetable robustness. However, the quality offered by railway services is strongly related to the efficiency of the real-time

traffic control, and any improvement of this process has a direct impact on the customer satisfaction.

Despite the efforts devoted to developing sophisticated dispatching procedures, few decision support systems exist to quickly and effectively reschedule train movements during operations (see, e.g. Schaefer (1995), Shoji and Igarashi (1997), Kawakami (1997), Vieira et al. (1999), Adenso-Díaz et al. (1999) and Arpaci and Jungherr (2006)). In this context, we also cite the European project COMBINE (see e.g. Mascis et al. (2002), Mascis et al. (2004), Giannettoni and Savio (2004), Pacciarelli and Pranzo (2006) and Mazzarello and Ottaviani (2007)) that has been of inspiration for the developments of this thesis.

Existing dispatching systems are able to provide viable solutions only for small instances or for simple perturbations. Hence, the real-time traffic management is still mainly under the control of human dispatchers who usually do not have precise information about the future evolution of train traffic and the chosen traffic control actions may be often sub-optimal (Van Den Top, 2006). For instance, Olsson and Hauglund (2004) report the factors that affect train punctuality for the Norwegian Railways, Kauppi et al. (2006) explain the limitations of the dispatching process in Sweden and Geitz (2007) gives an overview of the urgent need for railway traffic management tools in emerging economies. One of the possible reasons is the complexity of finding a good compromise between the solution quality, the time horizon (time span) of the traffic prediction, and the computational effort. In order to take effective dispatching decisions, the short-term consequences of the adopted measures should be known in advance. An accurate prediction of the effects of delays and other disturbances requires modeling the evolution of train traffic sufficiently in detail and reflecting the actual state of the network, both the dynamic behavior of circulating trains and the dispatching measures used to control traffic. Hence, the precise delay propagation cannot be predicted well by traffic controllers, especially in case of complex and large railway networks, high density traffic and severe disturbances. Furthermore, railway managers are looking for decision support systems that will enable their operators to determine implementable control actions quickly. For these reasons, there is a need for developing a more sophisticated and efficient decision support tool to forecast the delay propagation in the rail network for individual dispatching measures. This tool must operate sufficiently in advance with the aim to correctly quantify the effects of different dispatching measures and enable traffic controllers to perform frequent incremental modifications of the actual timetable in order to adapt to sudden traffic disturbances.

1.3 Research objectives

Based on the current limitations of operational traffic management, as summarized above, the following research objectives are the subject of this dissertation:

1. The main objective is to design, implement and evaluate an advanced and robust laboratory tool for supporting railway traffic controllers in the everyday task of managing timetable disturbances.

2. An innovative model for railway traffic optimization is needed in order to predict accurately train traffic flows and to enable the computation of globally improved schedules, i.e., all trains are managed simultaneously in a railway network for a given period of traffic prediction.
3. The development and implementation of fast and effective (re)scheduling algorithms for the real-time management of a complex railway network have to be addressed. The objectives are to predict the evolution of train traffic within short computation times and to improve the punctuality by pro-actively detecting and solving train conflicts.
4. A better use of rail capacity and a further improvement of punctuality is to be achieved by an iterative adjustment of train orders and routes in case of disturbances. The implementation of problem dedicated algorithms should highlight the potential use of rerouting instead of only rescheduling the trains in order to limit the delay propagation as much as possible.
5. Constructive algorithms for the dynamic modification of running times are to be developed and implemented that satisfy the timetable constraints of train orders and routes and guarantee the real-time feasibility of the running time profile, while respecting the signaling and safety systems in use.
6. A short-term traffic planning and control over a time period up to several peak hours is to be achieved in real-time since delays propagate considerably in time and space during heavily perturbed operations.

The achievement of the first objective clearly requires a strong connection with the resolution of the other five objectives. In fact, the development of an advanced decision support tool for real-time train dispatching must include suitable models and algorithms for retiming, reordering and rerouting of trains running through a railway network during a given time period of traffic prediction.

1.4 Research relevance

The problem of reliability of railway networks is getting more and more interest from scientists. Railway network optimization with respect to performance of individual links and nodes is reasonably well understood. Less well developed is the knowledge of the performance of larger dispatching areas and their optimization after the occurrence of unexpected events, such as train breakdowns, track damages, power failures and bad weather. The design of control actions is an important issue in such a way that a timetable disturbance has least impact on operations, and more predictable travel times and less vulnerable transport systems can be achieved.

The robustness of railway systems has to be improved and the conditions should be changed such that the propagation of train delays can be mitigated, e.g. by optimizing the traffic management with respect to the impact of disturbances. Railway networks suffer from incidents, accidents and congestion as road networks, but railway traffic differs significantly from road traffic because of scheduled line services, signaling and safety constraints, as well as limited opportunities for overtaking. Changing train orders and routes is a possible real-time traffic management measure that copes with a local disturbance.

The optimal use of railway networks compared to the impact assessment of unpredictable incidents can be efficiently modeled by global conflict resolution methods, which solve train conflicts and predict the delay propagation in the overall studied area. This is rather new for transport networks and especially for railway traffic management although much can be borrowed from other disciplines. Optimization of control measures (anticipating the reaction of railway customers, network managers and service providers) provides new possibilities to get a considerable higher network performance than the existing practice of iterative adjustment of control measures to changing traffic conditions and travelers' behavior.

For these purposes, advanced methods for optimization of real-time traffic control of bottlenecks have to be developed. This research aims at the development of an advanced real-time dispatching tool that would act in case of incidents as a decision support system to evaluate the impact of various traffic control strategies, such as retiming, reordering and rerouting of trains, on the performance of the network and the trains, and to propose a set of optimal traffic management measures. This includes the development of an optimization model for real-time traffic optimization based on flexible mathematical models, the introduction of suitable heuristics and enumerative methods for scheduling and routing problems based on constraint propagation, and the test of models and algorithms for dynamic traffic control in a simulation environment representing a dispatching area of the Dutch railway network. The efficiency of the advanced model for dynamic traffic control will be compared with the capability of simple dispatching strategies taking into consideration the influence of network structure and timetable characteristics. Moreover, the development of suitable objective functions, efficient algorithms for conflict resolution and powerful techniques to reduce the processing times of the support tool and to provide a robust real-time traffic control of large networks offers a great challenge for inter-disciplinary railway operations research and traffic engineering.

1.5 Thesis contributions

This thesis presents an innovative contribution to the combined resolution of the research objectives of Section 1.3. Next, we briefly introduce the main achievements.

A decision support system for real-time traffic management (Objective 1) is developed to assist train operators in their tasks and to provide dynamic train speed targets. The computerized system prevents the decision maker from taking less effective decisions, such as

causing a deadlock situation or unsatisfactory throughput (Goverde & D'Ariano, 2005). The proposed tool is able to estimate and control the future evolution of the railway traffic considering actual train positions, signaling and safety operating rules and conditions, as well as dynamic train characteristics. The resulting solution presents a new feasible plan of arrival and departure times, minimizes train delays with respect to a disrupted timetable and is compatible to the real-time train positions and speeds.

In our decision support system, a detailed discrete optimization model (Objective 2), called the alternative graph formulation (Mascis & Pacciarelli, 2002), is implemented to optimize the railway traffic flow when train operations are perturbed. This graph models all possible scheduling alternatives for a given set of train routes, and is able to efficiently treat the no-store aspect of the train scheduling problem due to trains traveling in the same or opposite directions. The no-store constraints of the optimization model require that a train, having reached the end of a first track segment, cannot enter the subsequent segment if the latter is still occupied by another train, thus also preventing other trains from entering the first segment. Special alternative arcs represent the available operational choices such as the train order at a crossing or merging section. A decision is made by selecting one of two alternative arcs which then fixes a precedence constraint between two trains at a potential conflict point. This alternative graph can be used to model different signaling systems and therefore offers a high reliability and flexibility of conflict resolution. In case of fixed block signaling, each block signal corresponds to a node in the alternative graph and the arcs between the nodes represent the blocking times and headway times respectively. A feasible schedule assigns passage times to each node such that all precedence constraints are satisfied (i.e., alternative graph solution). The objective function is the minimization of the consecutive delays for all trains at all visited stations and relevant points of the network (D'Ariano & Pranzo, 2004). The main value of the alternative graph is the strict consistency of the structure and the level of detail requested that can be included in the model. In fact, this graph incorporates a description of the network topology at the level of railway signal aspects and operational rules. Moreover, it can easily include other constraints relevant to the railway practice, without making the formulation more complex, such as:

- Flexible arrival/departure times at scheduled stops: A train may leave a station within a time window of [minimum, maximum] departure times in order to improve the reliability of operations, without decreasing the capacity of the lines. This can be formulated as an additional length for special arcs representing the objective function in the graph modeling (see D'Ariano, Pacciarelli, and Pranzo (2005) and D'Ariano, Pacciarelli, and Pranzo (2007b)).
- Route booking for trains approaching a station or a corridor: An interesting strategy for advanced traffic management consists in letting the circulating trains run at their scheduled speed and wait only at scheduled stops (green-wave). This can be modeled as an additional constraint (no-wait constraint) in the alternative graph (see D'Ariano, Pacciarelli, and Pranzo (2006) and Corman, D'Ariano, Pacciarelli, and Pranzo (2008)).

- Minimum required time for passenger and rolling stock connections: These important operational constraints of passenger satisfaction and operations performance are formulated as additional arcs in the graph (see D’Ariano, Corman, Pacciarelli, and Pranzo (2006b), D’Ariano, Corman, Pacciarelli, and Pranzo (2006a) and D’Ariano, Corman, Pacciarelli, and Pranzo (2008)).

In the proposed alternative graph formulations, a train scheduling problem is considered in which train sequences are optimized over a given set of train routes. Based on this assumption, sophisticated scheduling algorithms are designed and implemented to improve the punctuality compared to rule-based train dispatching (Objective 3). Basically, two classes of scheduling algorithms are implemented: rule-based dispatching (see, e.g. first come first served or train priorities (D’Ariano, Hansen, & Hemelrijk, 2006)) and innovative global conflict resolution methods based on alternative graph properties (i.e., greedy heuristics and an exact method (D’Ariano, Pacciarelli, & Pranzo, 2007a)).

A major contribution of this thesis to solving efficiently and timely the train scheduling problem consists in a branch and bound algorithm with fixed train routing. Given a perturbed timetable, the algorithm computes a new optimal train schedule using dynamic implications (see Mascis and Pacciarelli (2002)) and several speed-ups based on the infrastructure topology (see Pranzo, D’Ariano, and Pacciarelli (2005)). Computational experiments demonstrate that a truncated version of the algorithm provides near-optimal solutions within a limited computation time (D’Ariano, Pacciarelli, & Pranzo, 2007a).

The branch and bound algorithm is also used to compare near-optimal solutions when varying the departure times of trains at stations. Extensive tests show that flexible departure times may offer more freedom to solve conflicts and increase the punctuality without decreasing the throughput (D’Ariano, Pacciarelli, & Pranzo, 2008).

We next introduce the compound train sequencing and rerouting problem (Objective 4) in which train routes are also problem variables. This compound problem is solved iteratively by computing an optimal train sequencing for given train routes, and then improving this solution by locally rerouting some trains. We make use of our advanced algorithms for sequencing train movements, while a local search algorithm is developed for rerouting optimization purposes (as shown in D’Ariano, Corman, Pacciarelli, and Pranzo (2006b), D’Ariano, Corman, Pacciarelli, and Pranzo (2006a) and D’Ariano, Corman, Pacciarelli, and Pranzo (2008)).

The latter approach generated further relevant questions. The first concerns the extent at which different neighborhoods or more sophisticated search schemes might improve upon the local minima found by the local search algorithm. The second question is to how to achieve further algorithmic improvements, in terms of an increase the of the solution quality and a reduction of the computation time. To this end, an innovative rerouting approach is proposed in this thesis to determine the best routes and the corresponding train sequences in corridors and stations. The effectiveness of extensive rerouting strategies is explored by incorporating the search for new routes in a tabu search scheme, aiming to escape from local

minima (see D’Ariano, Corman, Pacciarelli, and Pranzo (2007) and Corman, D’Ariano, Pacciarelli, and Pranzo (2007)). We investigate the effectiveness of different neighborhood structures and evaluate the benefits of local rerouting strategies to minimize the delays between consecutive trains. A comparison with advanced rescheduling solutions shows the high potential of iterative rerouting and rescheduling strategies to minimize train delays and to improve the use of the infrastructure capacity.

The alternative graphs adopted to model the compound train sequencing and rerouting problem assume deterministic blocking and waiting times and, thus, do not take the impact of deceleration and acceleration into account in case of hindrance. We therefore must ascertain whether a safe space headway between trains is respected and address speed coordination issues among consecutive trains (Objective 5). In the proposed train management system, the schedule provided by our optimization algorithms is adjusted by updating the speed profiles of trains according to the actual signal aspects. Furthermore, an iterative scheduling procedure can be performed to improve the solution quality, while computing an acceptable speed profile for each running train. After a finite number of iterations, new train speed profiles are obtained that comply with the existing signaling system and rolling stock characteristics (see D’Ariano and Pranzo (2005) and D’Ariano, Pranzo, and Hansen (2007)). Finally, a constructive algorithm for the computation of optimal energy-efficient running times is described in D’Ariano and Albrecht (2006).

The problem of dispatching trains during operations becomes hard to solve when dealing with large train scheduling instances. To this end, a decomposition method is developed to enable the computation of effective dispatching solutions in a rather short computation time (Objective 6). We decompose a large time period into tractable intervals that are solved in cascade and we adopt advanced scheduling algorithms to pro-actively detect and globally solve the train conflicts in each time interval (D’Ariano & Pranzo, 2007). This is of special interest in case of large timetable disturbances due e.g. to technical failures or extremely adverse weather conditions, which affect the level of service severely and are too difficult to solve manually by dispatchers, while an automated tool may be used to forecast better the propagation of the actual delays over a railway network. If the effects of standard traffic management measures are deemed inadequate by the network traffic controller other drastic measures, such as the use of emergency timetables and the cancelation of train routes, may be adopted.

In this thesis, the test beds are the hourly timetables of the Schiphol railway bottleneck and of the Utrecht - Den Bosch railway network. For these real-world instances, a large set of computational experiments proves that our automated dispatching support system provides better solutions in terms of delay minimization compared to dispatching rules adopted by a dispatcher. Specifically, computational experiments based on combinations of flexible departure times, dynamic reordering and rerouting of running trains, and train speed coordination actions are presented e.g. in D’Ariano, Pacciarelli, Pranzo, and Hemelrijk (2007) and D’Ariano, Corman, and Hansen (2008).

1.6 Thesis outline

This section gives a short introduction to each chapter while Figure 1.2 explains the relationship between the next chapters of this thesis.

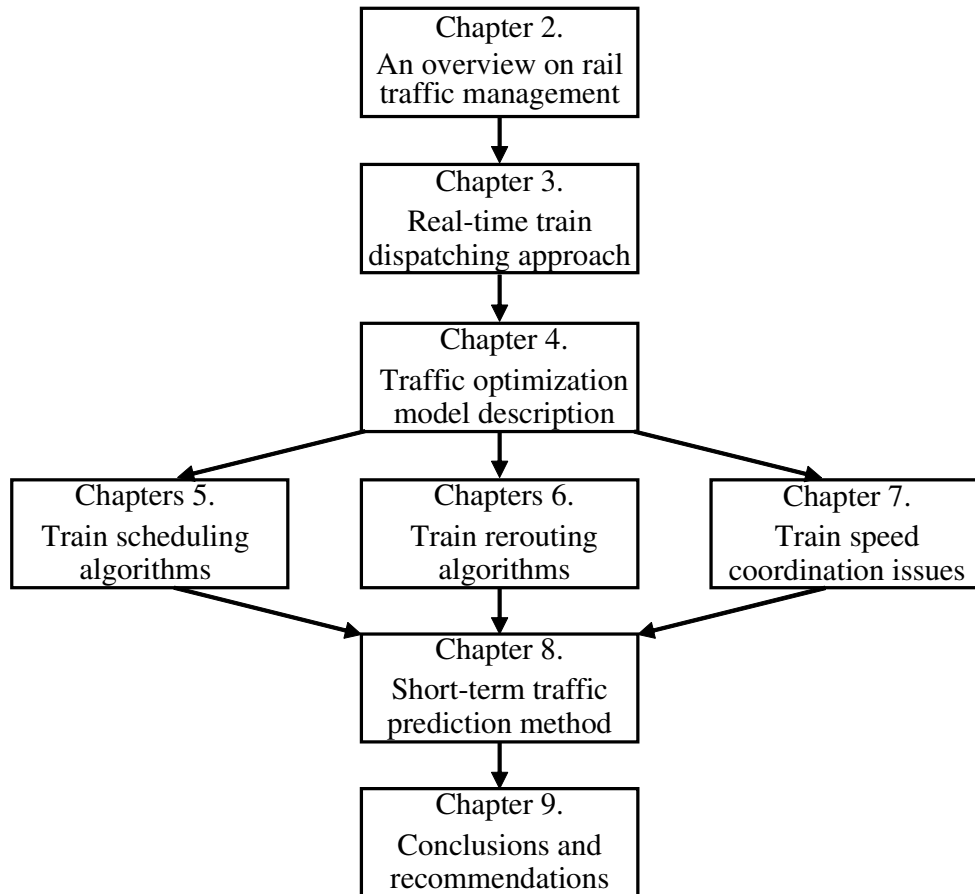


Figure 1.2: Thesis structure

Chapter 2 provides an overview of the state-of-knowledge in train traffic management. In a first part, recent contributions on timetable analysis are described and various models for timetable design are compared and classified with regard to train scheduling and routing problems. An overview of stochastic models is also presented. In a second part, the management of railway traffic is approached from a real-time perspective. Models for the resolution of train conflicts and the coordination of train speeds are characterized on the basis of different dispatching rules, i.e., changing train orders and/or routes, canceling of connections and/or train services and variation of dynamics of train movements. The traffic prediction period and the rail network size are considered important factors to evaluate the applicability and validity of the reviewed models. The third part of the survey introduces the new concept of railway dynamic traffic management inspired by ProRail managers. The current state-of-the-art limitations are finally discussed.

Chapter 3 presents the design and implementation of our advanced real-time train dispatching support system, called ROMA (Railway traffic Optimization by Means of Alternative

graphs). At the beginning of this chapter, a rigorous description of the problem and some fundamental notions on railway operations are introduced. Then, the dispatching support system architecture is illustrated, the characterization of its components and their interconnection are discussed, and each component is treated from an input/output point of view.

Chapter 4 introduces our mathematical formulation of the problem, Chapters 5, 6 and 7 describe, respectively, innovative algorithms for real-time train scheduling, rerouting and speed coordination procedures adopted by the tool ROMA. In Chapter 8, the tool ROMA is extended in order to estimate the propagation of train delays up to a day period of traffic prediction that, in railway operations research terminology, can be called short-term traffic prediction.

The main results obtained in this thesis are summarized in Chapter 9. Further research is also addressed in order to determine the next steps for the connection of the tool ROMA with a traffic simulator, which would enable more comprehensive experiments and the evaluation of the performance of various dispatching options on a much broader network scale.

In Appendices A and B, we give a resume of the notation adopted in this thesis and we illustrate the dispatching areas under study. We also include an alphabetical index of the main terms, with references to some pages of the thesis where they are described.

Chapter 2

An overview on rail traffic management

Due to the complexity of rail operations, the expected growth of traffic and the limited possibilities of enhancing the infrastructure, effective timetable design and real-time traffic management strategies play a key role in improving the level of railway service. Planning and operational processes are therefore fields rich in interesting optimization problems. In fact, there is an extensive literature on the timetable design problem. This is usually a long procedure, involving intensive negotiation among different stakeholders and taking advantage from the availability of sophisticated decision support systems (see, e.g. the Dutch timetable design system DONS (Hooghiemstra et al., 1999)). On the other hand, real-time traffic management received rather limited scientific attention, while the development of real-time decision support tools for traffic controllers enables a better use of rail infrastructure and a significant impact on train punctuality by means of effective dispatching actions. Possible reasons are the inherent complexity of the real-time process and the strict time limits for taking decisions, which leave small margins to computerized decision support systems. Another reason is that the management of real-time processes is often limited in time and space, and depends on the timetable robustness.

The aim of this chapter is to classify and compare various approaches for railway traffic management according to two different time perspectives: off-line timetabling and real-time traffic management. That is, timetabling is the process of constructing a schedule from scratch, while in real-time a schedule already exists and may be modified. We mainly focus on the problem of developing robust and optimized timetables and on the dispatching process of managing real-time timetable disturbances, without addressing important problems in cargo transportation such as empty car distribution, freight car management and locomotive assignment (we refer the interested reader to the surveys of Assad (1980), Cordeau et al. (1998) and Crainic (2003)). This state-of-knowledge overview also does not include other relevant problems such as rolling stock management, train unit shunting and crew planning. Very recent reviews can be found in Caprara, Kroon, Monaci, Peeters, and Toth (2006), focused on the European situation, and Ahuja et al. (2005), focused on the North American situation. Interesting contributions on railway operations research can also be found in Hansen et al. (2005) and Hansen et al. (2007).

2.1 Off-line timetabling

Timetable development is a complex problem in which a compromise between capacity utilization and timetable robustness has to be provided (see, e.g. Carey and Kwiecinski (1995), Wendler (2001), Barter (2004) and Landex et al. (2006)). Building a timetable may require several months, during which many variants are discussed in depth until all possible conflicts among train path requests are solved. However, during operations unforeseen events may disrupt the timetable, due to equipment failures in infrastructure and rolling stock, fluctuation of passenger volumes, behavior of railway personnel and weather influence. Train runs are perturbed, signaling and route conflicts arise with respect to the scheduled train paths. For this reason, time reserves are usually inserted in the timetable to reduce the effects of minor timetable perturbations (i.e., few minutes of train delays).

The key for high-quality timetables is a precise estimation of blocking times according to practical operations. The blocking time graph of a train represents the time instances that a train needs to run safely without hinder at design speed over a sequence of track sections. The critical track sections are situated at a location where the time gap between the blocking time graphs of two trains following each other on a route is minimal. The available time margins (buffer times) between the end and the begin of a sequence of two blocking time graphs at the critical block sections as well as the running time supplements are a measure of timetable slack. In case of absence of buffer time, the time gap between two consecutive blocking time graphs represents the minimal time headway. The buffer times are aimed to reduce knock-on delays in order to maintain a certain quality of operations in case of disturbances but, generally, can not limit initial delays. Running time reserves (recovery times) and dwell time margins are adopted to compensate the initial delays of each train (Nie & Hansen, 2005). Moreover, any timetable comprises scheduled waiting times and generates, in practical operations, non-scheduled waiting times. Scheduled waiting times are the times needed for a scheduled passing and overtaking and to synchronize the transfer times for connected train services of a fixed interval timetable. In general, these times cause an increase of dwell times (i.e., longer stopping times) or an extension of running times, except in some cases when an entire train path is moved (Wendler, 2007). For this reason, scheduled waiting times are used as indicator of timetable quality, whereas non-scheduled waiting times are experienced as train delays during operations. Since disruptive events are unpredictable, appropriate running time reserves and other time margins are distributed over the network and in the timetable (Hansen, 2004).

Larger time reserves allow increasing train punctuality, but reduce the capacity usage of the lines, i.e., the maximum number of trains which may be operated through a line per time period. Moreover, the railway infrastructure is becoming more and more saturated by which local delays are more difficult to manage and easily generate many knock-on delays (due to hinder between consecutive trains). In congested areas, the amount of time reserves that can be inserted in the timetable is limited. In fact, despite the big effort spent in the off-line development of timetables, in congested areas deviations from the timetable frequently happen, thus often requiring the restoration of a feasible schedule. As an example of this

situation, Figure 2.1 shows two time-space diagrams reporting (a) the scheduled and (b) the actual behavior of the rail traffic observed in March 2003 in a congested part of the Dutch railway network, on the line from Riekerpolder to the stations of Schiphol and Hoofddorp. The vertical axis shows a 15 minutes time interval, from 07:55 to 08:10, while the horizontal axis reports the distance from Riekerpolder to Hoofddorp. The diagonal lines in the diagram represent train positions during the time. Diagram (a) is the same for all the days of the observed period, while Diagram (b) reports on the daily observed behavior of the trains.

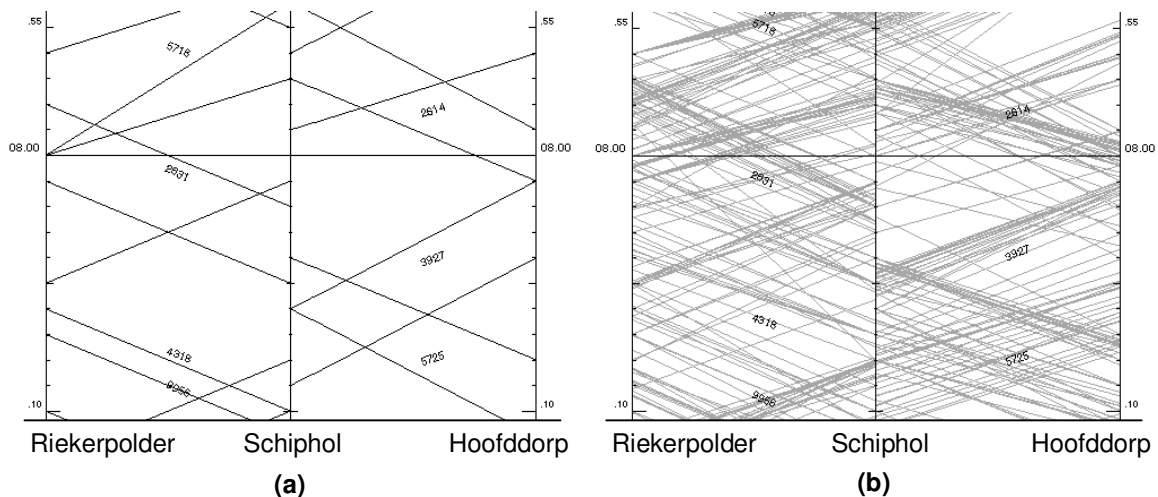


Figure 2.1: (a): Time-space diagram of the timetable, (b): Time-space diagram of the observed trains behavior during March 2003 (Source:ProRail)

In complex railway networks, the main performance factors to assess the quality of a railway system are its ability to perform well under disturbances, i.e., robustness, and to transport passengers and goods with a low variability of rail operations, i.e., reliability. Precisely, robust and reliable timetables should be able to deal with small delays, whereas larger disturbances and their propagation may be prevented or reduced significantly by suitable dispatching actions. An optimal timetable design contributes to improving the performance of railway operations while increasing the quality of passenger service and decreasing costs due to delays. In what follows, we describe the evaluation measures of timetable robustness and reliability.

The stability of railway systems can be defined as the aptitude to return to normal operations after disturbed circumstances. When unexpected events appear in the operating conditions, the railway traffic will be irregular for a transitional time period. A railway system is instable if the train traffic is irregular for a too long time period. That is, delays propagate quickly throughout the railway network in time and space, possibly causing a domino effect of increasing train delays. However, a common definition of stability and robustness of operations in transport networks does not exist and a validated, realistic model for the on-line prediction of the amount and settling time of the consecutive train delays in heavily occupied networks in case of recorded initial delays is still lacking.

Goverde (2005) describes an analytical stability theory for evaluating the timetable robustness. A discrete-event dynamic system models railway network timetables on the basis of

a linear system in max-plus algebra (an earlier contribution can be found in Goverde et al. (1998)). The max-plus model includes train interdependencies resulting from the scheduled process times, the timetable structure, and the infrastructure constraints (including signaling system). Using max-plus spectral analysis and critical path algorithms, the stability of large-scale periodic railway systems is analyzed to support the design of robust timetables in dense railway traffic networks. Goverde and Odijk (2002) present a tool, called PETER, for the automatic transformation of conventional periodic network timetable constraints into a max-plus state matrix of the travel times, minimal time headways and transfer times, for critical circuit and recovery time analysis. The impact of an increase or decrease of travel times and buffer times on the timetable slack and on the location of the critical circuit of periodic network timetable is estimated rapidly. PETER also computes the delay propagation of different delay scenarios in order to find the most sensitive links in the network, i.e., the time-space location of the less effective timetable time margins. However, the estimation of network timetable slack by the max-plus approach is deterministic and does not permit to change the train order of a given timetable.

The punctuality analysis represents the other important measure of rail operation performance and is often used as standard performance indicator. The punctuality of train services of the Dutch Railways is actually estimated on the basis of train detection data from measuring devices in the vicinity of large stations. This corresponds to ex-post delay estimations with regard to daily and hourly patterns in the variations of the arrival and departure scheduled times at stations. Furthermore, train delays are considered only if larger than a certain value, usually three minutes. However, smaller train delays which are due to non-technical reasons, in general, are representing a vast majority of all delays and their influence on the level of train service could be minimized by a higher precision and adequate slack of the timetable, as well as automatic piloting of trains according to their actual delay (Hansen, 2001).

To estimate precisely the punctuality, train detection devices (e.g. sensors of the safety and signaling system) are an important source of traffic flow data, i.e., the number of trains per unit of time. Goverde and Hansen (2000) have developed a tool, called TNV-Prepare, that is based on TNV-data from automatic train describers. TNV-Prepare enables accurate analysis of infrastructure utilization by matching information from safety and signaling system to train numbers. Infrastructure and train number messages are stored with a precision of one second, monitoring the change in status of a certain infrastructure element, such as a track section that gets occupied or has just been released.

Goverde (2005) uses TNV-Prepare to enable accurate calculation of train movements on track section occupancy level, stability analysis of infrastructure utilization, and estimates of arrival and departure times at platform tracks in complex railway stations. A linear regression analysis and accurate empirical data are adopted to investigate the dependencies between train lines, including transfer connections and conflicting routes.

A relevant aspect of delay management, that has recently drawn attention of researchers, is how to reduce or increase delays of trains in such a way that the inconvenience for passengers is minimized. An optimal coordination of connected train services is thus important to

improve punctuality whereas at the same time reliability of connections is obtained. In this context, Goverde (1998) introduces an innovative analytic approach for the deterministic modeling of passenger waiting times by means of recursive equations in max-plus algebra, and uses this model to solve the problem of determining the wait or no wait decision of transfer connections in case of delays. In other words, for each possible connection one has to decide if a connecting train should wait for a delayed feeder train or if it is better to depart on time. Schöbel (2001) also considers the problem of which connected train services at railway station should be maintained and which can be dropped in case of small disturbances. Differently from the previous approach, she proposes a linear mixed integer programming model. The goal is to minimize the inconvenience for all affected passengers, and her approach is considered for practical instances of the German railways. In Vromans (2005) and Vromans et al. (2006), a procedure to create more homogeneous timetables has been proposed by reducing the running time differences and imposing a similar number of train stops per track section. This is done to decrease the propagation of delays caused by interdependencies between trains. Some experiments on a Dutch railway corridor enable a comparison between real-life heterogeneous timetables with more homogeneous timetables. The obtained results show that the proposed measures are likely to improve the punctuality of rail operations.

In the remaining part of this section, we intend to review some of the most recent contributions dealing with optimization models for timetable design with regard to both freight and passenger trains (as shown in the outline of Figure 2.2). As described in Cordeau et al. (1998), timetables are generated by routing trains with corresponding start and end locations as well as station stops, and by sequencing the track segments to be used by each train, and when each track segment will be occupied. The first part of the review is devoted to routing problems, the second part deals with the temporal dimension of train management and the third part integrates both routing and scheduling decisions into a compound optimization problem. The final part of the review is focused on describing stochastic analytic and simulation methods.

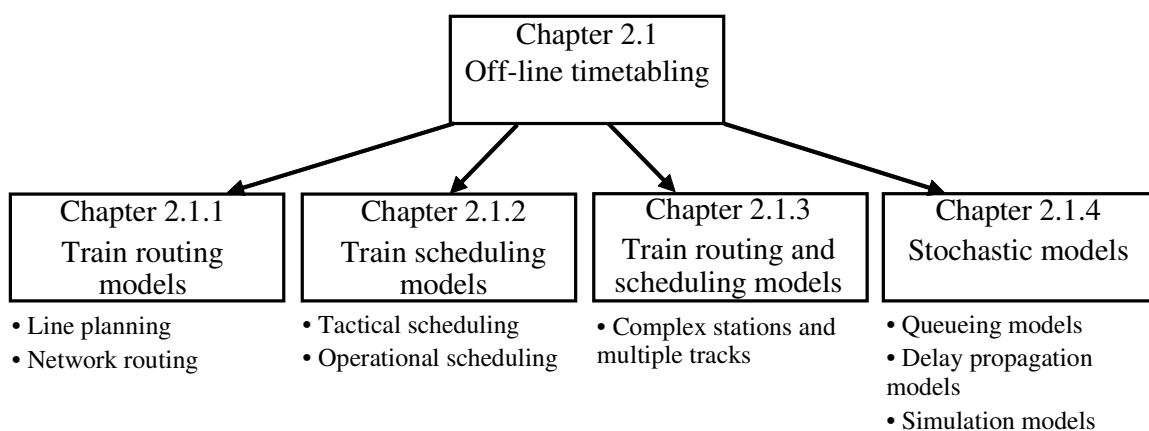


Figure 2.2: Road map of the off-line timetabling section

2.1.1 Train routing models

We consider two types of optimization models addressing the routing of trains. The line planning problem is to decide the routes for passenger trains as well as the types and frequencies of each train route (Caprara, Kroon, Monaci, Peeters, & Toth, 2006). On the other hand, network routing models address the different problems related to freight train routing. In both cases, train scheduling models are adopted to determine the arrival and departure times of the trains at all the relevant passing and/or stopping locations.

Line planning

Passenger railway timetables are mostly cyclic (at least in Europe), i.e., trains operate regularly with respect to a cyclic time period. Trains are grouped on the basis of their routes and platform stops. A train line specifies a set of trains which differ from each other only in their arrival and departure times. The cycle time of a line, e.g. one hour, denotes the interval of the departure times of the trains connected in the same direction.

The line planning problem considers how to choose a set of operating lines in a network of tracks, such that the provided transportation capacity is sufficient to meet the passenger demand. Typical objectives are maximizing the passenger service while minimizing the operating costs. The line plan does not incorporate the exact timetable for the operated lines. In the Netherlands, three different line types were distinguished: intercity trains (which do not stop at minor stations), interregional trains and local trains (which stop at all stations). Since 2007, intercity and interregional trains are merged into a single type. An approach to solve line planning problems for individual train types can be a priori split of the overall passenger flows, according to the frequency and speed of the lines, into the demands for the separate systems. Once the line plan is complete, a timetable for its train lines can be constructed.

Bussieck (1998) proves that the line planning problem belongs to the class of intractable recognition problems which justify an integer linear programming approach. He searches for a line system that maximizes the number of direct travelers, i.e., the number of travelers that do not have to change trains during their journey. Bussieck (1998) uses general integer variables denoting the frequency of each line and assumes that all trains have the same fixed capacity. The latter assumption strongly reduces the number of decision variables. Furthermore, an aggregation of the decision variables is used and the capacity constraints of the trains are relaxed. The model is solved by first applying several preprocessing techniques, and then by a general purpose solver. Finally, several valid inequalities are described to improve the line planning lower bound.

Goossens et al. (2004) present a model formulation of the line planning problem with all lines of exactly one type of trains and with the objective to reduce the total operating costs. Assumptions on the traffic demand are made such as passenger flows are symmetric and each line is always operated in both directions. A branch and cut approach is applied

based on preprocessing techniques combined with a variety of valid inequalities methods to improve the line planning lower bounds and a number of strategies for the branching process. Line planning models with multiple train types are discussed in Goossens et al. (2005). In this second paper, the authors consider the problem of designing a line system for several lines and train types simultaneously, without splitting the passenger flows a priori. To reduce the problem complexity, the origin and destination flows of the passengers are combined and disaggregated. In fact, the authors only minimize costs while passengers are not directly considered in the objective function. The results on practical problems show that solutions can be provided within reasonable time and quality by a branch and cut approach, even for the simultaneous optimization of several transportation systems.

Scholl (2005) has recently analyzed the problem of routing passengers through an expanded line network to minimize the number of transfers or the transfer time. That is, passengers have to follow an origin-destination pair through the railway network while their objectives have to be satisfied. Conflicting optimization criteria are proposed to serve all train operators while maximizing passengers convenience and minimizing the costs of the public transportation company. To solve this complex problem, a specific network is used to model the passenger connections and feasible solutions are generated by using Lagrangian relaxation and heuristic decomposition algorithms.

Network routing

The freight train routing problem includes assigning an origin-destination pair of demands into cars, assembling and disassembling cars into blocks, grouping and ungrouping blocks into trains, and determining the routing and frequency of trains. In what follows, we shortly review models dealing with the blocking policy, followed by models addressing the train routing and makeup problem.

In North America, there is a large number of individual shipments that cross over the main railroads from their origins to their respective destinations. To reduce the operating costs due to winding up of shipments and delay management over the railroad network, a set of shipments are sorted and grouped in order to be placed on outgoing trains. A block is created to associate individual cars that share a common origin and destination. The objective of the railroad blocking problem is therefore to select which blocks to set up at each yard and to assign the sequences of blocks to deliver each shipment so that the total carrying and handling cost is minimal. This classification plan is a very difficult network design and routing problem containing billions of decision variables.

Among the recent research on the railroad blocking problem, Newton et al. (1998) model the problem as a network design model (i.e., yards are represented by nodes and blocks by arcs) and formulate it as a mixed integer program in which attractive paths for each shipment are generated by solving a shortest path problem. They develop a strategic decision support tool based on advanced algorithms such as column generation and branch and price. Barnhart et al. (2000) use the same formulation of Newton et al. (1998) and propose

a decomposition method by using Lagrangian relaxation techniques. Their approaches focus on determining a near-optimal solution, but the computation times are not scalable with the increase in problem size, which inhibits a practical use unless shipments are limited to following a number of predetermined paths.

Ahuja et al. (2004) enable the formulation of the blocking problem as an integer programming problem. However, this formulation is too large to be solved to near-optimality using existing commercial software. Therefore, they developed a dedicated algorithm to solve these problems, taking advantage of the particular problem structure. The results demonstrate significant savings in intermediate handling of cars. A specific application of very large-scale neighborhood search techniques has been introduced in a potential commercial environment to solve the problem to near-optimality within few hours. The authors also state that this algorithm can be generally applied to solve large-scale and complex optimization problems arising in railroad planning and scheduling.

Blocking models are dedicated to build blocks at each yard of the network and select which cars should go into each block, while routing and makeup models specify train routes and assigns potential blocks to each circulating train. In this phase, a complete freight and train plan is computed. However, the scheduling task (i.e., the assignment of train arrival and departure times) is often solved afterwards. We now review some further network approaches for the routing of freight networks.

Marín and Salmerón (1996a) and Marín and Salmerón (1996b) study the tactical planning of rail freight networks to determine the optimal assignment of the trains and the routing of freight cars. The complex problem is modeled as integer linear programming based on a service network, including car transfer and classification and a limited availability of freight trains. To solve large networks, local search heuristics are proposed (i.e., descending heuristic, simulated annealing, and tabu search) to determine separately the assignment of cars to the routes and the choice of train service frequencies. The proposed objective is to minimize the operating and investment costs. The heuristic approaches are also compared with a branch and bound algorithm on a test case based on small size networks. In the proposed computational experiments and statistical analysis, simulated annealing outperforms the other heuristics in terms of solution quality but requires more computation time.

Various service network design aspects can be integrated in a mathematical network design formulation in order to minimize operating costs, meet the requirements of specific customers and respect all the railway regulations. In this context, Gorman (1998) proposes a novel decomposition method to address the weekly train routing and sequencing problem. The mathematical formulation has binary variables associated with each potential train service, the time horizon is discretized in hours and the problem is solved for actual train departure times. This work emphasizes the interesting perspectives offered by advanced heuristics, such as genetic and tabu search algorithms, that are used to generate aspirant train schedules, which are evaluated with the objective of minimizing the sum of operating and economical costs. Computational experiments on data from a major U.S.A. freight railroad produce a strategic scenario analysis of their operations that enable improvements with respect to the methods used by infrastructure operators.

2.1.2 Train scheduling models

The scheduling problem addresses the development of timing and ordering plans for both freight and passenger trains on the basis of the assigned train routes. The number of possible solutions can become very large depending on e.g. the network structure, the amount of traffic and type of trains. The train scheduling problem is a very large-scale integer programming problem, known to be NP-hard (Garey & Johnson, 1979), (Ullman, 1975). The train scheduling can be carried out with different time perspectives, i.e., on a tactical or operational level. Tactical scheduling usually involves scheduling for a large traffic network on a day-to-day basis and the time available for creating the timetable may be several months. Operational scheduling has a shorter time frame and is initiated closer in time to the departure of the trains. Note that tactical scheduling has a comparably less time restriction and focuses on solution quality rather than algorithmic speed. This classification has been introduced in the railway field not so long ago (see, e.g. by Harker (1995)) and successively largely adopted (as described in Törnquist (2006)).

Tactical scheduling

The tactical plan of operations, that can be executed in real-time, is called a *master schedule*. Tactical scheduling focuses on a network level with the objective to satisfy the demand of several stakeholders and to cope with the maintenance of railway infrastructure. Models and algorithms for tactical scheduling usually perform optimal slot allocations (train paths) for each route or even block section (i.e., time windows in which the block section is dedicated to a specific train) without a strict time limit of computation. At least in Europe, this is the traditional procedure to create timetables.

Early methods for train scheduling on a single track line with meets and overtakes date back to the paper of Szpigel (1973). He describes a model for solving conflicts at meets and overtakes based on the job shop scheduling problem. A branch and bound algorithm is adopted for scheduling trains on a single track railroad in Brazil. Branching is carried out by selecting two trains and fixing the meeting constraints between them. Computational experiments are presented for five tracks and ten trains.

Jovanovic (1989) introduces a general formulation of the minimum tardiness cost train dispatching problem as a mixed integer program with binary variables indicating the location of meeting points and continuous variables representing the arrival and departure times. The model can be used for both single and double track segments and a complex set of real-world constraints imposes logical conditions concerning meeting, passing and following of trains. Algorithms for the allocation of arrival and departure times are presented considering a prescribed ordering of trains with the objective of minimizing the sum of the costs of all deviations from an original schedule.

Jovanovic and Harker (1990) and Jovanovic and Harker (1991) propose the SCAN (SCHEDULE ANALYSIS) system for the tactical scheduling of trains and maintenance operations. The

advanced tool enables the creation of robust schedules combining simulation and combinatorial optimization, adopting the algorithms developed by Jovanovic (1989). The system starts with an initial schedule and verifies its feasibility on various traffic scenarios by separately analyzing each line of the network. This procedure incorporates a simulation method to model train movements and interactions (i.e., a probability distribution of the running times is generated using Monte Carlo simulations). Time window constraints on the arrival and departure of each train at the boundaries of the dispatcher's territory and at other relevant points along the line are also present. The system performs well on a real-life network with 24 lines and approximately 200 trains and 130 meets.

Carey and Lockwood (1995) describe methods and algorithms for the train dispatching problem on a single line composed of several links and stations, where overtaking can take place. The line is dedicated to traffic in one direction but trains operate at different speeds. The scheduling problem is formulated as a mixed integer program incorporating headway constraints, bounds on departure and arrival times at stations, and additional other constraints. An initial schedule is created by scheduling each train individually and a heuristic is successively applied to improve the schedule to minimize deviations from the desired schedule. In a companion paper, Carey (1994a) proposes an extension from one-way to two-way tracks and shows that the same solution methodology still applies in that more complicated situation.

Brännlund et al. (1998) introduce an optimization approach to determine a profit maximizing schedule where profit is measured by estimates of the value of running different types of services at specified times. The utility function therefore maximizes the allocation of track capacity. The problem is formulated as a general integer program and is solved with a Lagrangian relaxation approach in which track capacity constraints are relaxed. They assign a price to each track and separate the problem into dynamic programs, each one referring to a physical train. Near-optimal solutions are computed for a single track with 17 intermediate stations in Sweden and a daily timetable with 26 passenger and freight trains.

Oliveira and Smith (2000) and Oliveira (2001) model the train scheduling problem for a single track railway network as a job shop scheduling problem, and include in the model several additional real-world constraints. They point out the importance of considering constraints related to the satisfaction of required train services, such as passenger or rolling stock connections at station. Their objective function is the minimization of the total delay.

Caprara, Fischetti, and Toth (2002) and Caprara, Monaci, Toth, and Guida (2006) present a linear integer programming model based on a graph theoretic representation of the train timetabling problem, considering track capacity and operational constraints. Time is discretized in minutes and graph nodes are used to model departure and arrival times at each station. In both papers, Lagrangian relaxation is used to derive bounds on the optimal solution value and to drive heuristic procedures. They report computational results on real-world instances (provided by an Italian railway company) that are based on a single, one-way track linking two major stations, with a number of intermediate stations in between.

Zhou and Zhong (2005) consider a multi-objective train scheduling problem for a Chinese

rail line with different categories of circulating trains, i.e., high-speed and medium-speed trains. The authors model the problem as an integer programming formulation with acceleration and deceleration time constraints to run trains in double-track train timetabling applications, and introduce dominance rules to compute Pareto-optimal schedules in an intercity passenger corridor. The resolution method is a branch and bound algorithm with a breadth-first search scheme and a bicriteria objective function, minimizing the scheduled waiting times for high-speed trains and the total travel times for both high-speed and medium-speed trains. A beam search algorithm is also proposed to generate non-dominated solutions for the proposed bicriteria problem.

In Zhou and Zhong (2007), a single track train timetabling problem is formulated as a generalized resource-constrained project scheduling problem with a set of operational and safety constraints. The formulation is similar to a job shop structure, where each train is assumed to have a predefined constant departure time, traveling route and running time. A branch and bound method is adopted to obtain optimal schedules to minimize the total train travel time, which is the sum of the running time and delay due to conflicting trains. Train conflicts are solved chronologically by adding ordering constraints between trains, and sub-problems are then generated according to the selected precedence relations and solved as longest path problems. To reduce the search space, a fast lower bound rule is proposed to estimate the minimum additional train delay for resolving all the existing crossing conflicts in a partial schedule (see also Higgins et al. (1996)), an upper bound is constructed by a beam search heuristic method and a Lagrangian relaxation technique is adopted to relax segment and station headway capacity constraints. Computational experiments show that the Lagrangian relaxation technique can generate good lower bound estimates but its computation time increases drastically when enlarging the number of considered trains. On the other hand, the fast lower bound rule can improve considerably the performance of the branch and bound algorithm. Finally, the solution quality of the beam search heuristic method is compared with other constructive methods.

Another effective policy of timetable design focuses on cyclic timetables that have special properties of synchronization, periodicity and symmetry. Synchronization is the coordination of the departure of a train to arrivals of other trains to offer a connection for transferring passengers. Main railway stations are served by various train lines of different directions which are synchronized to offer seamless transfer opportunities. In a periodic railway timetable, train lines are operated with regular intervals throughout a day and consistent synchronized transfers are provided at stations between train lines of different type or directions. A periodic railway timetable is called symmetric, if for every directed line there exists another directed line serving the same stations just in opposite order, and the trains running on the two lines are connected by symmetric arrival and departure times at scheduled stops and/or by symmetric passing times at relevant points in the network. Moreover, the concept of symmetry makes only sense, if the running and stopping times are the same for both directions of the same traffic line, and if the passenger flow is also symmetric (Liebchen, 2004). However, cyclic timetables impose additional restrictions on the train schedules, which bounds the amount of time reserves that can be built during the timetable

design process. We next give a recent literature review of the related approaches.

Schrijver and Steenbeek (1994) use a combinatorial model to construct periodic railway timetables with periodic time window constraints for arrival and departure times at stations. This is an NP-complete problem and is based on the Periodic Event Scheduling Problem (PESP) introduced by Serafini and Ukovich (1989). Schrijver and Steenbeek (1994) also propose a powerful constraint generation method which is capable of computing feasible timetables for the national Dutch railway network within a few minutes. Odijk (1996) describes a cutting plane algorithm based on PESP and applies it to solve a real-life example with 6 platforms and 12 stopping trains. The example demonstrates that a family of timetables can be utilized to specify railway station infrastructure and reasonable extensions.

Goverde (1999) introduces a model with mixed integer linear constraints to compute periodic network timetables with an optimal network synchronization and allocation of time reserves over the network. The variables are the buffer times and the objective function is to minimize a weighted sum of individual costs for each buffer time. A convex cost function of the buffer times is derived by representing the individual required performance of a tight connection, a reliable connection, or a transfer connection, and evaluating the priorities between the individual buffer time performance costs.

Lindner (2000) considers the construction of a cost optimal train schedule, which is a timetable that minimizes the cost of the corresponding rolling stock plan. A combination of cyclic railway timetabling and railway line planning is developed using a branch and bound method for finding a cost-optimal line system.

Liebchen and Möhring (2002) and Liebchen (2004) generate a feasible timetable, as a PESP, with consideration of many criteria such as the amount of rolling stock required, average passenger changing time, average speed of the trains, and the number of cross-wise correspondences (i.e., the meeting frequency of feeder and connecting vehicle). However, passengers may still wait at the interchange platform for a long time.

Peeters (2003) presents an integer programming model for cyclic railway timetabling. He illustrates the modeling power of periodic constraints through extensive examples of practical situations. A constraint graph model allows the incorporation of variable trip times, flexible train connections, and station capacity restrictions. Three real-life cyclic timetabling instances of a Dutch railway network underline the effectiveness of the models and the solution techniques developed in his thesis. However, this model is limited to off-line design and optimization of network timetables and does not try to anticipate the effects of real-time timetable disturbances.

Wong and Leung (2004) and Wong et al. (2008) propose a mixed integer programming model for a timetable synchronization problem where scheduled waiting times of all train passengers are minimized. The schedule is synchronized by adjusting the running times and dwell times of each train precisely. In their formulation, binary variables are used to represent accurately the scheduled waiting times for transfer to the next available train at interchange stations. An advanced heuristic and a branch and bound method are described and compared by running experiments on a mass transit railway in Hong Kong. Some

numerical experiments also show potential improvements with respect to the adopted practice. Furthermore, the authors demonstrate that flexible dwell times allow a better average improvement of transfer scheduled waiting times with respect to fixed schedules.

Operational scheduling

Operational scheduling (or dispatch planning) is commonly used e.g. in North America and Australia. Whereas a master schedule is developed a long time before operations, the operational scheduling is created shortly in advance (White, 2007). In the case of freight transportation, trains sometimes operate without schedules and simply depart when potential time slots are available to them. In case a draft timetable (i.e., the train routes and the departure and arrival times at the corresponding origin and destination stations are generally fixed but not their exact timing) has been developed from a strategic point of view, the detailed movements of freight and passenger trains on the lines of the physical railway network still need to be precisely determined. The dispatch planning problem consists then of determining a feasible plan of train movements that is able to minimize train delays from the scheduled times while satisfying all the operational constraints. Specifically, trains can only overtake and cross each other at specific locations on single track lines (i.e., sidings or meet-points), and an operational schedule must order the trains while respecting the safety distance headway for trains traveling either in opposite directions or in the same direction. The main issue in train dispatching is therefore how to control trains utilizing maximally the track capacity while avoiding deadlocks. If no train on a segment of railway line can advance without causing a collision, that segment is said to be in a state of deadlock (Dorfman & Medanic, 2004). The resolution of this problem is subject to time restrictions (i.e., up to few hours) but this is still far from a real-time applicability.

A pioneer optimization model for the planning of meets and passes is the system developed at Norfolk-Southern Railroad by Sauder and Westerman (1983). This computer-aided dispatching system was implemented on a portion of the railroad. The proposed model is based on a simple partial enumeration scheme for generating an optimal meet-pass schedule for a single track rail line and selecting the one minimizing the weighted total delay.

Kraft (1987) develops a dispatching rule to order trains based on train priority, running times and a penalty function. A branch and bound procedure uses the proposed rule to resolve train conflicts, corresponding to the minimization a weighted sum of delays.

Kraay et al. (1991) approach a compound problem in which train velocity and interactions with the other trains are treated in sequence in order to conserve fuel and minimize train delays while satisfying scheduled arrival and departure times. They prove that some excess timetable slack could be used wisely by slowing down or pacing trains, as opposed to running trains at full speed, with important effects of reducing energy consumption, maintenance costs and the variability of expected arrival times at scheduled stops.

Kraay and Harker (1995) observe that the implementation and validation of the SCAN system (described in Jovanovic and Harker (1991) and Jovanovic and Harker (1990)) showed a

limited correspondence between tactical and operation scheduling. They therefore propose a system for generating in real-time the necessary target times to be respected by train dispatching models, with particular reference to the SCAN system. A non-linear mixed integer programming model for the real-time optimization of freight train schedules is implemented and considers explicitly the current position and relative importance of each train. The resulting solution indicates the target time for each train at each relevant point. The resolution method first determines the binary variables that specify meets and overtakes. The model then reduces to a continuous variable sub-problem. A simple heuristic approach and local search methods can be used to determine feasible values for the integer variables. The approach has been tested on a North American railroad, for which local search heuristics perform better than simple heuristics but require excessive computing time.

Higgins et al. (1996) formulate the train scheduling problem as a non-linear mixed integer program aiming to the improvement of timetables and at the development of decision support systems for real-time traffic management. Their model is applied to a long single line track. The authors present a branch and bound algorithm based on priority rules and using a shortest path method for estimating the lower bound. The criteria for conflict resolution takes into account train priorities, current train delays and expected remaining delay due to conflicts. Successively, Higgins and Kozan (1997) and Higgins (1997) propose several meta-heuristics for the real-time train scheduling problem, including local search, genetic algorithms, tabu search and hybrid techniques.

Cai and Goh (1994) and Cai et al. (1998) describe constructive heuristics for timetabling and dispatching trains on a single railway track. Their approach incorporates fixed minimum headway constraints and resolves train conflicts at crossing and merging points on the basis of a local optimality criteria. The conflict solution strategy is based on assigning trains to a position-time pair that is updated dynamically at each time instant of the traffic prediction. The proposed strategy has been implemented and adopted by a major Asian railway.

Dorfman and Medanic (2004) propose a discrete-event model for scheduling trains on a single line and a greedy strategy to obtain sub-optimal schedules. The model incorporates priority rules, similar to those adopted by dispatchers, to scheduling trains in railway networks with double-track sections. The authors show that adding information on the blocking aspect of the problem due to trains traveling in opposite directions on a line with single sections, a capacity check can be incorporated to prevent deadlocks. The resulting approach can quickly handle timetable perturbations but the quality of the heuristic approach is not compared with mathematically optimal solutions.

G. Şahin et al. (2004) model the train scheduling problem as a flexible multicommodity flow problem enabling the formulation of several practical constraints on a space-time network. Particularly, an innovative constraint permits to manage a maximum allowable delay for all the traveling trains, while traffic regulation constraints enable a correct formulation of minimum distance headway constraints. An integer programming based heuristic, a simulation-based construction heuristic and a greedy enumeration heuristic are proposed to schedule trains with the objective to minimize the total unweighted delay of all the trains.

A real-life test case of major Brazilian freight railroad is presented in which 25 trains (with fixed routes, running times and scheduled arrival times at stations) are to be dispatched over a 24-hour period on a double-direction single track that links two major stations. The results show that the integer programming based heuristic outperforms the other heuristics and the solutions computed manually by dispatchers. Moreover, the best heuristic also presents near-optimal solutions for the proposed small instances, for which the optimal solution can be computed by using commercial software packages within a few minutes. In accordance with the computational results, the authors state that the proposed approach can also be considered by railroads for real-time dispatching.

Despite the effort devoted to developing sophisticated mathematical formulations and solution algorithms, the existing scheduling methods offer heuristic results and compute solutions requiring a very large computation time. The main challenge is to detect a suitable resolution method for both planning and operational applications, enabling the computation of an upper bound on alternative scheduling scenarios by taking advantage of the specific structure of the train scheduling problem. In other words, additional information about the railway infrastructure or the type of trains should be considered to guide the search to the direction of more appropriate scheduling solutions.

2.1.3 Train routing and scheduling models

Railway networks are often highly utilized and require extensive effort to develop conflict-free schedules, because several running trains, at different speeds, pass at multi-platform stations while several constraints and objectives have to be satisfied. Searching for optimal platform stops and passing through complex stations and tracks (see Figure 2.3) is important in order to fully explore the capacity of the rail infrastructure and to design robust timetables. The literature on integrating both routing and scheduling decisions into unique optimization models experienced a slow growth from the pioneering paper of Frank (1966) and only in the recent years the related research has produced detailed models and effective algorithms for the combined adjustment of train orders and routes. We next review recent works addressing this complex category of optimization problems.

A complex routing problem in railway stations consists of assigning trains to time slots at platforms, so as to satisfy several constraints, such as headway and platform occupation. A train therefore passes through and, possibly, stops at a platform within the station. During the timetabling phase, draft schedules have to be adjusted in order to satisfy customers' priorities which can be expressed in terms of desired train arrival and departure times, platform stops, etc. In case of simple stations with few lines, the problem is relatively easy since there are a limited number of routes for each train. The problem becomes difficult when dealing with complex station topologies, having busy lines and several alternative platforms. In case of condensed timetables and scarce infrastructure capacity, routing trains in major stations becomes increasingly difficult as the chosen routes not only have to meet safety restrictions, but also guarantee reliable operations if delays occur.



Figure 2.3: Train traffic at a complex station

Kroon et al. (1997) investigate the computational complexity of several variants of the problem of routing trains through railway stations. They show that the problem can efficiently be solved considering a subset of the sections and routes of a railway station. They also prove that the routing problem is NP-complete if each train has at least three routing options. On the contrary, a solution can be computed in polynomial time, with time increasing with the number of circulating trains. Given a fixed layout and capacity constraints of a railway station, the authors finally prove that a polynomial algorithm can be adopted for routing a maximum number of trains under safety restrictions and including coupling and uncoupling of trains plus other service aspects such as the cyclicity of a timetable.

Beneath these assumptions, Zwaneveld et al. (1996) study the problem of assigning platforms and routes to trains moving through a railway station for timetable design purposes. Several practical aspects are taken into account as capacity and safety constraints and customer's objectives. Zwaneveld et al. (2001) formulate the routing problem via a node packing problem, in which each possible train route is represented as a node and each pair of routes in conflict is connected with an edge. Pre-processing techniques to reduce the search space and valid inequalities for the node packing problem are also presented. These elements are implemented in a branch and cut algorithm for solving this routing problem to optimality. A computational study is also discussed in which the model, the preprocessing techniques and the algorithm are tested based on real-world instances of the Dutch railways.

Carey and Carville (2003) deal with the problem of routing and scheduling trains at stations. A combinatorial optimization model is proposed to avoid train conflicts and to minimize schedule deviations and a weighted combination of costs. Heuristic techniques are designed according to train planners' objectives. Several practical constraints are also considered like minimal time headways and platform occupation in case of complex stations.

Given a draft timetable, the goal of Herrmann (2005) is to find routings that satisfy safety requirements and at the same time optimize a target function incorporating stability aspects.

The task is to develop a timetable whose periodicity is as small as possible to maximize network capacity by examining routing alternatives, especially numerous in station regions. This routing problem is modeled as an independent set problem while a set of delays is distributed on discrete values with equal distance in between (pulse delay distribution). The node set corresponds to all possible train routes and two nodes are connected by an edge, if their respective routes are mutually exclusive. The independent set problem is then solved using a fixed-point heuristic. Results on the Bern station region show that the tighter the timetable becomes the more effective is the improvement of the available railway capacity.

Caimi et al. (2005) also consider the problem of generating robust train routings through a station, given a timetable and a layout of the station. The problem is formulated via a node packing model (as in Zwaneveld et al. (2001)) and a fixed-point iteration algorithm is adopted to compute an initial solution. A local search scheme is then applied to increase the length of the time slot of a chosen route, i.e., the time interval during which a delayed train may arrive and find its designated route still available. A test case is based on the Bern station and a timetable of 19 trains arriving from six major directions in half an hour. Initial solutions are computed within minutes even for difficult cases while optimized solutions require some hours of computation but offer the chance to find delay-tolerant routings and to decrease the impact of late trains.

In railway networks, it is not uncommon to find segments containing multiple tracks. The design of optimal network timetables may therefore consider the flexibility of multiple lines or rail corridors, traversed by trains of differing types and speeds. This has to be done while minimizing costs of deviations from desired times, platforms and lines. We now give a review of network scheduling and routing models.

Carey (1994a) extends the original model of Carey (1994b) to introduce choices among multiple lines in each direction and choices of platforms to use for departures, arrivals and stops at stations. Successively, Carey and Crawford (2007) face the same problem on a corridor including several busy complex stations linked by multiple one-way lines in each direction, while evaluating costs of using lines, platforms and times for each train. In these papers, the problem is modeled as mixed integer linear program (MILP). Their heuristic algorithms are quite efficient and flexible (i.e., several additional information and objectives can be included) for the design of optimal timetables, and can be adopted to assist timetable planners in their task of finding and resolving train conflicts in draft train schedules.

Delorme et al. (2001) evaluate the capacity and the level of saturation of a railway junction. A greedy heuristic is implemented based on a constraint programming model and a GRASP (Greedy Randomized Adaptive Search Procedure) metaheuristic is described via a node packing formulation. The two heuristics are tested on a number of mixed traffic instances for the Pierrefitte Gonesse junction located at North of Paris. Computational results suggest to redirect the search towards a hybrid model. In fact, the constraint programming approach is more efficient on finding good schedules but has limitations in the number of evaluated routes, while the node packing approach is successful when searching for better routes but consider fixed running times within a route. Recently, Gandibleux et al. (2005) propose an

ant colony optimization metaheuristic as an alternative to GRASP, and present promising results on two real-world test cases.

Ghoseiri et al. (2004) develop a multi-objective optimization model for the passenger train scheduling problem in which the fuel consumption cost and the total passenger waiting time are minimized. A two step resolution method is proposed in which first the Pareto frontier is generated and then a multi-objective optimization is performed by using distance-based methods to compute a feasible solution. The studied railway network includes bi-directional single and multiple tracks, as well as multiple platforms with different train capacities. A sensitivity analysis of a number of worked numerical examples is given to show the applicability of the models and solution procedure.

Semet and Schoenauer (2005) study the problem of repairing a slightly perturbed timetable to minimize the total accumulated delay. A local reconstruction of the schedule is based on adjustments of departure and arrival times at stations and allocation of resources. The problem is solved with a permutation-based evolutionary algorithm. The permutation of trains is manipulated by means of a greedy heuristic that iteratively insert trains in an initially empty schedule while respecting all of the problem constraints. Experimental results are presented on a real-world test case with a single train being delayed for 10 minutes at a large connecting node, requiring timetable modifications of neighboring trains.

Lusby et al. (2006) approach the problem of routing trains through highly interconnected multiple railway lines. The problem is formulated as a set packing model in which the primal variables are all possible train paths, including the dynamics of the trains, which are represented by tree structures. Train routings are computed by a branch and bound procedure, which exploits the mathematical structure of the dual of the linear relaxation in order to efficiently generate good lower bounds. A “toy” example of a rail junction is presented to illustrate the procedure and to discuss the limits of the proposed approach from a computational point of view.

Dessouky et al. (2006) describe a train dispatching system based on reordering and local rerouting decisions. A branch and bound procedure has been implemented and deadlock avoidance checks are adopted to reduce the search space. The resulting approach is able to find exact solutions for a single track network with 14 train routes within 2 hours of computation time. The development of effective heuristics is addressed for real-time purposes.

Caimi et al. (2007) address the problem of generating a feasible timetable and propose a decomposition of the railway network into condensation and compensation zones. Condensation zones lie in the proximity of main stations while compensation zones connect the condensation zones and here traffic is less dense. They focus particularly on the timetabling and routing problem in condensation zones. The problem is modeled as an independent set problem in a conflict graph, which is then solved using a fixed-point iteration heuristic. Results show that even large-scale train scheduling problems with dense traffic and large stations can be solved within a minute.

To summarize, routing and scheduling of trains in open tracks and stations have been investigated extensively and different models for the assessment of capacity and scheduled

waiting time at junctions and crossings have been developed. However, there are only a few models which describe the relationships between scheduled train services of different lines in interconnected railway networks. Moreover, the use of stochastic optimization techniques should be considered to estimate the impact of delay propagation and the effects of alternative scheduling decisions. A timetable should be therefore designed in such a way that it can cope with small stochastic disturbances while the resolution of (unpredictable) large disruptions is leaved to real-time traffic management.

2.1.4 Stochastic models

The development of a feasible timetable is usually performed on the basis of deterministic running times and dwell times. However, train services are very sensitive to small delays that could be minimized by a higher precision and adequate timetable slack. Evaluating railway performance in case of small variations is an important aspect to design robust timetables. Generally, optimization models are based on deterministic variables and do not reflect variations in running times, time headways and order of trains during practical operations. On the other hand, stochastic approaches model these variations on the basis of assumed random distributions. The non-scheduled waiting time of a timetable is therefore a function of the track occupancy and the coefficients of variation of the time headways. We next discuss the influence of basic stochastic variables for timetable design and its impact on railway performance, as described in Hansen (2006). We focus on queueing models, stochastic delay propagation and network simulators that are applied successfully to estimate stochastic non-scheduled waiting times and to test the robustness of a timetable against disturbances in practical operations.

Queueing models

Queueing models enable the computation of buffer times needed to minimize scheduled waiting times at railway bottlenecks, where train arrival times and service times are assumed to be independent random variables. This theory that has been mainly developed in Germany and is composed of queueing models characterized by different approximation methods for the coefficients of variation of inter-arrival times and minimal time headways as a function of the track occupation rate.

Schwanhäußer (1974) develops analytical solutions for the estimation of the mean queue length as function of the distribution of initial delays, buffer time, mean headway, train sequence and priority rules. For heterogeneous train traffic, a mean buffer time of one minute is estimated in case of a minimal time headway of two minutes and all trains delayed. Wakob (1985) extends the previous method to compute the scheduled waiting time at railway stations and open tracks. A decomposition method is adopted by dividing the station layout into route sections, that are modeled as single-server queues, and assuming random train orders and priorities between different train types. Successively, Hertel (1992)

describes a queuing model for the determination of the optimal train intensity, which is situated between the minimal relative timetable sensitivity (partial derivative of mean waiting time to track occupancy) and the maximal traffic energy (defined as product of train intensity and speed).

Wendler (1999) proposes several extensions of the existing two-train queuing model to a three-train model in order to estimate the impact of available time lags between the time headways on a main railway track for intermediate merging and crossing of trains. This approach is suitable for modeling the process of independent random requests for infrastructure capacity from different train operators and estimating the resulting waiting times due to timetable adaptation of train paths, time headways and synchronization. The estimation of the scheduled and non-scheduled waiting times as a measure of operations quality of a given timetable, infrastructure and signaling system is implemented in the software tool ANKE developed at RWTH Aachen (Vakhtel, 2002). Recently, Wendler (2007) describes how to predict the scheduled waiting time for bottlenecks with mainly non-cyclic timetable structures by a semi-Markovian queuing model and assuming random train orders.

In the Netherlands, Huisman and Boucherie (2001) provide a queuing model to forecast running times and delays due to different train speeds along a single railway line. Running time distributions for each train service are modeled by a system of differential equations that can be explicitly solved by assuming deterministic free running times. For random train order and independent buffer times an infinite server resequencing queuing model is proposed where inter-arrival and service times depend on train types. Huisman et al. (2002) describe an analytically tractable queuing network model by considering only basic components of the railway network, such as stations, junctions, and connecting open tracks. However, these queuing models assume the train intervals and the service times to be independent random variables, which is questionable in case of periodic timetables with high frequencies.

Delay propagation models

Probabilistic delay propagation models have been recently introduced in literature. Weigand (1981) emphasizes the need for detailed statistical analysis in order to determine buffer times and running time supplements by route sections. Mühlhans (1990) furnishes analytic stochastic methods for the determination of train delays by a convolution of the initial and knock-on delay distributions, assuming independence. Initial delay distributions can be determined on the basis of empirical data, but a uniform distribution of consecutive delays in a network is not evident.

Carey and Kwiecinski (1994) develop simple stochastic approximations to simulate the knock-on delay occurring on a single link due to tight headway and speed variations. Carey (1999) then presents *ex-ante* stability measures for public transport services which can be used for estimating the effects of schedule deviations during off-line implementation and evaluation of alternative timetables. He uses probability density functions, such as the

probability of arriving some minutes late, based on scheduled train arrivals and departures at complex and busy stations, and proves convexity properties of these cost measures. The proposed heuristic measures can be used in the process of generating more reliable schedules. A stochastic delay propagation model is therefore proposed to compute optimal arrival and departure times and to distribute optimally time margins in the timetable.

Kaminsky (2001) introduces a heuristic limit of the buffer time distribution at bottlenecks in network timetables in order to compensate for most of the smaller initial delays. The author analyzes the blocking and buffer times in the existing timetable, as well as the recorded train delays for a large network of more than 2000 km in Germany and finds a good fit of the buffer times with a negative-exponential distribution. The available buffer time is less than one min in about 8% of the train sequences. The impact of the length of buffer times results to correspond to 80% of the recorded train delays at a number of stations by means of multiple simulation while maintaining the train order. The results show a mean consecutive delay per delayed train of about 30 seconds and a reduction of 9% of the estimated hinder between trains. The expected operations quality could therefore be improved by means of tailored buffer times per route section as function of the distribution of train delays.

Pudney and Wardrop (2004) describe a method to generate optimized train plans automatically for very general railway networks, including parallel tracks, alternative routes, complicated junctions and realistic separation rules. They use a probabilistic search technique, called problem space search, that quickly generates a large set of train plans by using a fast dispatch heuristic and minimizing delays or lateness costs. An application of this system is to plan accurately train movements for an Australian mineral railway, allowing for changes in railway infrastructure and for the introduction of additional rolling stock.

Vromans (2005) presents a stochastic optimization model for improving the stability of passenger timetables based on a two-stage recourse model. The method starts with the improvement of a given feasible timetable by redistributing the available time margins and maintaining fixed train orders and routes. The timetable is then evaluated using network wide simulation and small random disturbances. The timetable constraints and the simulation model are formulated as a large-scale linear programming problem, where the event times are the decision variables and the objective function is to minimize a weighted average delay.

Yuan (2006) proposes analytical probability models that enable a realistic estimation of the knock-on delays of trains, while predicting its impact on the train punctuality at stations and reflecting the real use of track capacity. Some important aspects are modeled at a very high level of detail, such as possible dependences of dwell times at stations, stochastic interdependencies between train movements of different lines in complicated stations and interlocking areas, speed fluctuations in case of different aspects of the station signals, the dynamic delay propagation between trains and the alteration of the order of pairs of trains inside stations.

Recently, Conte and Schöbel (2007) apply a stochastic approach based on a graphical model, called tri-graph, that is a compact representation of probability distributions. Three

types of dependencies among train delays are introduced: the delay carried over along the path of each delayed train, the delay propagated from one train to another train due to connections, and the delay due to the limited capacity of infrastructure. They propose a procedure which enables the detection of dependencies of delays of the third type without explicit knowledge about the tracks and platforms of the railway system. Some numerical experiment is presented on a part of the German railways but a comparison with other advanced optimization techniques is still missing.

Simulation models

A reproduction of railway operations in lines, stations and networks can be implemented by using network simulation models. Microscopic simulation systems, like STRESI (Schultze, 1985), RailSys (Radtke & Hauptmann, 2004) and OpenTrack (Nash & Huerlimann, 2004), model accurately the traffic flow in a railway network by using detailed information on track configuration, signaling system, timetable, rolling stock characteristics, and simple automated dispatching rules. These systems are used to estimate the effect of exogenous random initial delays on track occupation and consecutive delays of hindered trains. The induced initial delays are drawn from assumed or empirical distributions. Route conflicts between consecutive trains are automatically detected and timetable simulation can be used to estimate the timetable stability and the distribution of consecutively delayed trains in disturbed operations (e.g. stochastic dwell time perturbations or departure time delays) and different time periods (Demitz et al., 2004).

Asynchronous simulation models (e.g. STRESI) are methods to simulate railroad operations by scheduling stochastically generated train paths. The different train classes are scheduled one after the other (asynchronous) in accordance to their priority. Train path conflicts are solved by scheduling rules. The effect of random initial delays on operations are generated by Monte-Carlo simulations and measured by non-scheduled waiting times, knock-on delays, and the variation of occupancy of critical track segments. These tools are typically applied for detailed timetable design of individual routes and corridors and are not suitable for modeling network capacity or delay propagation between multiple interconnected lines.

Synchronous simulation models (e.g. RailSys and OpenTrack), are characterized by a detailed description of the track infrastructure as directed graphs, of the individual train paths and blocking times and a simultaneous train disposition. Random initial train delays generated by multiple simulations are imposed and the consecutive delay distributions referring to specific locations, lines or time periods are estimated. Synchronous simulation tools can handle very large networks, but require extensive work to model infrastructure topology, signaling and timetables. A generic approach to estimate the scheduled waiting times in order to design optimal network timetables and to achieve a certain level of punctuality, as well as the empirical verification of track occupancies and consecutive train delay distributions has still not been provided.

In the Netherlands, Hooghiemstra et al. (1998) and Hooghiemstra et al. (1999) describe the network timetable designer tool DONS, which checks the feasibility of timetable options of the strongly interconnected Dutch network. The network solver indicates whether or not a solution exists and, in the latter case, which set of constraints causes inconsistency. By relaxing some of these constraints, e.g. permitting a shorter time headway or a longer connection time, a feasible solution may be found. The effects of small disturbances, e.g. running and dwell time prolongation, on other trains can be simulated by the macroscopic tool SIMONE (Middelkoop & Bouwman, 2001), which permits an assessment of the robustness of the basic hourly timetable. Recently, a microscopic simulator, called FRISO (Middelkoop & Loeve, 2006), is under development to enable the testing of railway dynamic traffic management measures.

Furthermore, Watson (2005) compares the characteristics of current commercial tools and concludes that modeling the signaling system is necessary for stochastic simulation. Simulation tools mostly require interaction of the user in case of conflicts between blocking time graphs or the application of a predefined automatic conflict resolution strategy. For the evaluation of different dispatching measures on the stability margin and the location of network bottlenecks in case of disturbance different options can be computed. The analysis of individual link and train dependent recovery times would allow a variety of experiments to estimate the robustness of different dispatching options.

The most advanced existing simulation models (e.g. SIMONE, RailSys or OpenTrack) can estimate the knock-on and total delays of all scheduled trains in large networks for any given deterministic and stochastic initial delay at a certain location of a specific train. But the macroscopic model SIMONE does not take into account the impact of block signal spacing and sectional route release on headway and buffer times, while the microscopic simulation models OpenTrack and RailSys are still based on assumed random initial train delays at network borders and deterministic rules of conflict resolution which may not well represent the dynamic interaction between the behavior of train drivers, the signaling system and the traffic control measures. The discussed simulation models can only be used for off-line timetable analysis and not for on-line dynamic traffic management, as the computation time for large networks lasts too long, the interpretation of results requires extra time and academic skills, and the generation and evaluation of dispatching measures in case of conflicts between trains is not interconnected with the train detection, signaling and interlocking systems. A slot research engine is currently developed and tested for upgrading the simulation tool RailSys but this is aimed at supporting the timetable design and not the real-time traffic management (Sewcyk et al., 2007).

2.2 Real-time traffic management

We have shown that off-line timetabling mainly consists of designing a robust schedule, which enables a reduced propagation of delays and has to be faced periodically. Train services are planned in detail, defining several months in advance the train order and timing

at crossings, junctions and platform tracks. A robust timetable is able to deal with minor perturbations (i.e., few minutes of delays) occurring in real-time by using smart planning rules. However, no reasonable railway plan is robust or reliable enough in case of large delays or the blocking of some tracks (Vromans et al., 2006).

Despite the big effort spent, technical failures and disturbances, such as delayed trains or temporary unavailabilities of some routes, may influence the running times, dwelling and departing events, thus causing initial delays. Due to the interaction between trains, these delays may be propagated as knock-on delays to other trains in the network. In fact, trains may be required to stop in front of crossings or junctions, causing non-scheduled waiting times and longer running times due to slowing down and subsequent re-acceleration. Hence, managing railway traffic in real-time requires modifying the timetable, minimizing delays between consecutive trains and ensuring the feasibility of the resulting plan of operations. This short-term process requires effective solutions within minutes and is called real-time traffic management.

The traffic control of complex railway networks is usually managed by a set of regional control centers. For instance, the control of the entire Dutch railway network is subdivided in one main center in Utrecht, four regional centers (Amsterdam, Eindhoven, Rotterdam and Zwolle) and thirteen traffic control offices. Each traffic control center uses a centralized regional traffic control that is a complex system in which the local interlockings are remote-controlled by dispatchers and the train movements are governed in a certain railway area by signal indication.

In each dispatching area of a traffic control center, there is (at least) a dispatcher who receives, at appropriate intervals of time, actual information on the destination, current location and speed of every train, and the status (whether occupied or empty) of each track in the railway system. The dispatcher analyzes the data (checking if the timetable is coherent with the current trains positions and speeds), calculates whether and where conflicts are set to occur and solves them on the basis of experience and rules, respectively. Specifically, experienced dispatchers have developed strategies allowing them to anticipate requirements for changes to schedules and planned meets (train orders at merging and/or crossing points) early so as to have time to take compensatory action to reduce delays. Possible control actions include changing dwell times at scheduled stops, and changing train speeds along lines or train orders at junctions, stations and passing points. Other control actions involve major modifications such as changing train routes or even canceling train runs.

A real-time dispatching support system is an automated traffic management system with a short term planning, called time horizon (or time span). This system is designed to aid the traffic controllers to manage train traffic in a given time horizon with the objective to minimize arrival delays at stations and at dispatching area borders. In order to provide physically feasible solutions with short computation times, a limited time horizon and a detailed representation of the area managed by a single dispatcher is of special interest. If a short time horizon is adopted, only few trains, and few conflicts, can be detected and solved. On the other hand, a longer time horizon leads to a larger number of trains running

in the system, and a larger number of unsolved conflicts. This means that there is a trade-off between the size of the time horizon of traffic prediction and the solution quality. In fact, the solutions computed with few running trains could be myopic, since the real-time dispatching does not take into account conflicting trains further outside the time horizon. On the other hand, a conflict arising far in the future may not be as relevant as a closer conflict, since other unforeseen events could still affect the farther conflict.

In the Netherlands, the dispatchers acknowledge a train as delayed only when the delayed train is recorded to be at least three minutes behind the schedule; they get no statistical information about the past performance of the same train number, line and timetable reference point, neither about the propagation of delays along the route and lines. The only decision support the dispatchers have are a visual representation of the actually occupied block section and the corresponding train numbers on the PC screen and on the panoramic route layout on the wall of the traffic control center, eventually printed or oral information on perturbed trains from neighboring control areas.

The network traffic controllers need to supervise the train movements on a much bigger scale in order to recognize sufficiently in advance the impact of temporarily blocked tracks, and significantly delayed international, national and freight trains on the actual operations. Based on rules of thumb concerning the standard minimal headway times, the rounded-up actual delay data and the personal estimation of the probable locations and severity of present and future conflicts between different trains trying to claim simultaneously a same part of infrastructure or not being able to proceed because of congestion on the tracks, the network traffic controller has to take a decision which train to give preference, which train to hold or which train to re-route. The evaluation of the pro's and con's of different traffic control measures is made within a very short time mentally and human traffic controllers are not sure of the costs and benefits of their intervention. Especially in case of perturbations creating multiple consecutive delays, they cannot oversee the dynamic propagation of delays in larger interconnected railway networks and estimate correctly the impact of different dispatching strategies on the network performance of the available infrastructure. They would need to be supported by a consistent model of the interaction between a range of typical operations perturbations and the impact of dynamic traffic management options to be simulated and evaluated in real-time in such a way that they can select the most beneficial and less costly measure.

Automated dispatching systems can be considered as part of a Decision Support System (DSS) aiding dispatchers to reduce delays from a global perspective. Such system would be designed to quickly reschedule train movements during real-time perturbations and typically contains the following components (Figure 2.4):

1. Conflict detection (CD): Given the current infrastructure status, timetable, rolling stock information, the position and speed of each running train, find potential conflicting train routes in a pre-established period of traffic prediction;
2. Conflict resolution (CR): Given the actual train delays and predicted conflicts, propose how to solve them on the basis of the most suitable and robust dispatching op-

tions (e.g. train rescheduling, rerouting, removing train runs and/or connected train services (Takeuchi & Tomii, 2005));

3. Train Speed Coordination (TSC): Given the train orders and routes, compute new targets (time and advisory speed at key locations) to be transmitted to the drivers, respecting railway traffic regulations (e.g. minimum time headway between trains) and minimizing delays and energy consumption.

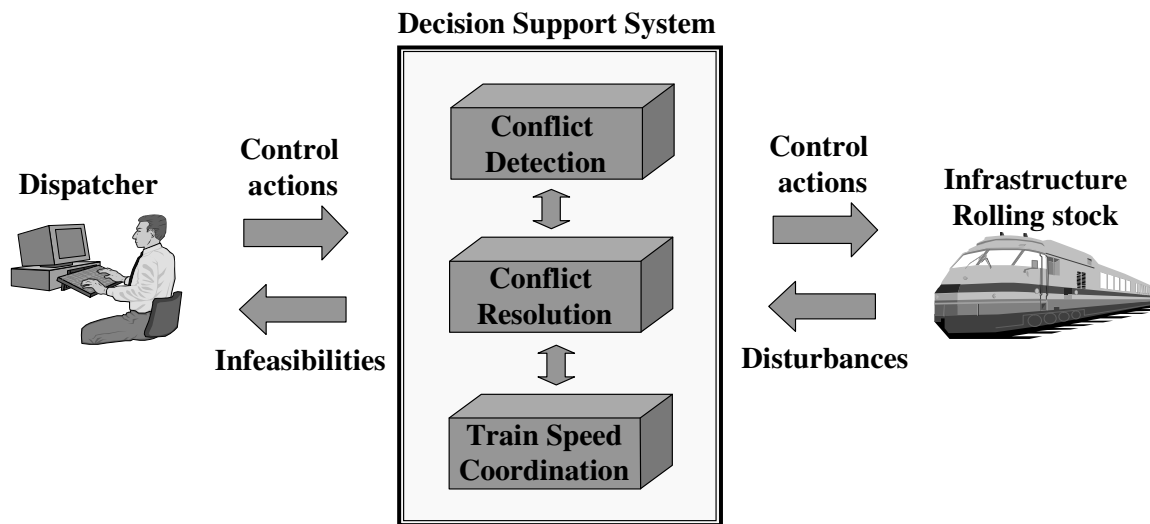


Figure 2.4: Components of a proactive train dispatching support system

Figure 2.4 shows the general architecture of a real-time train dispatching support system (a similar architecture description can be found e.g. in Shoji and Igarashi (1997), Kawakami (1997), König and Schnieder (2001), Giannettoni and Savio (2004), Gély et al. (2006), Lüthi et al. (2007) and Montigel et al. (2007)). The critical factors for a good system operability can be summarized in the limited time to compute solutions, the ability to recover train delays or even heavy disruptions, such as blocked tracks, and a fast feedback with rail operations. In the practical implementation of this system, important issues are to compute the most appropriate dispatching measure and to coordinate the actions of each dispatching support system at network level. This is a feasibility problem for which deadlock-free and conflict-free solutions have to be computed within a computation time compatible with rail operations.

Train dispatching support systems can be broadly divided into fixed and variable-speed models (as in Cordeau et al. (1998)). Models with fixed dynamics often assume that trains operate at maximum speed wherever possible. This corresponds to implement CR and TSC as independent systems that operate in cascade to compute train dispatching solutions. On the other hand, variable-speed models update the train speed profiles in order to include the consequences of conflicts due to constraints imposed by the signaling system. In this latter case, CR and TSC cooperate in computing a dispatching solution by adopting a feedback loop between the two systems. In fact, Goh and Mees (1991) have proved that solving CR and TSC as an overall problem is extremely difficult when dealing with large rail networks.

In the following subsections, we present a description of some of the most relevant research contributions on the development of an automated system for real-time train dispatching, divided in fixed-speed and variable-speed models (as shown in the outline of Figure 2.5). However, this survey does not explicitly review the performance of industrial tools and simulation tools used by railway authorities and railway management companies due to the lack of available publications.

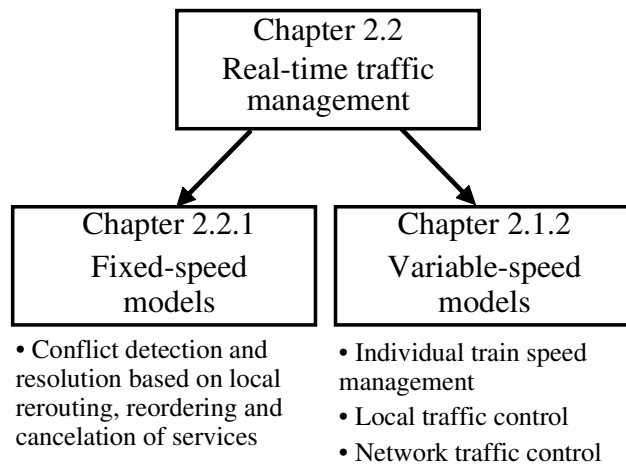


Figure 2.5: Road map of the real-time traffic management section

In this thesis, we also do not review CR problems in other transportation fields. For example, the air traffic management is similar to rail traffic management in the sense that the main objective is to minimize the passengers' disadvantages during operations. In case of small delays, the rescheduling techniques are compatible even if these are applied in a two level stage: first in the airport routes and then in the routes to be flown. This means that air traffic can be rerouted in the airspace according to a multi-dimensional set of tracks, leaving more space for optimization. However, in case of large disruptions the possibilities to modify the timetable of airlines are very limited and correspond to late departures or to flight cancelations. The main problem is the very high competition between flight companies to keep the level of service satisfactory and economical factors could influence strongly the resolution of the CR process. Moreover, the cancelation of connected services is a very critical factor for airlines companies, especially in case of intercontinental flights. A detailed description of air traffic control during irregular operations can be found e.g. in Ball et al. (2007).

2.2.1 Fixed-speed models

The aim of real-time conflict detection and resolution models is to minimize train delays or deviations from the planned schedule while satisfying a set of operational constraints. In fact, the problem complexity is similar to operational scheduling. However, in real-time train scheduling the computation time is really an important factor and the determination of train priorities is more complex since this depends on many dynamic factors, such as

platform capacity and the tardiness of trains. For these reasons, the real-time resolution of the problem is relatively complicated and has been approached in literature only recently.

Because the meeting and passing of trains are intimately related to their operating speed, a complete model would need to treat velocity as a decision variable. However, most existing dispatching models use a sequential approach and assume a fixed speed profile for each running train that will operate at maximum velocity whenever possible. In these cases, the feasibility of the different speed profiles is later checked for each train. Next, we give a review of recent approaches designed to resolve train conflicts during operations.

The main goal of train rescheduling strategies is to reduce delay propagation by changing the precedence between trains at crossing and merging points. In other words, the train rescheduling process requires determining a new feasible plan of meets and overtakes such that the new schedule is deadlock-free, there are no more conflicts between trains and the delay at all the stations and relevant infrastructure locations is minimized. In fact, the problem of deciding whether a deadlock-free schedule exists or not, being fixed the speed profiles and routes of the trains, is an NP-complete problem (Garey & Johnson, 1979), (Ullman, 1975). Furthermore, within a real-time environment it is necessary to solve the problem under severe time requirements.

İ. Şahin (1999) studies the real-time conflict resolution problem of a single track railway. Conflicts between trains are solved in the order they appear. An algorithm based on look-ahead measures detects potential knock-on delays and takes ordering decisions at merging or crossing points. The problem is formulated similarly to a job shop scheduling problem and the objective is to minimize the average knock-on delays.

Adenso-Díaz et al. (1999) consider the problem of managing real-time timetable perturbations in a regional dispatching area. A mixed integer programming model is adopted to maximize the number of passengers transported and backtracking heuristic techniques are developed to solve train conflicts. A conflict resolution system using their algorithms has been implemented to support dispatchers at a traffic control center of the Spanish national railway company.

Ping et al. (2001) propose an optimization method based on genetic algorithms for adjusting the orders and times of trains on a double-track line. The objective is the minimization of a total delay while the parameters used are the train departure orders in each station. Simulations are presented on the Guangzhou-Shenzhen high-speed railway link.

Jacobs (2004) has developed a detailed model based on the identification of possible route conflicts with high accuracy, using blocking time theory as described e.g. in Pachl (2002), when the objective is to minimize additional running times. In the presence of disturbances, an algorithm detects the infeasible train routes and solves each conflict locally postponing the trains with less priority.

Törnquist and Persson (2005) evaluate different rescheduling alternatives for minimizing the total delay in case of real-time disturbances. The problem is solved using an iterative two-level process. The upper level handles the order of meets and overtakes of trains on

the track sections, using simulated annealing and tabu search, while the lower level determines the start and end times for each train and the sections it will occupy. Experiments on passenger trains running on a Swedish railway are presented to compare the proposed approaches with the optimal solutions.

In case of complex railway networks, rescheduling trains can be very complex and time consuming. The problem is solved by local reordering trains to obtain good quality solutions. However, we believe that there is room to approach the problem from a global resolution perspective, which means rescheduling trains while considering the propagation of each local ordering decision to the entire period of traffic prediction. This may lead to new models for real-time traffic management of general railway networks, enabling the computation of near-optimal solutions while respecting the required practical constraints.

Sophisticated decision support tools based on mathematical programming techniques are needed to help dispatchers to control the railway traffic under severe disturbances, i.e., large entrance delays, blockage of some tracks and failures of rolling stock. To this end, centralized decision-making focuses on the point-to-point routing and scheduling of trains in a railway network by reoptimizing the use of infrastructure capacity and minimizing the propagation of train delays. A point-to-point train refers to freight or passenger trains wherein the origin (A) and destination (B) stations are fixed a priori but local routes and meet-pass decisions between A and B may be ascertained dynamically. This corresponds to a compound rescheduling and local rerouting problem which is very difficult to solve in real-time for large networks and only some heuristics have been proposed, so far.

The general railway traffic management problem can be formulated as a job shop scheduling problem with several additional constraints (Szpigel, 1973). In this context, Chiang and Hau (1995) propose a two-steps repairing heuristic that starts with an initial schedule, not necessarily conflict-free, and searches for possible improvements. In a first step, a route preprocessing algorithm based on local search techniques is developed to determine a promising initial track assignment for each train. A second step is applied to compute a conflict-free schedule and optimal train routes within stations based on simulated annealing and tabu search techniques. In both steps, the search of conflict-free routes is guided by an earliest-conflict-first heuristic that attempts to repair the earliest headway constraint violation while minimizing the total running time and the initial time deviation of each train. Numerical experiments on randomly generated instances with 100 trains and 10 stations are presented to underline the potential of the preprocessing step and the performance of the proposed algorithms in terms of number of iterations and average total costs.

A similar research line focuses on the alternative graph model that is a generalization of the job shop scheduling problem with blocking (or no-store) constraints (Mascis & Pacciarelli, 2002). A first alternative graph formulation of the train scheduling problem with fixed routes is developed by Mascis et al. (2001). In Mascis et al. (2002), rerouting measures are chosen on the basis of priority rules while ordering decisions are taken by using a greedy heuristic based on the alternative graph with the objective to minimize a maximum delay. Flamini and Pacciarelli (2007) address the problem of routing trains through an

underground rail terminus and develop a heuristic algorithm for a bicriteria version of the problem in which earliness/tardiness and time headways have to be optimized.

Törnquist and Persson (2007) introduce a model for dispatching trains in a railway network with several merging and crossing points. The problem is formulated as a MILP and solved with commercial software packages. Different dispatching strategies are proposed to reduce the search space based on restrictions on reordering and local rerouting actions. Computational experiments based on instances with a single delayed train show that performing a number of order swaps for specific segments enables the computation of good quality solutions. On the other hand, allowing all possible changes of train orders and routes requires a too long computation time. Törnquist (2007) presents an improved version of the former dispatching heuristics. A larger number of disturbed trains is considered for reordering and rerouting with the aim of reducing the delay propagation. Experiments are presented for various disturbance settings on a large part of the Swedish railway network and for an hour of traffic prediction.

Rodriguez (2007a) focuses on the real-time CR problem through junctions and proposes a computerized routing and scheduling system based on constraint programming. The experiments show that a truncated branch and bound algorithm can find satisfactory solutions for a junction within a computation time compatible with real-time purposes. Rodriguez (2007b) extends the model to cope with large stations by considering conflicts between trains running in opposite directions. The tracks which are shared between trains running in two directions are modeled with state resources.

It is of interest to highlight the potential use of rerouting instead of only rescheduling the trains in order to limit the delay propagation. Moreover, practical real-time problems involve many conflicting trains, due e.g. to the presence of several delays and blocked tracks. This requires implementing extensive local rerouting actions and reordering trains at merging and crossing points. However, due to the significant complexity of the compound rerouting and rescheduling problem, even recent approaches are able to provide good quality solutions only for small instances or for simple perturbations.

Train traffic can be seriously disrupted when accidents, natural disasters or some technical problems occur on railway lines. In some cases, rail service is seriously disrupted and traffic controllers must prepare an adequate plan by making a series of strong modifications to the current train schedules, so that an emergency timetable has to be adopted. That is, a drastic dispatching solution would be needed such as the cancelation of some train lines and/or scheduled connections in order to satisfy the requirements of at least some of the train operating companies. Considering these effects, a dispatcher must evaluate possible kinds of conflict resolution and select the best solution possible. When a time critical conflict arises at short notice this task requires a decision support system.

Fay (2000) describes an expert system using Petri Nets and suggests a fuzzy rule-base for train traffic control during disturbances. A number of experiments are performed on fictional data with some trains and one station. In case of strong disorder, an adopted control strategy is the cancelation of train paths. Zhu (2001) introduces a simulation model based on

stochastic Petri nets in order to evaluate the performability of a railway system and to assess the impact of incidents on the quality of operations expressed by the distribution of initial and consecutive delays. The latter approach focuses on how to determine train traffic delays caused by primary stochastic disturbances, especially technical failures. Computation times for larger networks still seem to be a problem.

Wegele and Schnieder (2005) use software packages with genetic algorithms to reschedule trains with the objective of minimizing passenger annoyance, e.g. delays, change of platform stops and missed connections. The adopted dispatching strategies are dwell time modifications and local rerouting of trains inside a station. Examples of application on a part of the German railways are reported. Particularly, two disturbed situations are described: the insertion of a new train line and a set of stochastic disturbances. Their algorithms take a few minutes of computation but no comparison with other optimization techniques or dispatching rules is presented to establish the solution quality of the proposed methodology. Recently, the proposed algorithms have been inserted in a dispatching support system with a graphical user interface (Wegele et al., 2007). A test case with one disturbed train is reported to show in detail how the dispatcher should interact with the system.

Armstrong and McDonald (2005) state that railway operations are particularly vulnerable to disruptions, leading to large delays or even cancelation of services. Their paper presents which control actions are desirable to respond to disruptive events on Britain's mainline railways, and reflect the priorities of different passenger and freight operators. They suggest to assign penalties to non-scheduled waiting time and journey time, and use this information to calculate optimal train orders and times in case of conflicting routes.

Sakowitz and Wendler (2006) present a database management system to generate economically evaluated train priorities for conflict situations on the basis of active, deductive and normative rules. Such a system would allow new, deducible facts to be specified, administered and especially derived from explicitly introduced facts. That is, priorities between running trains and scheduled connections at stations are generated and assigned for lightly and heavily disrupted railway traffic respectively.

Hirai et al. (2006) propose an algorithm for train dispatching with a train rescheduling pattern language processing system. In case of severe train traffic disruptions, caused by accidents that may require the suspension of some train line, the algorithm is helpful for the preparation of practical rescheduling plans. Applying actual train schedule data, the authors claim that their approach works satisfactorily. However, they apply local modifications of the original schedule that result in sub-optimal rescheduling decisions.

Jespersen-Groth et al. (2007) present an operational system to decide when the reinsertion of train lines, in the suburban rail network of Copenhagen, shall start on each rolling stock depot to return to regular service. The authors propose a mixed integer programming model for finding a heuristic reinsertion plan minimizing the latest completion time. A strong operational constraint is not to alter the order in which the trains are inserted within a depot and between depots. The solutions generated with the reinsertion model outperform the rolling stock dispatching rules previously adopted by operators.

In the practice of rail operations, when the timetable is cyclical and larger disturbances occur an adopted countermeasure is to cancel connected train services or even entire train lines. However, a quantification of the effects of such dispatching measures is still lacking and it is not clear whether the use of such strong countermeasures can improve the quality of railway service.

2.2.2 Variable-speed models

Variable-speed models include the re-planning of train movements according to the current state of the signaling system and other traffic regulations, when service anomalies and simple perturbations occur. This consists in timing all movements of trains along a line or in stations and setting the commands for modifying the actual speed profile. In literature, only few studies are known which take into account the possibility of speed coordination to improve the rescheduling solution. This would be achieved by advanced traffic management systems which are able to consider a complicated network for the CR system and improve the obtained solution by speed optimization afterwards. Among recent contributions related to the real-time dispatching problem with variable-speed profiles, we consider the approaches based on optimization of each individual train speed, then we review the recent contributions on optimal control of rail networks at a local and global level.

Individual train speed management

Many knock-on delays can be prevented if traffic is timely managed. Based on an accurate monitoring, train positions and speeds could be predicted in advance and resolved in real-time (Hansen, 2001). The adjusted targets (location-time-speed) have then to be communicated to the relevant trains. In this way, informed train drivers (see Figure 2.6) would prevent delays by suitable adjustments of train speed profiles when the movement authority and the signal aspects change, i.e., they may decelerate the train in advance with the aim of avoiding complete stops on open tracks.

Driver Assistance Systems installed on-board have to consider the constraints of the train protection system to avoid undesired automatic braking, consequent slow downs and extra energy consumption (see, e.g. Schutte (2001), Matsumoto (2005), Stadlmann (2006) and Yasui (2006)). They would suggest an individual train speed management allowing a more precise control of trains and effective train speed coordination on open tracks, securing time windows at junctions/crossings, or synchronizing arriving trains at stations in case of delays and expected route conflicts.

Howlett et al. (1994) study the applicability of optimal control theory for the control of trains in normal traffic conditions. Train movements are performed under a variety of realistic operational constraints to reduce energy or fuel costs. For urban and suburban passenger rail services, a traffic control system is presented to dynamically adjust distance headways between consecutive trains, corresponding to determine energy-efficient driving strategies.



Figure 2.6: A train driver in action

Khmelnitsky (2000) investigates the problem of minimizing energy consumption for moving a train along a given route for a given time. The objective is to determine optimal train operation for traction and brake applications subject to arbitrary variable grade profile and speed restrictions. This problem is solved by constructing a numerical algorithm which exploits analytical properties of the optimal solution obtained from the maximum principle analysis. The algorithm could be incorporated in a decision support system for train drivers in order to recalculate the optimal velocity profile in case of perturbed operations.

Franke et al. (2000) also adopt the concept of energy as state variable and position as independent variable but, on the contrary to the latter approach, they use a discrete dynamic programming algorithm to optimize the driving of trains. Simulation results on individual trains running in a corridor show promising energy savings of the proposed algorithm compared to skilled train drivers. Moreover, the authors state explicitly that time restrictions due to interactions between trains, caused e.g. by time delays or temporary speed limits, can be taken into account, although they did not perform this kind of experiments.

Albrecht and Oettich (2002) present a new approach to fulfil the conflicting goals of dynamic schedule synchronization and energy saving in rapid rail transit systems. They present an algorithm for the dynamic modification of train running times in such a way that the probability of arriving on-time to transfer to other means of public transport can be increased and the overall energy consumption of train operation is minimal. The optimal train timetable can be computed in real-time using dynamic programming, while energy consumption and non-scheduled waiting time due to missing a connection form the bicriteria optimization function. Albrecht (2005b) and Albrecht and Van Luipen (2006) provide a new simulation tool for driver-assistance which enables an early recognition of conflicts based on the actual state of the signaling system and the manual control of the train speed.

Asuka and Komaya (2003) mainly focus on urban rapid transit with low speeds and short distances between stations. Their paper presents an advanced traffic management system

which is able to consider a complicated network for the CR system and to compute optimal speed profiles for the involved trains. The key idea is to distribute trains regularly in the network by modifying train speeds well in advance before each detected conflict and generating only a minimal delay after the signaling system upgrades.

Pacciarelli and Pranzo (2006) propose a TSC system that takes as input the schedule produced by a CR system and computes feasible time windows for the circulation of each train on each block section. This speed control is able to suggest speed targets that satisfy the traffic regulations, improve train punctuality and save energy. The advisory speeds have then to be transmitted to train drivers.

Local traffic control

Traffic conflicts at railway junctions are very common, particularly on congested rail lines. While safe passage through the junction is guaranteed by the signaling and interlocking systems, a local traffic control minimizes the delays imposed on the trains by assigning the optimal sequence. This is a challenge for improving the quality of rail service.

Komaya and Fukuda (1991) provide a new architecture for knowledge-based integration of simulation and rescheduling. Strategic knowledge is used to emulate the reasoning processes by performing partial simulation and local traffic control repeatedly in a specific order. The order of trains is decided on a segment between two ordered stations using an expert system with the objective to minimize the total accumulated delay for all trains.

Ho et al. (1997) describe a traffic controller which minimizes a total weighted delay due to train conflicts at a railway junction by means of dynamic programming. This approach results in a local optimization of the train reordering procedure. Ho and Yeung (2001) use a deterministic method to resolve the train conflicts that appear in real-time by means of genetic algorithms, tabu search and simulated annealing.

Huerlimann (2001) focuses on speed optimization at a single junction. Anticipating train control would increase track capacity and reduce energy costs and the avoidance of unplanned stops. These effects are obtained by anticipatively slowing down the train before the critical conflict and then passing the critical infrastructure element with the smallest possible delay.

Albrecht (2005a) and Albrecht (2007) analyze the consequences of anticipating train control, i.e., slowing down a train before a possible conflict in order to avoid unplanned stops and reduce its consequences (high delay, high energy consumption, low comfort for passengers). In these two papers, a straight concept for the minimization of time losses is proposed whose integration into dispatching algorithms promises better computation of running times.

Tazoniero et al. (2005) describe a knowledge-based system that uses fuzzy rules to schedule the train circulation in freight railroads as close as possible to reference trajectories. The system can easily include dispatching rules and computes feasible circulation plans for a

general railway network very quickly. The simulation results on a Brazilian railway line with three trains running in each direction show that the solutions obtained by a MILP model would perform better than fuzzy decision-making solutions in terms of total train circulation delay.

Takagi et al. (2006) perform numerical optimization on a rail junction when the service is disrupted. The program uses a traffic management simulator, called object-oriented multi-train simulator. In the optimization routine, a genetic algorithm is used to improve the order of route setting, i.e., to minimize a weighted sum of train times. When a route setting decision has been taken, the signaling system and the electric power supply system are automatically set. The proposed simulator was originally designed and tested for metro lines and suburban lines.

Network traffic control

On a main line railway network with many junctions, the delay of a train is likely to cause delays to many other trains, especially because of conflicts at junctions. Optimizing one junction, however, may have an adverse effect on other parts of the rail network because of the mixed traffic situation of most main line railways. A network traffic control is therefore required that manages the train traffic in a larger railway area (Makkinga & Zigterman, 2002). The major difficulty of a network traffic control approach is that the execution time and the memory requirements increase non-linearly as the system grows in size with more trains, tracks, and stations. A time-efficient solution is handling larger numbers of trains on the basis of a distributed resolution of potential train conflicts, i.e., distributed traffic control. However, distributed control of railway traffic needs a coordination framework enabling to verify the feasibility of the proposed solution and to estimate the propagation of delays at network level.

Vernazza and Zunino (1990) propose an approach to solve conflicts locally by enabling a negotiation between the involved trains and the local infrastructure administrator. The control problem is modeled in terms of resource allocation tasks and priority rules are adopted for each local control decisions. The system has been implemented by simulating a real network and some train conflicts at rail junctions have been tested.

Iyer and Gosh (1995) introduce a new decision process for every train executed by an on-board processor that claims the setup of train routes, dynamically and progressively, through explicit processor to processor communication primitives. Each train negotiates to get access to block sections while minimizing its total travel time. The decision process of each station is executed by a dedicated processor that, in addition, maintains absolute control over a given set of tracks and participates in the negotiation with the trains. Their experiments show that the computation time increases nonlinearly with the number of trains and block sections. Successively, Lee and Gosh (2001) report on a decentralized train scheduling algorithm to control large networks within a short computation time. They study the system performance by simulating part of the eastern U.S.A. railroad network. The obtained results indicate that the proposed algorithm is stable with respect to input traffic rate

perturbations of finite durations, depending on the size and the degree of capacity use, and unstable under permanent track blockage and communications link failures.

Lamma et al. (1997) present a distributed advisory system which helps human traffic controllers in the management and control of traffic within railway stations and along railway branches. The scheduling of trains along a railway line is performed by several modules each one controlling a limited number of block sections and applies conflict resolution rules based on local priority minimizing train delays.

Missikoff (1997) proposes an object-oriented analysis and design of the conflict detection and resolution problem, knowledge-based application modeling and advanced search techniques paying attention on solving train conflicts based on local and approximated costs.

De Schutter et al. (2002) and De Schutter and Van Den Boom (2002) use the Model Predictive Control (MPC) for traffic coordination in railway systems. They state that the MPC design problem for railway systems leads to a non-linear and non-convex optimization problem solvable as an extended linear complementarity problem. The proposed control would be able to recover from delays in an optimal way while breaking connections and letting some trains run faster than the planned speed profile. In Van Den Boom and De Schutter (2007), the authors present a model predictive control method for dynamic traffic management of railway networks and consider the change of the departure times of trains. A switching max-plus-linear system is proposed, which has a structure similar to a mixed integer linear programming problem. A solution for the Dutch railway network would be computed using existing solvers for this problem, or with a genetic algorithm or tabu search techniques.

Mazzarello and Ottaviani (2007) describe the development and implementation of a traffic control system, developed in the European project COMBINE. The aim is to test the feasibility of a completely automated system for conflict detection, resolution and subsequent speed regulation. A coordination level for traffic control along a railway line is tested on a pilot site in the Netherlands. An in-depth description of the models and algorithms proposed in COMBINE can be found in Mascis et al. (2002), Mascis et al. (2004) and Pacciarelli and Pranzo (2006).

Chou et al. (2007) present a distributed control system based on several railway areas that are mutually influenced. A novel time-shift coordination strategy between neighboring traffic control areas is proposed to obtain collaborative rescheduling. Various policies for the distributed control regions are studied in order to evaluate the delay cost of adopting distributed control in neighboring regions.

Furthermore, Davidsson et al. (2005), Negenborn et al. (2006), D'Ariano and Hemelrijk (2006), Mes et al. (2007), Salido et al. (2007), Boudali et al. (2007), Negenborn (2007) and Katwijk (2008) provide some contribution and literature review of existing research on agent-based approaches to transportation and traffic management. A general conclusion is that agent-based approaches seem very suitable for train traffic control, but their applicability still needs to be verified by more advanced decentralized or distributed systems.

In the current literature on train traffic control, there is a lack of attention to the precision on simulating traffic flows. Basic requirements consist of the identification of possible route conflicts with high accuracy and the dynamic estimation of the impact of different dispatching alternatives. By using detailed information on the actual network status, accurate running times can be calculated and the effective remaining distances to conflict points and braking curves can be estimated. The predefined minimal headway and transfer time constraints applied in combinatorial models do not guarantee their practical feasibility in complex junctions and need to be proven by a detailed analysis of blocking times. A prerequisite of any conflict resolution measure is a reliable and exact real-time prediction of running times and blocking times of the involved trains based on actual data concerning train weight, length, acceleration, deceleration and even weather (strong wind, rainfall, ice, etc.). Moreover, there is no clear modeling of the signaling system in case of a conflict, i.e., the involved trains may have short distance headways, the signaling system changes status and an acceptable speed profile for each hindered train at the conflict point must be recomputed. This means the specific technical characteristics of the rolling stock assigned to each train path, the loading weight and the available friction rates of the rails along the line should be known or estimated in order to enable a precise recalculation of the required minimal running time and the effective remaining distance until the next conflict points. Therefore, for complex network problems and increasing values of disturbance, the successive approximations on estimating the travel time of circulating trains can accumulate errors and worsen the quality of the traffic prediction.

2.3 Dynamic traffic management

The standard practice in railway traffic management consists of the off-line design and construction of a conflict-free timetable and the real-time control of trains based on a schedule planned in advance with set times. As a result, the railway system is not very flexible. In fact, when train operations are perturbed conflicts arise with respect to the scheduled train paths. Perturbed operations cause delays and, at least in congested areas, this requires frequent real-time timetable modifications. In the previous sections, we have described models and algorithms for off-line and real-time traffic management. Table 2.1 summarizes the similarities and differences between the two approaches.

Trains behave in real-time differently from what is planned in the timetable. When the scheduled railway traffic is disturbed, decisions have to be taken that modify the plan of operations in order to reduce delay propagation. Schaafsma (2001) proposes a new concept of *Railway Dynamic Traffic Management* (RDTM) for improving railway system robustness (i.e., the resilience to disturbance in operation), without decreasing the capacity of the lines. The basic idea is to keep train traffic flowing in the bottleneck by avoiding unnecessary waiting time. This can be achieved by relaxing some of the timetable specifications, such as train routing, arrival/departure times and sequencing. This concept has been developed within the Dutch railway undertakings in the last years (see, e.g. Middelkoop and Hemelrijk (2004), Schaafsma (2005), Van Den Top (2005), Middelkoop and Loeve

Factors of comparison	Off-line timetabling	Real-time traffic management
Main objective	Design optimal schedule	Implement optimal control
Schedule validity	Up to some years	Up to few perturbed hours
Degree of flexibility	Any change applicable	Minor timetable modifications
Traffic conditions	Usually ideal situation	Perturbations or disruptions
Time span of prediction	Long time horizon	Up to some hours
Space span of prediction	Large traffic network	Rail junction or small network
Computation time	Up to several months	Up to few minutes

Table 2.1: Off-line timetabling versus real-time traffic management

(2006), Schaafsma and Bartholomeus (2007)) resulting in a number of principles for the management of congested areas by planning less in the off-line phase, and by postponing the resolution of possible conflicts among trains to the real-time traffic management. The RDTM principles consist basically of the following dynamic strategies:

1. Strict arrival/departure times are replaced by time windows of [minimum, maximum] arrival/departure times at each platform and relevant timetable points of the network. A large time window corresponds to having more flexibility. In this case, the operational timetable, used by railway managers, includes both minimum and maximum arrival and departure times, while the public timetable, available to passengers, includes the maximum arrival time and the minimum departure time only. The longer travel times would be compensated by a higher reliability of train services, i.e., travel times and connections. This would allow a greater possibility of control traffic.
2. The scheduled order of trains at overtakes and junctions may be provisionally, or even partially defined in the operational timetable and finally determined in real-time. In the latter case, the timetable might contain conflicts to be solved during operations. Enabling the change of order in real-time would allow to reduce delay propagation.
3. The default platform/passing track for a train at a station is replaced by a set of feasible platform/passing tracks, leaving the final choice to traffic control. In this case, an additional dynamic information system would guide the passengers to their trains. The operational timetable might also specify a set of routing options for each train that would offer a possibility to improve the use of infrastructure capacity.

This information should be partially defined in the off-line timetable and then fixed in real-time, based on the actual status of the network and on the current train positions. When an increasing degree of freedom is left to real-time control, traffic control requires several real-time actions and a computerized traffic management system is necessary to support human dispatchers to perform a better recovery of delays during operations. Traffic controllers need real-time traffic management systems either to improve the effectiveness of the dispatching decisions and/or to automatically re-regulate the speed and routes of the

involved trains. RDTM distinguishes among local simple rule based systems and advanced traffic management systems. A local system acts only “on the spot” and “now” and may reflect simple dispatching rules. Advanced traffic management systems would take into account the whole traffic in a larger area, predict conflicting train movements (that have direct impact on the level of punctuality), calculate automatically optimal traffic flow and advice a change of order or route to the dispatcher, as well as display advisory speeds to train drivers.

Middelkoop and Hemelrijk (2004) investigate the possible effects of introducing the RDTM principles in real-time traffic management by using the macroscopic simulation tool SIMONE. Successively, Middelkoop and Loeve (2006) report on a computational analysis using the microscopic simulation tool FRISO. Recently, Schaafsma and Bartholomeus (2007) describe the first implementation of some of the RDTM principles at the Schiphol bottleneck of the Dutch railway network. In the latter paper, the authors limit the assessment to routing flexibility and to train resequencing using the first come first served rule. In fact, up to now the three principles of dynamic traffic management have been evaluated separately and on a single case study. Hence, there is a need to test the three principles simultaneously on different networks and timetables to assess the potential and limitations of these principles more in general. Furthermore, the proposed RDTM principles have been applied by using simple dispatching procedures based on local information while their effects should be evaluated in combination with advanced traffic management systems, enabling precise estimations of traffic flows and the computation of optimal dispatching solutions.

2.4 Short discussion

This chapter presents a variety of optimization models to solve off-line and real-time traffic management problems in rail transportation. Application of railway operations research tools and methodologies may enable the resolution of difficult real-world combinatorial NP-hard problems by exploiting the structure and peculiarities of each problem. Recently, traffic management models are solved with very sophisticated mathematical programming techniques, others still are solved using heuristics that have proven to be very effective for several classes of discrete optimization problems. Of course, this progression is also made possible by the increased power of computers and information systems.

Timetabling has traditionally been a very active area of rail research. However, it is possible that past research emphasis on finding an optimal plan has been misplaced. By definition an optimal solution is the best feasible solution, so any plan must be at least feasible to have any hope of successful implementation. The algorithms developed so far rely on simplifications of the modeling of train conflicts recognition and resolution between different train paths and routes. So, a feasibility check, proving a more reliable service at the network level, is needed to identify and resolve resource conflicts within the timetable design process.

During the planning phase, several negotiations between the authority and train operators are performed to solve train conflicts and the railway traffic is managed as a carefully

planned schedule. In real-time, timetable perturbations often arise in highly dense networks and dispatchers have several train ordering and routing alternatives, but they do not have enough time to evaluate different dispatching measures in order to improve the original plan. Therefore, operational rules are often set as off-line agreements.

Recent applications demonstrate an increasing effort to deal with complex yet relevant characteristics of the actual dynamic processing of trains in railway networks. As a result, real-time traffic management problems in complex networks which, in the past, were only solvable by simulation are becoming tractable by mathematical optimization methods. Nevertheless, simulation techniques have also made considerable progress in the last decade and remain a very useful tool for analysis and support to decision making. However, the real-time algorithms developed so far to manage train traffic efficiently are applicable only off-line for small instances, or simple disturbances, or generate solutions far from optimal. Even though most dispatching models are tested on realistic data instances, very few have been implemented and used during practical rail operations. Hence, railway operations research must be directed to bridge the gap between theory and practice.

Railway dynamic traffic management offers an interesting possibility to improve railway services by operating flexible timetables in which a train has to fit in a time window of arrival at a given set of feasible platform/passing tracks. As the planning is less detailed, traffic control thus requires more actions and the development of advanced control systems. Such systems should regulate train movements with the objective of minimizing the propagation of delays while assuming the standards of safety and security. Moreover, a real-time decision tool is a critical requirement because unexpected events require train schedules to be constantly updated during operations. Even a perfect operational schedule needs to support parameter variation and other dynamic features in order to aid effectively traffic operators in their task. To conclude, a real-time decision support system to automatically evaluate the impact of different traffic management measures in case of technical failures which interrupt the operation of trains on a track or in case of train delays which lead to timetable and route conflicts is still lacking.

Chapter 3

Real-time train dispatching approach

Regulation of railway traffic aims at ensuring safe, seamless and as much as possible punctual and energy-efficient train operations. Due to the strict time limits for computing a new timetable in the presence of disturbances, train dispatchers usually perform manually only a few modifications, while the efficiency of the chosen measures is often unknown. Some computerized dispatching support systems have been developed, so far, which can provide fairly good solutions for small instances and simple perturbations. However, this kind of systems cannot deal with heavy disturbances in larger networks as the actual train delay propagation is simply extrapolated and does insufficiently take into account the train dynamics and signaling constraints. Therefore, extensive control actions are necessary to obtain globally feasible solutions.

A basic requirement for regulating train traffic flows is an identification of route conflicts with a high level of accuracy. While most existing studies concentrate on conflict detection and resolution, but disregard the dynamic train characteristics, we introduce a variable-speed dispatching system which can control the railway traffic in a regional network more realistically. Acceleration and deceleration times are explicitly modeled according to the constraints of the signaling system and rolling stock characteristics. Our research effort is focused on the real-time optimization of train scheduling, routing and speed coordination.

This chapter describes the implementation of our real-time train dispatching system, called ROMA (Railway traffic Optimization by Means of Alternative graphs), that is designed to support dispatchers in the everyday task of managing disturbances. Particularly, the ROMA dispatching support system is implemented in C++ language, is compatible with Linux and Windows platforms and uses the *AGLibrary* developed by the “Aut.Or.I.” Research Group of Roma Tre University. Given a disturbed timetable, the real-time train dispatching problem is divided into four subproblems: (i) data loading and exchange of information with the field, (ii) assigning a passable route to each train in order to avoid blocked tracks, (iii) defining optimal train routes, orders and specifying the exact arrival and departure times at stations as well as at a set of relevant points in the network, such as junctions and passing points, and (iv) ensuring a minimum distance headway between trains while maintaining acceptable speed profiles. ROMA is able to address the resolution of the four subproblems.

This chapter is organized as follows: Section 3.1 describes formally the train dispatching problem and Section 3.2 deals with the ROMA architecture and how the four subproblems are solved. Concluding remarks indicate the implementation status of the system.

3.1 Problem description

In its basic form a railway network is composed of stations, links and block sections separated by signals. For safety reasons, the signals control the train traffic on the routes, and impose a minimum distance headway between consecutive trains. Signals, interlocking and Automatic Train Protection (ATP) control the train traffic by imposing a minimum safety separation between trains, setting up conflict-free routes and enforcing speed restrictions on running trains. The minimum safety distance and time headways depend on the speeds of the consecutive trains, the braking rate of the second train, the train length of the first train and the signal spacing. In case of technical or human failures, ATP ensures safe rail operations. In particular, ATP causes automatic braking if the train ignores the valid speed restrictions. Signals are located before every junction as well as along the lines and inside the stations. A block section (in a fixed block system) is a track segment between two main signals, that governs the train movements, and may host at most one train at a time.



Figure 3.1: A train facing a red signal aspect while waiting for the passage of a preceding train

The standard feature of a railway signaling system is characterized by the three-aspect fixed block signaling. For fixed block signaling systems, a train may enter a block section only after the train ahead has completely cleared the block section and is protected by a stop signal. A signal aspect may be red, yellow or green. A red signal aspect means that the

subsequent block section is either out of service or occupied by another train (see e.g. Figure 3.1), a yellow signal aspect means that the subsequent block section is empty, but the following block section is still occupied by another train, and a green signal aspect indicates that the next two block sections are empty. A train is allowed to enter the next block section if the signal aspect is either green or yellow, but the latter requires deceleration and stop before the next signal if this remains red. A detailed description of different aspects of railway signaling systems and traffic control regulations can be found e.g. in Pachl (2002), Goverde (2005) and Goddard (2006).

The passage of a train through a particular block section is called an *operation*. A *route* of a train is a sequence of operations to be processed in a track yard during a *service* (train run). At any time a route is *passable* if all its block sections are available and the corresponding block signal is green or yellow, i.e., there are no blocked tracks. The *timing* of a route specifies the starting time t_i of each operation in the route. Each operation requires a traveling time, called *running time*, which depends on the actual speed profile followed by the train while traversing the block section. A speed profile is furthermore constrained by the rolling stock characteristics (maximum speed, acceleration and braking rates), physical infrastructure characteristics (maximum allowed speed and signaling system) and driver behavior (coasting, braking and acceleration profiles when approaching variable signals aspects, see Chapter 7). The running time includes the time needed to accelerate (or decelerate) a train due to a scheduled stop, as well as speed variations between two consecutive speed signs. Furthermore, the running time is known in advance since all trains travel at their scheduled speed, which usually contains some margins for recovery. In yards, the routes for the individual trains need to be setup before entering and cleared after leaving, which takes a certain switching time.

A delay may occur when a train reaches the end of a block section and the subsequent block section is still occupied by another train. The running time of a train on a block section starts when its head (the first axle) enters the block section. Safety regulations impose a minimum distance separation among the trains, which translates into a minimum *setup time* (time headway) between the exit of a train from a block section and the entrance of the subsequent train into the same block section. This time takes into account the time between the entrance of the train head in a block section and the exit of its tail (the last axle) from the previous one, plus additional time margins to release the occupied route and sighting distance (as described e.g. in Nie and Hansen (2005)). Railway timetable design usually includes *recovery times* and *buffer times* between the train routes: recovery times can be utilized to recover from delays by running at maximum speed and shorten the scheduled train stops to a minimum dwell time, whereas buffer times prevent or reduce the delay propagation to other trains.

We consider a cyclic timetable which describes the movement of all trains circulating in the network during subsequent time periods, specifying, for each train, the planned arrival/passing times at a set of relevant points along its route (e.g. stations, junctions, and the exit point of the network). At stations, a train is not allowed to depart from a platform stop before its scheduled departure time and is considered late if arriving at the platform

later than its scheduled arrival time. At a platform stop, the scheduled stopping time of each train is called *dwell time*. Additional practical constraints related to passenger satisfaction are to be taken into account, such as minimum transfer times between connected train services. This is the time required to allow passengers alight from one train, move to another platform track and board the other train. Constraints due to rolling stock circulation must also be taken into account. In fact, a train completes a number of round-trips during the service of a line and its length may be changed by (de)coupling. For this reason, railway timetables include a *turn-around time* at terminal stations, which is a time margin between the arrival of the train and the start of a new train service in the opposite direction using the same rolling stock. In case of severe disturbances a scheduled train may be delayed from the beginning because of the unavailability of rolling stock or train personnel.

Timetables are designed to satisfy all traffic regulations. However, during operations, unexpected events occur that can make the original timetable infeasible. Unscheduled braking and stopping of trains increase the running time and cause delays. The delay may propagate causing a domino effect of increasing disturbances. We define an *entrance perturbation* as a set of delayed trains when entering a dispatching area to be controlled and a *timetable disruption* as the modification of scheduled train speeds and routes, and dwell times relative to the considered dispatching area. Specifically, *entrance delays* are due to the propagation of delays from previous dispatching areas. Running time prolongation may occur because of headway conflicts between consecutive trains or technical failures, route changes are due to some block section being unavailable for a certain amount of time and dwell time perturbations are due to traffic delays at stations.

Real-time train dispatching must cope with momentaneous infeasibility by adjusting the timetable of each train, in terms of routing and timing, and/or by resequencing the trains at the entrance of each merging/crossing point. The railway traffic is predicted over a given time horizon, i.e., a given number of timetable hours. The main goal of dispatching is to minimize train delays while satisfying traffic regulation constraints and guaranteeing the compatibility with the actual position of each train. The latter information enables the computation of the *release time* of each train at the starting time t_0 of traffic prediction, which is the expected time of which each train clears the end of its first block section in the area under study. The *total delay* is the difference between the estimated train arrival time and the scheduled time at a relevant point in the network, and can be divided into two parts. The *initial delay* (primary delay) is caused by original failures and disturbances and can only be recovered by exploiting available running time reserves, i.e., trains traveling at maximum speed. The *knock-on delays* (consecutive or secondary delays) are caused by the hinder from other trains or dispatching measures like late setting-up of train routes.

A *conflict* occurs when two or more trains claim the same block section simultaneously, and train ordering decisions have to be taken. Some of the trains involved in the conflict have then to change their speed profile according to the *approaching speed* required by the current signaling system. A *deadlock* occurs when several trains claim the same block section ahead which is not available and at least one train would need to be driven backward until a meet-pass becomes feasible.

The *blocking time* is the time interval in which a section of track (usually a block section) is exclusively allocated to one train and therefore is blocked for other trains (Pachl, 2002). A virtual *blocking time overlap* would arise if the minimum distance headway between two trains, which is needed for safety reasons and smooth running, is not satisfied. In that case, the approaching train is forced to decelerate and experiences a knock-on delay. A train speed profile is *acceptable* if this is compatible with the acceleration/braking rates usually performed by the train. For the Dutch signaling system, a dispatching solution is therefore *feasible* if each train has an acceptable speed profile, train distance headways are respected and there are no unsolved overlaps between circulating trains (i.e., trains are allowed passing yellow signals with the required approaching speed but must stop when face red signals).

The real-time train dispatching problem can, therefore, be defined as follows: given a railway network, a set of train routes and passing/stopping times at each relevant point in the network, and the position and speed of each train being known at time t_0 , find a deadlock-free and conflict-free schedule with no unsolved overlaps between the blocking times of subsequent trains, compatible with their initial positions and such that the selected train routes are not blocked, the speed profiles are acceptable, no train appears in the network before its intended entrance time, no train departs from a relevant point before its scheduled departure time, rolling stock constraints and connected train services are respected, and trains arrive at the relevant points with the smallest possible knock-on delay.

3.2 System architecture

This section describes our variable-speed train dispatching support system, which takes into account the consequences of braking (and subsequent acceleration) imposed to avoid headway conflicts between trains. Therefore, the solution generated provides accurate train dynamics and describes a realistic traffic plan over the intended time horizon.

Figure 3.2 presents the overall ROMA dispatching support system architecture, which is composed of interrelated procedures. A human dispatcher can interact with the system by adding/removing constraints or changing the timetable. We now describe the function of each procedure and how the four introduced subproblems are solved:

- Data loading (subproblem (i)): Collect data from the field such as the current infrastructure status, the existing timetable, the actual position and speed of all running trains, and forecast the time needed to complete the next scheduled operations (e.g. entrance delay of a train in the network, dwell time perturbations, etc.).
- Disruption recovery (subproblem (ii)): Given a default route and a prioritized set of rerouting options, find a passable routing for each train by avoiding, eventually, the tracks blocked in the studied area.

- Real-time traffic optimization (subproblem *(iii)*): Given a set of dynamic traffic management strategies, i.e., flexible orders, routes and departure times, find a new deadlock-free and conflict-free schedule by rescheduling and/or rerouting trains. In this phase the minimum traversing time for a block section for a train is considered fixed.
- Train speed coordination (subproblem *(iv)*): Given the schedule computed by the previous procedure, check if the solution is consistent with the train dynamics and if the blocking time of each train in each block section overlaps with those of the following trains, i.e., if the distance headway between two or more trains is not respected. In case of an overlap of blocking times, perform an iterative procedure to compute acceptable train speed profiles on the basis of the actual signal aspects, infrastructure and rolling stock characteristics.

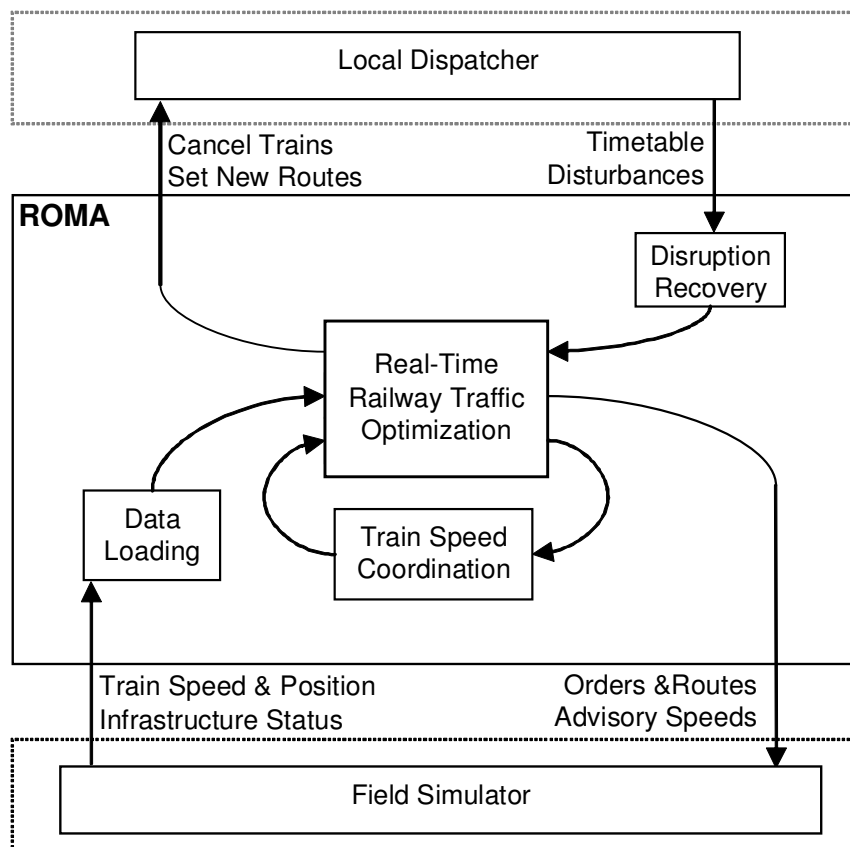


Figure 3.2: ROMA dispatching support system architecture

The data loading procedure has to gather static (off-line data such as infrastructure and timetable information) and dynamic (real-time data that varies in time) data from the field. We assume that the exact speed and location of each train are updated in real-time. Hence, the impact of inaccurate location and speed data is supposed to be negligible.

The dispatching support system must be able to efficiently detect and solve the conflicts arising in the rail network during perturbed operations. A fixed-speed traffic optimization procedure identifies the potential headway and route conflicts with a high level of accuracy

while considering all trains simultaneously. The potential headway and route conflicts are determined by predicting the future location of trains based on information about the actual state of the rail network. The traffic optimization procedure allows to take train rescheduling and rerouting decisions considering all the train speed profiles fixed as scheduled and aims at the minimization of train delays in the network. To alter the original timetable as little as possible, we focus here on the development of conflict resolution actions based, in first instance, on local dispatching measures. Large timetable modifications and cancelation of train routes are among the possible dispatching measures but not performed automatically.

A variable-speed model for train dispatching then is performed by an iterative procedure which coordinates a set of procedures to build a feasible solution for the train scheduling problem with variable-speed profiles. At each iteration the conflict detection and resolution algorithm with a fixed speed profile is used to obtain a solution taking into account the physical characteristics of the rolling stock and infrastructure.

The system solution obtained by the variable-speed dispatching system is finally suggested to the traffic controller before its actual implementation. After checking the suggested solution, the traffic controller could confirm the proposed actions or choose other dispatching measures. The dispatcher's decision would then be communicated to the interlocking system, that sets the interconnected switches and signals, and to the drivers in the train cabins by means of radio data transmission and indication of the new target speed on display. Consequently, we suppose that trains are equipped with on-board computers for automatic train control.

Due to the synchronization time, i.e., the time to react to changing conditions, speed and location modifications may happen while the dispatching support system is computing a solution. However, since the proposed variable-speed dispatching support system is able to compute a feasible solution in a few seconds, depending on the time period of traffic prediction, we assume that such real-time variations would not affect the principal validity of the rescheduling solution.

The next three subsections address each procedure separately, and we point out the limits and approximations of the proposed approach.

3.2.1 Load information

The *data loading* procedure periodically collects all the information from the field, which is required by the other procedures (subproblem (*i*)). The primary condition for calculating the future train movement is the availability of a detailed and accurately updated data set. Precisely, running times, setup times and blocking times for each operation are computed in accordance with the actual speed and position of each train at its entrance of the network, the current infrastructure status (e.g. track layout, signals, speed limits), the timetable data and the rolling stock characteristics. We next describe *real-time* and *off-line* data.

We consider real-time data gathered from the field that can change or be decided during real-time operations. Clearly, a continuous and reliable communication with the trains is

assumed, i.e., a real-time data processing unit on-board and in the traffic control center is necessary. Among the real-time data, the current operating situation has to be included, i.e., actual position and speed of the running trains at the beginning of the considered time horizon (i.e., time t_0). The expected entrance time/route for each incoming train, time windows of availability for all block sections/platforms, possible temporary speed limits occurring at some block sections, and additional scheduled stops on open tracks and stations with their scheduled arrival and departure times are real-time data which the traffic controller has to set in the dispatching support system before execution. All this information is stored before the other procedures starts.

Off-line (planning) data consists of detailed information about the infrastructure, timetable and rolling stock characteristics. The timetable contains a list of arrival/departure times (time windows of minimum/maximum arrival/departure times) for a set of relevant points in the network, including all the station platform tracks visited by each train. The infrastructure consists of a set of available block sections delimited by signals. For each block section the status, length, grade, speed limitations, traversing directions and maximum speed are given. The route release and switching times are also known off-line. Permissive speed signals are used to indicate speed restrictions in case of virtual blocking time overlaps. The specific technical characteristics of the rolling stock of each train (power, train length and weight, maximum speed and friction rates between rails and wheels) are recorded off-line in order to enable a re-calculation of the required minimal running time. The data associated with each train includes also a prioritized list of routing options (the most evident and frequently used alternative routes are selected by the dispatcher and given to the dispatching support system). The (mean) acceleration and braking rates are to be calculated on the basis of traction force/speed diagrams and scheduled maximum speeds. Here we apply speed profiles on the basis of standard acceleration and braking tables used by the Dutch infrastructure manager ProRail. Besides, we assume that the driver behavior is known in advance and follows standard braking and acceleration profiles. The weather condition, the train load (number of passengers) and weight are assumed to be a priori defined. Although some of these data may be different from day to day, for the purpose of rescheduling they are computed as off-line data. In case of substantial real-time variability of these factors, a more accurate estimation of the trains speeds and movements should be considered.

We distinguish between scheduled speed profiles (used during the timetable planning phase) and operational speed profiles (adopted in the conflict detection and resolution phase). Operational speed profiles, used in the rescheduling process, suppose that trains travel at their maximum speed according to the train characteristics, infrastructure speed restrictions and adopted standard drivers' behavior, and they are obtained by using off-line data. A detailed description of the train running speeds will be introduced in Chapter 7.

After the completion of the loading phase, the other ROMA procedures are executed assuming that real-time variations of these data would not affect the principal validity of the rescheduling solution. In Chapter 8, we adopt ROMA to predict railway traffic for several time horizons. The solutions obtained are thus applicable in real-time only for those instances solved within a few minutes, depending on the prevailing traffic conditions. How-

ever, the proposed dispatching support system is a laboratory version tested on a real-world off-line data set and, so far, does not include transmission of actual train monitoring data and data communication protocols between the dispatching system and the trains' on-board units.

3.2.2 Disruption recovery

The *disruption recovery* procedure checks if there are unavailable block sections in the network (i.e., track blockage situation of Figure 3.3 (a)), which make some train route unpassable. This activity corresponds to the resolution of subproblem (ii). For each disrupted train, this procedure discards infeasible routes, sorts the passable routing options on the basis of a priority list (given by traffic controllers) and then assigns the one with the highest priority, called the *default routing*. The default routing of each train and the set of remaining passable routes are then given to the real-time railway traffic optimization procedure of the ROMA dispatching support system.

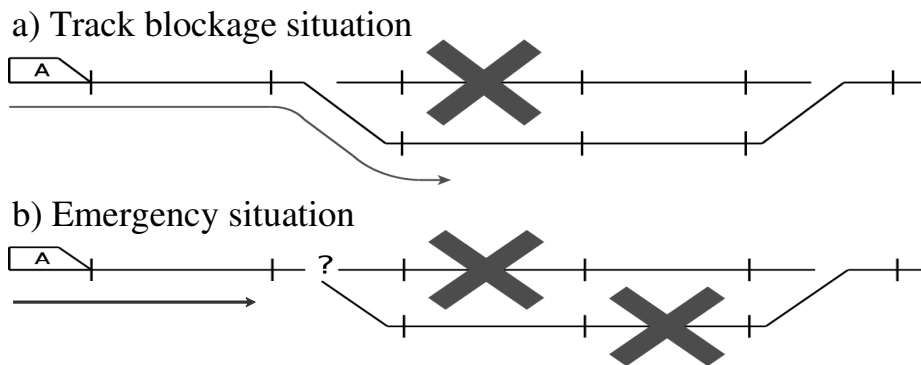


Figure 3.3: Disrupted railway corridor

If no passable route is available for a train, the system requests external support by the human dispatcher (e.g. emergency situation of Figure 3.3 (b)). In case of a heavy disruption, emergency timetables are used in which train routes are strongly modified, enabling a train to reverse the running direction according to a specific movement authority given by the traffic controllers.

3.2.3 Real-time traffic optimization

The *real-time railway traffic optimization* procedure is the decisional kernel of the train dispatching support system which is responsible for detecting and solving train conflicts while minimizing the train delay propagation and maximizing the dynamic utilization of railway capacity. Given all the necessary information by the data loading procedure and (at least) a passable route for each train by the disruption recovery procedure, the conflict detection and resolution problem (i.e., subproblem (iii)) is addressed as follows. A conflict detection procedure checks whether the timetable is deadlock-free and detects potential

conflicting train paths in a given period of traffic prediction (e.g. 15 minutes ahead). Given the actual train delays and predicted conflicts, a conflict resolution procedure computes a new feasible schedule (i.e., deadlock-free and conflict-free) compatible with the status of the network, by defining routes, orders and times for all circulating trains.

The conflict detection and resolution problem can be formulated as a job shop scheduling problem with no-store and no-swap constraints (Mascis et al., 2004). Furthermore, Mascis and Pacciarelli (2002) shows that the alternative graph is a suitable model for this job shop problem and several real-world constraints can be easily modeled by it. The real-time railway traffic optimization procedure uses a problem-dedicated alternative graph formulation (see Chapter 4) in which we formulate the train scheduling problem in first instance as a fixed-speed model. This graph represents the routes of all trains in a given control area along with their precedence constraints (minimum headways). Since a train must traverse the block sections in its route sequentially, a route is modeled in the alternative graph with precedence constraints and a chain of associated nodes. This formulation requires that a passable routing for each train is given and a fixed traversing time for each block section is known in advance, except for a possible additional waiting time between operations in order to solve train conflicts. A train schedule therefore corresponds to the set of the starting time of each operation. Since a block section cannot host two trains at the same time, a potential conflict occurs whenever two or more trains require the same block section. In this case, a passing order must be defined between the trains which is modeled in the graph by introducing a suitable pair of alternative arcs for each pair of trains traversing a block section. A deadlock-free and conflict-free schedule is next obtained by selecting one of the two alternative arcs from each pair, in such a way that there are no positive length cycles in the graph (i.e., deadlock). In other words, the alternative arcs represent operational choices such as the train order at a crossing or merging section. In order to evaluate a schedule, we use the maximum consecutive delay as performance indicator of a solution, which is the maximum delay introduced when solving conflicts in the dispatching area. This is caused by the propagation of the input delays of late trains to the other trains in the railway area. The main value of the alternative graph is the detailed but flexible representation of the network topology at the level of railway signal aspects and operational rules. In case of fixed block signaling each block signal corresponds to a node in the alternative graph and the arcs between nodes represent blocking times or headway times. Moreover, other constraints relevant to the railway practice can be included into the model, such as the ones described in Chapter 4.

The real-time railway traffic optimization procedure is in charge of computing a first feasible schedule and then is looking for better solutions in terms of delay minimization. Its architecture is described in Figure 3.4.

Given a timetable, a set of passable routes associated with each train and the current status of the network, the train scheduling procedure returns a feasible schedule for each train, i.e., defines its entrance time on each block section. Specifically, the first run of this procedure considers the default routings defined by the disruption recovery procedure. If no feasible schedule is found within a predefined time limit of computation, the human dispatcher is in

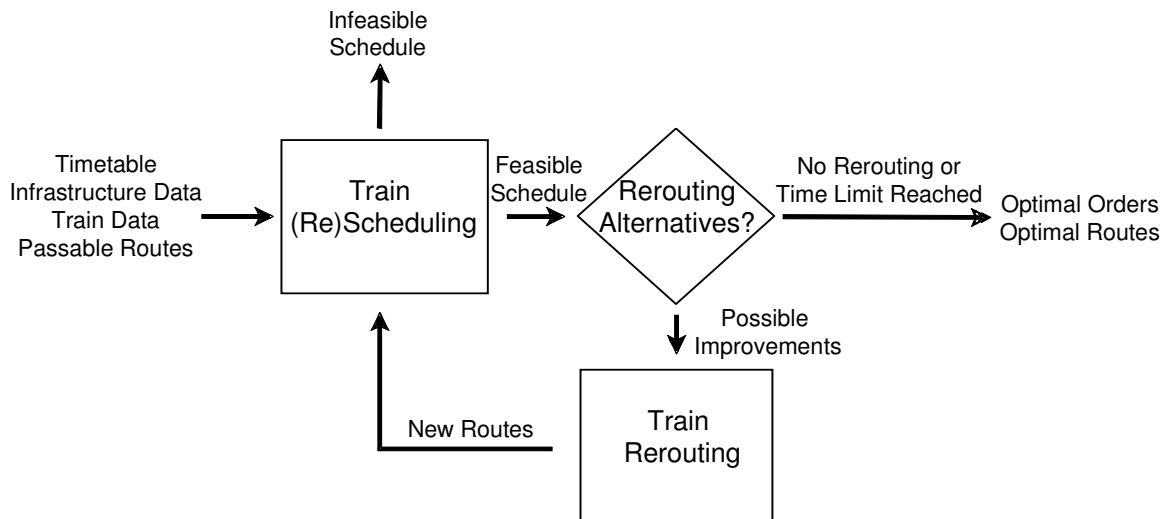


Figure 3.4: Architecture of the real-time railway traffic optimization procedure

charge of avoiding deadlocks by taking decisions that are forbidden to the automated system, such as the cancelation of a train connection and movement authorities. When a feasible schedule is found, the train rerouting procedure verifies whether local rerouting options may lead to better solutions. For each changed route, the running times and setup times are modified accordingly. Whenever some route is replaced, the train scheduling procedure computes a new deadlock-free and conflict-free timetable by thoroughly rescheduling the train movements. The combined scheduling and rerouting procedure returns the best solution found when a time limit of computation is reached or no local rerouting improvement is possible. We next introduce the algorithms used by ROMA.

Since the resolution of train conflicts has a direct impact on the level of punctuality, we implement and compare two categories of scheduling algorithms (see Chapter 5):

Dispatching rules : Well-known simple decision rules (e.g. a completely automated version of the ARI system (Berends & Ouburg, 2005), using *what-if-scenarios*), timetable orders and routes, and priority rules are evaluated as base case that simulates the practice of traffic management adopted in the Netherlands.

Efficient algorithms : Greedy heuristics and enumerative methods are explored with the objective to compute near-optimal train schedules for practical size instances within a short computation time.

In Chapter 6, we also present rerouting algorithms based on advanced heuristics, i.e., local search and tabu search. The heuristics analyze the alternative routes of each train, searching for a train route potentially leading to a better schedule. Whenever a better schedule is found, the new route is set as default route and the search is repeated. Specifically, the effectiveness of extensive rerouting strategies is explored by incorporating the local search for new routes in a tabu search scheme, in order to escape from local minima.

If the real-time railway traffic optimization procedure is unable to find a deadlock-free and conflict-free schedule, the dispatcher has to carry out other types of timetable modifications such as introducing new train routes, applying short-turning of trains in case of track blockage or even canceling of train services at some stations.

3.2.4 Train speed coordination

In the ROMA dispatching support system, the solution to subproblem (iii) presents deterministic blocking and waiting times and thus does not model explicitly the dynamic consequences of braking and subsequent acceleration imposed to avoid conflicts in the network. For the Dutch signaling system (see, e.g. Bailey (1995) and Goverde (2005)), this means that the traversing time of each train on each block section is computed assuming green signal aspects.

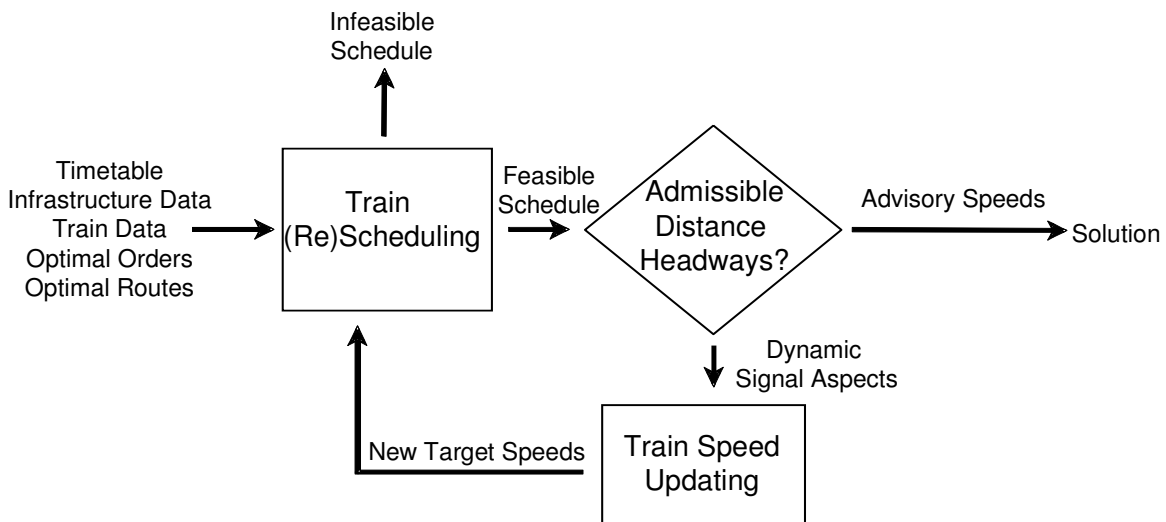


Figure 3.5: Architecture of the train speed coordination procedure

Figure 3.5 presents the variable-speed model, based on an alternative graph formulation of blocking time theory (see, e.g. Schwanhäuber (1994) and Pachl (2002)), that is able to handle subproblem (iv). We now give only a brief introduction to the train speed coordination procedure adopted by our variable-speed model and refer the reader to Chapter 7 for a detailed description.

A *feasibility check* procedure verifies whether or not the schedule is compatible with the actual train dynamics and signal aspects. For the Dutch railways, the latter case corresponds to the infeasible situation in which a train traverses a block section at yellow signal aspect without changing its speed profile, i.e., minimum safety distance headways are not respected and there is an overlap of blocking times in our model. The *speed updating* procedure therefore adjusts the speed profile of the second train (i.e., the one passing the yellow signal) according to typical driver behavior in case of variable signal aspects, and to the dynamics of the rolling stock. In the next iteration possible new “queueing” conflicts are solved, which again may lead to slowing down of trains. Feasibility checks and

speed updates are performed until all the overlaps of blocking times are solved and an acceptable speed profile is obtained for each train. As a result of this, the blocking times are then correctly recomputed. This dynamic control therefore coordinates the speed of successive trains on open tracks, secures the minimum time windows needed at junction/crossing points and synchronizes the trains arriving at stations.

In this thesis, we propose the following ROMA configurations, that require different computation times:

Iterative scheduling strategy : The real-time optimization, feasibility check and speed updating procedures are performed iteratively. At each iteration, the real-time optimization procedure solves subproblem (iii), using a chosen scheduling algorithm, and calls the feasibility check and speed updating procedures for the resolution of subproblem (iv). The speed profile of the “first” unacceptable train is updated and another run of the real-time optimization procedure is performed considering an acceptable speed profile for this train. The iterative procedure is executed only a finite number of times.

Single scheduling strategy : This is a simplification of the iterative scheduling strategy. The subproblems (iii) and (iv) are solved in cascade. The real-time optimization procedure solves subproblem (iii), using a chosen scheduling algorithm, and calls the feasibility check and speed updating procedures until admissible speed profiles are calculated for every train. Following this strategy, the sequencing obtained by the real-time optimization procedure is therefore not modified during the resolution of subproblem (iv).

The combined adjustments of train speeds and orders result into computationally complex rescheduling instances. For this reason, we do not make use of train rerouting algorithms when dealing with the variable speed model (as shown in Figure 3.5). In the actual implementation of our dispatching support system, optimal train routes are therefore computed by the real-time railway traffic optimization procedure, and cannot be changed during the following train speed coordination procedure.

When dealing with large train rescheduling instances, we also introduce a temporal decomposition approach in which the railway traffic flow is divided into tractable time intervals to be solved in cascade. The independent resolution of each time interval permits handling large instances within a linear increase of computation time (as described in Chapter 8).

3.3 Conclusions

A real-time train dispatching system has to solve disturbances in the traffic flow by adjusting the timetable of each train in terms of routing and timing, and by resequencing the trains at the entrance of each merging/crossing point. This chapter presents the architecture of

our laboratory dispatching support system designed for the real-time optimization of train scheduling, routing and speed coordination. The main goal is to minimize knock-on delays (i.e., the difference between the arrival time at each relevant point in the new schedule and that in the timetable) while satisfying the traffic regulation constraints and the compatibility with the real-time position of each train. Such a system can be implemented in an advanced traffic management system which improves the operations reliability and stability with low investments compared to the alternative of building new infrastructure.

The future evolution of railway traffic could be predicted using real-time data gathered from the field (subproblem *(i)*). However, the technical implementation issues concerning the practical operation, such as data transmission, communication of delays and the realization of the proposed dispatching measures, are outside the scope of this thesis (we refer the reader to e.g. Santos et al. (2005), Hailes (2006) and Neil (2006)).

We are aware that the required information may not always be readily available, but the current railway signaling systems are fitted with intermittent and continuous automatic train protection systems. The existing train describer system records automatically the occupation and release of each signal block and the train number and its actual passing time at the critical block signals of the network. This information is used by many railways for comparison with the scheduled passing times. Thus, the difference between the scheduled and the measured times is computed and could be used, too, as input data for the dispatching support system (see Goverde and Hansen (2000) and Daamen et al. (2007)).

ROMA is able to optimize railway traffic also if the timetable is not deadlock-free and conflict-free, and can be used to test the schedule robustness. This enables managing railway traffic even in case of severe traffic disturbances, including the presence of completely blocked tracks and routes (subproblem *(ii)*), such as when emergency timetables are required and dispatchers need efficient support to solve conflicts.

The proposed approach combines rerouting and rescheduling strategies for solving real-time disturbances, which require extensive timetable modifications to reach a feasible schedule. All trains are taken into account simultaneously, aiming at maximizing the train punctuality and the use of rail capacity (subproblem *(iii)*).

By using detailed information on the actual infrastructure status and rolling stock characteristics, the ROMA dispatching support system can calculate accurate running times and blocking times that identify the effective remaining distances and running times until the next conflict points. Since the estimation of the train speed profile has a strong impact on the consecutive delays, an iterative scheduling strategy is implemented to combine the conflict resolution and train speed coordination control actions (subproblem *(iv)*). However, in the current implementation of our dispatching support system the train dynamics are computed separately from the scheduling problem.

Chapter 4

Fixed-speed traffic optimization model

The design and implementation of advanced mathematical models is a prerequisite to the development of innovative decision support systems for solving the on-line train dispatching problem. This chapter is concerned with the modeling of the conflict detection and resolution (CDR) problem for a regional railway network. The CDR problem is the real-time problem of computing a conflict-free and deadlock-free schedule compatible with the actual status of the network and such that the circulating trains arrive and depart with the smallest possible delay. In other words, the CDR problem consists of assigning a passable route to each train and timings for all chosen routes (i.e., starting times t_1, \dots, t_n to operations with $t_0 = 0$) such that setup time constraints between operations are satisfied, no train enters the network before its release time (i.e., its actual schedule is adapted if required because of disturbances) and expected consecutive delays are minimized.

From the above description, it can be observed that the combinatorial structure of the CDR problem is similar to that of a job shop scheduling problem with several additional constraints (see, e.g. Szpigel (1973), İ. Şahin (1999) and Oliveira and Smith (2000)). In the usual definition of the job shop scheduling problem, a job must be processed by a prescribed sequence of machines and each machine can process one job at a time. The processing of a job on a machine is an operation. The job shop scheduling problem therefore consists of defining starting times for all operations such that each operation of a job starts after the completion of its predecessor and no machine processes two operations simultaneously. A detailed description of the job shop scheduling problem and complexity issues can be found e.g. in Manne (1960), Lenstra and Rinooy (1979), Graham et al. (1979), Pinedo (1995), Brucker (1998), Jain and Meeran (1999), Shakhlevich et al. (2000) and Flamini (2005).

In terms of the CDR problem, jobs correspond to running trains and machines to block sections of the signaling and train control system. The processing time of an operation can be used to represent the running time of the associated train on the corresponding block section. Since a block section cannot host two trains simultaneously, there is a conflict if two or more trains claim the same block section at the same time and therefore would be in conflict with the minimum setup time required for that block section. Solving the conflict corresponds to defining a processing order and time between incompatible operations (i.e.,

claims of infrastructure capacity by different trains). In a CDR solution, a set of passable routes and timings are feasible if, for each pair of operations associated to the same block section, the minimum setup time constraints are satisfied and there is no deadlock in the network. This implies that even answering the question of whether a schedule compatible with the real-time train positions exists or not (i.e., solving the CDR problem) is an NP-complete problem, as proved by Mascis and Pacciarelli (2002).

In scheduling theory the presence of an infinite buffer capacity between two consecutive machines is usually assumed. Obviously this is not the case for the train scheduling problem; in fact, it is impossible to “store” trains between two consecutive block sections. Blocking (or no-store) constraints model the absence of infinite buffer capacity between two consecutive machines, i.e., a job cannot be stored between consecutive operations. In other words, these constraints impose that a train, having reached the end of a track segment, cannot enter the subsequent segment if the latter is occupied by another train, thus preventing other trains from entering the former segment. Differently, no-swap constraints impose that a job can move to the next machine only strictly after that the previous job left it, thus forbidding two jobs to exchange their respective machines.

Mascis and Pacciarelli (2002) introduce the alternative graph formulation to model variants of the job shop problem. An alternative graph formulation of the CDR problem was first developed within the European project COMBINE (we refer the reader to e.g. Mascis et al. (2001), Mascis et al. (2002) and Mascis et al. (2004)). In this thesis, we extend the alternative graph formulation of the CDR problem to predict accurately the future evolution of the railway traffic on the basis of the actual train positions and speeds, signaling and safety system constraints. Our optimization model incorporates a detailed description of the network topology, including railway signal aspects and safety rules, on the basis of each track segment between block signals being able to host at most one train at a time. We make use of an alternative graph formulation of the CDR problem that allows the modeling of job shop scheduling problems with no-store and no-swap constraints. We also show that the alternative graph model can easily incorporate several other railway relevant traffic regulation rules and constraints, which are rarely taken into account in the literature, as observed by Oliveira and Smith (2000). Examples of such aspects include speed restrictions, precedence and meeting constraints, setup times, route booking constraints, scheduled arrival and departure times. Additional real-world constraints due to passenger satisfaction are also considered, such as minimum transfer time between connected train services.

It should be remarked that the proposed definition of the CDR problem is slightly different from the one mostly used by traffic controllers. In the dispatching practice, the usual approach to manage train traffic in a railway network is to identify and resolve potential conflicts *one at a time*. According to Kauppi et al. (2006), this lack of overview in the time/distance domain causes sub-optimization. The CDR aim is to develop a *new conflict-free plan*, such that the overall deviation from the planned schedule is minimized (see e.g. Jacobs (2004)). We consider a *global conflict resolution* problem in which all trains are managed simultaneously within the overall studied area for a given time period.

Due to the inherent complexity of the CDR problem, we generate a solution solving the

routing and scheduling problems separately. The compound CDR problem can therefore be partitioned into two subproblems: (i) given a route for each train, define the starting times of each operation (i.e., train ordering and timing); (ii) given a solution to sub-problem (i) and a set of rerouting possibilities associated to each train, define new train routes potentially leading to better schedules.

We next refer to the problem of *conflict detection and resolution with fixed routes* (CDRFR) to denote subproblem (i) in which we order the trains being fixed to their respective routes, with the objective function of minimizing the maximum consecutive delay. For subproblem (ii), we assume that the output of the train rescheduling is an optimal schedule for the given route assigned to each train, even if this is not a necessary condition, and we try to improve the solution modifying the route of some train. The two sub-problems are then solved iteratively until no improvement is possible or a given time limit of computation is reached. The scheduling and rerouting algorithms are described, respectively, in Chapters 5 and 6.

This chapter is organized as follows. We adopt the alternative graph model in order to formulate the CDRFR and CDR problems. Section 4.1 describes briefly the alternative graph model while Section 4.2 provides a formal description of the CDRFR problem formulation. Several railway constraints are formulated and illustrative examples are also reported. Section 4.3 then introduces the CDR problem with some definitions and gives a numerical example of optimal CDRFR and CDR solutions. Some conclusions follow on the limitations and possible extensions of the proposed formulations.

4.1 Alternative graph model

The alternative graph model (Mascis & Pacciarelli, 2002) generalizes the disjunctive graph formulation (Roy & Sussman, 1964) and is able to take into account several constraints arising in real-world scheduling applications. In order to formally define an alternative graph, let us introduce the following definitions. A triple $\mathcal{G} = (N, F, A)$ is an alternative graph, where N is a set of nodes, F is a set of *fixed* directed arcs and A is a set of pairs of *alternative* directed arcs.

We denote with $N(F)$ a subset of N such that for each node $i \in N(F)$ there is at least an arc of F adjacent (connected) to node i . Set N includes two dummy nodes 0 and n , called *start* and *finish* respectively, such that for each node $i \in N(F)$ there is a directed path from node 0 to node i and from node i to node n in the graph $\mathcal{G}(\emptyset) = (N, F, \emptyset)$. We call a node *isolated* if this is element of N but not element of $N(F)$. Note that the dummy nodes 0 and n are isolated only if there are no other nodes in N .

Except for dummy nodes, each node i of the alternative graph is associated to an operation o_i . We have a set of operations o_0, o_1, \dots, o_n to be performed on m machines m_1, m_2, \dots, m_m , where o_0 and o_n are (dummy) operations with zero processing time. A machine m_i can process only one operation at a time, and an operation o_i cannot be interrupted from its starting time t_i to its completion time.

The length of arc $(i, j) \in F$ is a given quantity b_{ij} . Each arc (i, j) , either fixed or alternative, represents a *precedence relation* constraining the starting time (t_i) of operation o_i , with respect to the starting time t_j of operation o_j , i.e., $t_j \geq t_i + b_{ij}$. Set F also includes outgoing arcs from node 0 and ingoing arcs to node n .

In general, the alternative arcs of each pair $((i, j), (h, k)) \in A$ are the decision variables in the alternative graph model, i.e., a decision is made by choosing arc $(i, j) \vee$ (or) arc (h, k) . A *graph selection* S is a set of alternative arcs obtained from A by choosing at most one arc from each alternative pair. Given a pair $((i, j), (h, k)) \in A$, we say that arc (i, j) is *selected* in S if $(i, j) \in S$, whereas we say that arc (i, j) is *forbidden* in S if arc $(h, k) \in S$. We also say that (i, j) and (h, k) are *paired*, and that (i, j) is the *companion* of (h, k) . Finally, the pair is called *unselected* if neither (i, j) nor (h, k) is selected in S .

Given a selection S , we let $\mathcal{G}(S)$ indicate the graph $(N, F \cup S)$. The selection S is *consistent* if the graph $\mathcal{G}(S)$ has no positive length cycles. Given two nodes $i, j \in N(F)$, we denote with $l^S(i, j)$ the length of the longest path from node i to node j in $\mathcal{G}(S)$. We say that the quantity $r_i = l^S(0, i)$ is the *head* of node i , and the quantity $q_i = l^S(i, n)$ is the *tail* of node i . The longest path from the start node 0 to the finish node n in $\mathcal{G}(S)$ is called the *critical path set* $\mathcal{C}(S)$ of the graph. A selection S is *complete* if exactly one arc from each alternative pair is selected. Given a consistent selection S , we call an *extension* of S a complete consistent selection S' such that $S \subseteq S'$, if this exists. The objective function is the minimization of the starting time t_n of operation o_n , i.e., the *makespan*. A *solution* is a complete consistent selection S , and its value is therefore $l^S(0, n)$. An optimal alternative graph solution is indicated as S^* while an empty selection is denoted as S^\emptyset .

To summarize, the alternative graph model corresponds to the mathematical problem:

$$\begin{aligned} \min \quad & t_n - t_0 \\ \text{s.t.} \quad & t_j - t_i \geq b_{ij} && (i, j) \in F \\ & (t_j - t_i \geq b_{ij}) \vee (t_k - t_h \geq b_{hk}) && ((i, j), (h, k)) \in A \end{aligned}$$

4.2 CDRFR problem formulation

In our formulation of the CDRFR problem, each train (job) must pass through a prescribed sequence of block sections (machines). An operation o_i corresponds to the traversing of a block section B_u for a train T_y , i.e., is associated with the pair $\langle B_u, T_y \rangle$. The route of T_y is therefore a sequence of operations $\langle B_1, T_y \rangle, \langle B_2, T_y \rangle, \dots, \langle B_k, T_y \rangle$, where the operation o_i associated with the pair $\langle B_u, T_y \rangle$ is the successor of $\langle B_{u-1}, T_y \rangle$ and the predecessor of $\langle B_{u+1}, T_y \rangle$. The traveling of T_y through B_u (an operation) is a node of the alternative graph. The route of T_y is therefore a chain of nodes to be processed in sequence. Given an operation o_i on a route, we define operations $o_{\sigma(i)}$ and $o_{\mu(i)}$ as the following and preceding block sections to be traveled by the train on its route. Two additional dummy operations o_0 and o_n represent the start and end of the schedule, and each train route is modeled with a chain of fixed arcs from node 0 to node n . Moreover, we assume that o_0 is an ancestor of all o_1, \dots, o_n while o_n is a descendant of all o_0, \dots, o_{n-1} .

Train routes are represented with fixed arcs that correspond to the precedence constraints between consecutive operations. For each operation o_i of a train route there is a fixed arc $(\mu(i), i) \in F$ with length $b_{\mu(i)i} = p_{\mu(i)}$. Denoting with $p_{\mu(i)}$ the processing time of operation $o_{\mu(i)}$, the running time constraint imposes that o_i can start only after at least $p_{\mu(i)}$ time units from the completion of $o_{\mu(i)}$, i.e., $t_i \geq t_{\mu(i)} + p_{\mu(i)}$. Furthermore, fixed arcs are used to impose other railway constraints (e.g. release times, rolling stock constraints, connected train services) and to compute the train delays.

For each block section to be traversed by at least two trains, a set of all associated operations is considered. Since a block section cannot host two trains at the same time, alternative arcs are adopted to model the no-store (or blocking) constraint of the CDRFR problem, i.e., are used to determine the train orders. The length of the alternative arcs is given by the setup times that ensure a minimum distance headway between consecutive trains. For each pair o_i and o_j of operations associated with the entrance of two trains in a block section, we introduce the pair of alternative arcs $((\sigma(i), j), (\sigma(j), i)) \in A$, with $b_{\sigma(i)j} = a_{ij}$ and $b_{\sigma(j)i} = a_{ji}$. Assuming that operation o_i precedes o_j , let a_{ij} be their setup time (i.e., length of arc $(\sigma(i), j)$). The setup time constraint requires that the train associated to o_i must leave the block section at least a_{ij} time units before the train associated to o_j can enter the block section, i.e., $t_j \geq t_{\sigma(i)} + a_{ij}$. Similarly, if o_j precedes o_i then $t_i \geq t_{\sigma(j)} + a_{ji}$ holds.

A timing is *feasible* if the precedence constraints are satisfied for all operations. In the alternative graph, running and setup times depend on block section characteristics and on the assigned train speed profile. We next describe, with some example, the formulation of running and setup constraints, CDRFR solutions, scheduled stops and other additional precedence constraints, such as rolling stock and passenger connections.

4.2.1 Running and setup time constraints

Figure 4.1 shows the alternative graph formulation for two trains (T_A and T_B) at a junction. A fixed arc (i, k) (depicted with solid arrows in Figure 4.1) represents the traveling of T_A through the junction, i.e., the block section delimited by two signals. The arc length b_{ik} corresponds to its running time p_i . Similarly, the arc length b_{jh} corresponds to the running time p_j of T_B through the junction. Since the junction cannot host T_A and T_B at the same time (i.e., two jobs in the alternative graph require the same resource), there is a potential conflict. In this case, a processing order must be defined between the incompatible operations, and we model this constraint by introducing in the graph a suitable pair of alternative arcs from the set A (depicted with dashed arrows in Figure 4.1). Each alternative arc models a possible precedence relation between two operations, and the length of an alternative arc is the minimum time headway between the associated trains. In the example of Figure 4.1, given a fixed arc (i, k) , we say that o_k is the successor of o_i and denote it by $o_k = o_{\sigma(i)}$. By letting o_i and o_j be the two conflicting operations at the railway junction, we model the two possible processing orders with the pair of arcs $((k, j), (h, i)) \in A$. If o_i is scheduled before o_j , the alternative arc (k, j) is selected and T_A precedes T_B . The length of the alternative arc (k, j) corresponds to the setup time a_{ij} between o_i and o_j ($t_j \geq t_k + a_{ij}$).

Similarly, if o_j is scheduled before o_i , then T_B precedes T_A and the length of (h, i) is the setup time a_{ji} ($t_i \geq t_h + a_{ji}$). If $t_i < t_j < t_k + a_{ij}$ or $t_j < t_i < t_h + a_{ji}$ then a conflict between T_A and T_B is detected and must be solved choosing one of the two arcs. Specifically, if o_i is processed first, then o_j must wait for the completion of the running time p_i and the starting of $o_{\sigma(i)}$ plus the setup time a_{ij} between o_i and o_j . If the selection of the alternative arc (k, j) increases the starting time t_j of operation o_j , a conflict between T_A and T_B has been detected and solved for the CDRFR problem.

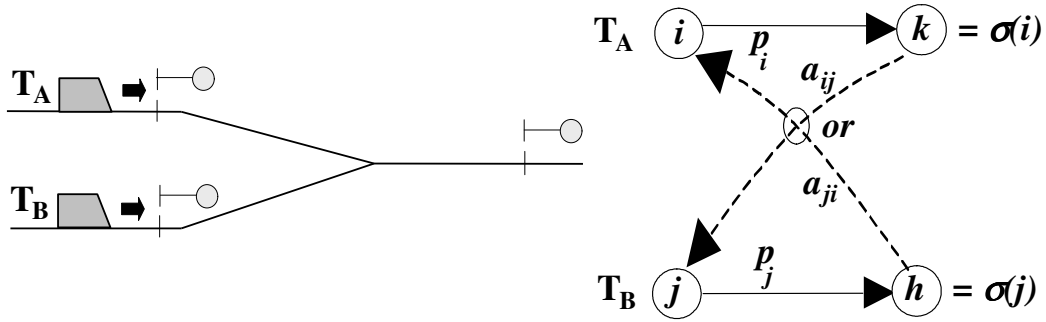


Figure 4.1: The alternative graph formulation for two trains at a junction

4.2.2 CDRFR solution description

In terms of the alternative graph formulation, a solution to the CDRFR problem is a feasible starting time t_i to each operation o_i (i.e., the exact time in which each train will enter each block section) such that all fixed precedence relations are satisfied, exactly one of each pair of alternative precedence relations is selected and the resulting alternative graph has no positive length cycles. Note that a positive length cycle represents a deadlock situation (i.e., an operation preceding itself), which is infeasible. In the problem we deal with in this thesis, negative length cycles only occur in case of perishability constraints. In general, negative length cycles allow to model more general scheduling situations (Mascis & Pacciarelli, 2002). To summarize, a conflict-free and deadlock-free schedule for the CDRFR problem is associated with a selected alternative graph $\mathcal{G}(S)$.

Given an initial selection S^I , potentially empty, the objective of the CDRFR problem is to find a complete consistent selection S such that $S^I \subseteq S$ and the length of the longest path from node 0 to node n in $\mathcal{G}(S)$ is minimized. By choosing suitable lengths on the arcs entering node n , the maximum consecutive delay of the solution is $l^S(0, n)$. The selection S^I represents the precedence constraints due to the initial positions of the trains at time t_0 and/or to their order of entrance into the network.

We now model the minimization of the maximum consecutive delay in our alternative graph formulation. Typically the punctuality of railway operations is measured at some relevant points in the network on the basis of the times scheduled in the timetable. A *relevant point* can be either a terminal block section (the last block section before the exit from the dispatching area), a route node block section where two or more trains cross, merge or

diverge, and a platform block section corresponding to a scheduled dwell time at a station. Consecutive delay minimization at relevant points can be achieved introducing suitable scheduled times. Let α_{yu} be the scheduled arrival time of a train T_y at a block section B_u in the timetable, which can be infeasible in case of real-time disturbances. Let τ_{yu} be the earliest possible arrival time of T_y at B_u computed according to its initial position, initial speed, assigned route and following a maximum speed profile (allowed by the train characteristics and infrastructure) in the empty network (i.e., by disregarding the presence of other trains). Note that τ_{yu} does not take into account possible conflicts with other trains, and therefore τ_{yu} is a lower bound on the feasible arrival time of T_y at the u -th relevant point, i.e., the starting time t_i of the corresponding operation o_i . We define the total delay of T_y at B_u as the difference between its actual arrival time t_i and α_{yu} . We divide the total delay into two parts as follows. If $\tau_{yu} > \alpha_{yu}$, then the quantity $\tau_{yu} - \alpha_{yu}$ is an unavoidable delay that cannot be recovered by real-time rescheduling of train operations. We call $\max\{0, \tau_{yu} - \alpha_{yu}\}$ the initial (primary) delay of T_y at B_u . The quantity $\max\{0, t_i - \max\{\tau_{yu}, \alpha_{yu}\}\}$ is called consecutive (secondary) delay of T_y at B_u , which is the additional delay due to the solution of conflicts between T_y and the other trains circulating in the network. We do not take into account the initial delays in the scheduling phase, and define a *modified due date* for T_y at B_u as $\max\{\tau_{yu}, \alpha_{yu}\}$. Adding an arc from node $i \in N(F)$ to node n (with length $-\max\{\tau_{yu}, \alpha_{yu}\}$) for each relevant point, the makespan $l^S(0, n)$ corresponds to the maximum consecutive delay for the CDRFR problem.

4.2.3 Scheduled stop formulation

We show the alternative graph formulation of a train approaching its (flexible) scheduled stop and how the longest path $l^S(0, n)$ can represent the maximum consecutive delay of the associated schedule (see Figure 4.2).

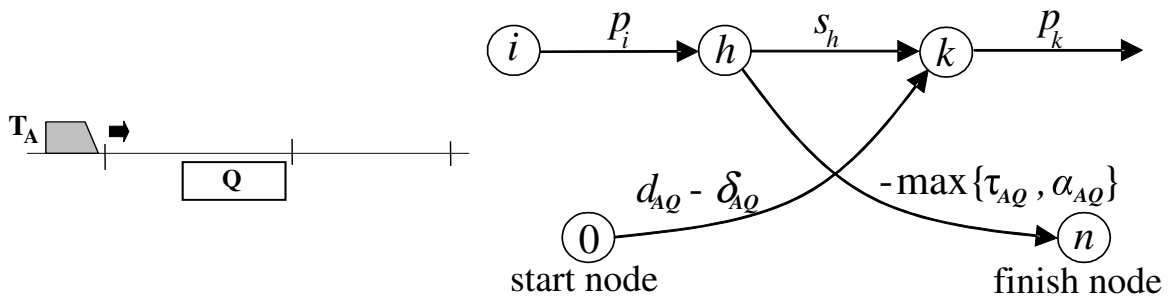


Figure 4.2: The alternative graph formulation of a scheduled stop

The formulation of arrival and departure times requires the introduction of two nodes i and h for a train T_A stopping at a scheduled stop Q . Let o_i be the operation associated to the entrance of T_A at the block section (relevant point) B_u with the scheduled stop Q , and p_i the running time of T_A at B_u until the complete stop at the end of the block section. Let o_h be the operation associated to the scheduled dwell time s_h of T_A at B_u , i.e., the scheduled stopping time at operation o_h . Let o_k be the operation associated to the entrance of T_A at

its next block section, with running time p_k . In Figure 4.2, the value d_{AQ} [α_{AQ}] represents the scheduled departure [arrival] time of T_A at Q , respectively. Precisely, T_A is not allowed to depart from Q before d_{AQ} and is considered late if arriving after α_{AQ} . The constraint imposed by the timetable on the scheduled departure time d_{AQ} ($t_k \geq d_{AQ}$) is obtained by adding the fixed arc $(0, k)$ with length d_{AQ} . On the other hand, by adding a fixed arc (h, n) from node $h \in N(F)$ to n , having length $-\max\{\tau_{AQ}, \alpha_{AQ}\}$, the length of the path from node 0 to n passing through (h, n) is equal to the consecutive delay of T_A at Q . The delay over the scheduled time $-\max\{\tau_{AQ}, \alpha_{AQ}\}$ observed in a schedule is the contribution of the relevant point to our objective function. Adding one such arc to the alternative graph for each train at each scheduled stop, as well as at its exit point from the network, the minimization of the maximum consecutive delay is obtained by minimizing $l^S(0, n)$ in the alternative graph with modified due dates.

In what follows, we construct a *flexible departure time* for T_A at Q by decreasing the scheduled departure d_{AQ} over a flexibility value δ_{AQ} . The length of arc $(0, k)$ is therefore equal to $d_{AQ} - \delta_{AQ}$ (see Figure 4.2). In general, we define the *timetable flexibility* as the quantity $\delta = \min_{y,u} \{\delta_{yu}\}$. In case of fixed departure times, $\delta = 0$ holds. In other words, introducing a positive value δ simply corresponds to the possibility for a train to leave (if possible) a scheduled stop earlier than scheduled in the timetable. On the other hand, we only consider a fixed arrival time α_{AQ} since increasing this value would immediately result in a reduced train delay without changing its actual arrival time.

4.2.4 CDRFR illustrative example

We present a small railway network with trains running at different speeds. Figure 4.3 shows the studied infrastructure with four block sections (denoted as 1, 2, 3 and 9), a simple station with two platforms (6 and 7) and three junctions (4, 5 and 8). In this example, we only show the location of the most relevant block signals. However, each block section has, clearly, the capacity of one train at a time. At the starting time t_0 , there are two trains in the network. Train T_A is a slow train running from block section 3 to block section 9 and stopping at platform 6. T_A can enter a block section only if the signal aspect is yellow or green. Train T_B is a fast train running from block section 1 to block section 9 through platform 7 without stopping. T_B can enter a block section at high speed only if the signal aspect is green. At t_0 , we therefore assume that T_B requires two empty block sections.

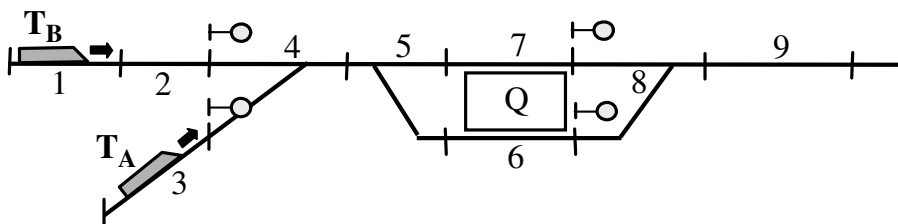


Figure 4.3: A small railway network with two trains

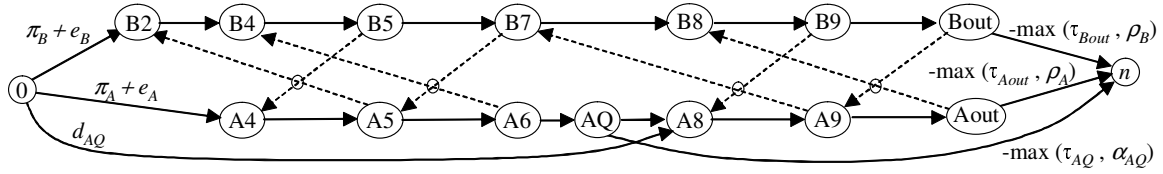


Figure 4.4: A first alternative graph formulation of the CDRFR example

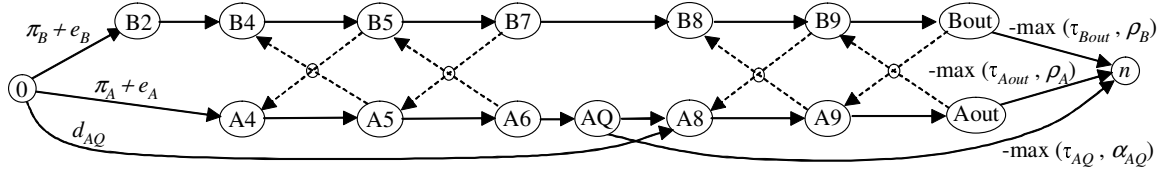


Figure 4.5: A second alternative graph formulation of the CDRFR example

In Figures 4.4 and 4.5, two alternative graph formulations of this example are reported. For the sake of clarity, a node of the graph can be identified by a pair (train, block section), by a pair (train, scheduled stop) or by a pair (train, exit point), except for the dummy nodes 0 and n . Each pair of alternative arcs is associated with the usage of a common block section by two trains (i.e., with two conflicting operations), and is represented by connecting the two paired arcs with a small circle. For simplicity, the length of fixed and alternative arcs representing running and setup constraints is not depicted. Since T_A and T_B share block sections 4, 5, 8 and 9, there are four pairs of alternative arcs. The values $\pi_A + e_A$ and $\pi_B + e_B$, depicted respectively on arcs $(0, A4)$ and $(0, B2)$, represent the time at which T_A and T_B are scheduled to reach the end of their current block sections. The values π_A and π_B are their scheduled entrance times while the values e_A and e_B are their entrance delays. The length d_{AQ} of arc $(0, A8)$ is the scheduled departure time of T_A from the scheduled stop Q , while the length of arc $(AQ, A8)$ (not depicted in Figure 4.4) is its scheduled dwell time. The length $-\max\{\tau_{AQ}, \alpha_{AQ}\}$ is the modified due date of T_A at the scheduled stop. Let ρ_A and ρ_B be the scheduled exit times of T_A and T_B , the modified due dates at the exit of the network (“out”) are $-\max\{\tau_{Aout}, \rho_A\}$ and $-\max\{\tau_{Bout}, \rho_B\}$.

We now discuss the different formulations of Figures 4.4 and 4.5 with respect to the resolution of train conflicts for trains traveling at different speeds. The alternative graph of Figure 4.4 considers the different type of trains (i.e., slow and fast) by introducing alternative pairs to one and two block sections ahead. This formulation permits to include information about the different speed profiles of trains while solving train conflicts. Specifically, in the proposed example the problem arises when the slow train is scheduled before the fast train. On the other hand, the alternative graph of Figure 4.5 does not include any of this information and treats the two trains in the same way when taking ordering decisions. However, in both cases a feasible speed profile needs to be computed if the two trains do not respect the minimum required distance headway in each block section. In this thesis, we make a clear distinction between conflict resolution and train speed coordination, and use of the formulation of Figure 4.5 to deal with the necessary speed profile adjustments after solving the CDRFR problem (a detailed description can be found in Chapter 7).

4.2.5 Additional railway constraints

The alternative graph formulation allows the modeling of situations more general than the ones we have mentioned. For example, Oliveira and Smith (2000) point out the relevance of precedence and meeting constraints in the management of railway operations. A precedence constraint between two trains at a station may be necessary if one train is carrying the crew or a vehicle necessary for another train. In such cases the latter train must wait for the arrival of the former one before departing. This constraint can be modeled by simply adding one fixed arc of suitable length from the node associated to the arrival of the former train to the node associated to the departure of the latter one. A meeting constraint allows specification of a minimum dwell time during which two given trains must be together at a given station for exchanging passengers or goods. This can be modeled by adding two fixed arcs of suitable length: an arc from the node associated with the arrival of the former train to the node associated with the departure of the latter one, another arc from the node associated with the arrival of the latter train to the node associated to the departure of the former one. In the remaining part of this subsection, we describe the alternative graph formulation of some additional railway constraints.

Rolling stock connection constraints

Figure 4.6 shows an example of *rolling stock connection* constraints. A train stopping at a station platform and departing in the opposite direction is shown. This constraint is modeled by adding a fixed arc from the node associated with the stopping of a train T_A at a scheduled stop Q , to the node associated with the departure of a train T_B , i.e., node $B3$. The same model can be used for representing other constraints, such as the case in which part of the rolling stock of T_A is used to form T_B or the case when T_A carries part of the crew of T_B . Note that the length of arc $(AQ, B3)$ should include a minimum turning-around time since T_A and T_B are scheduled to be coupled at Q .

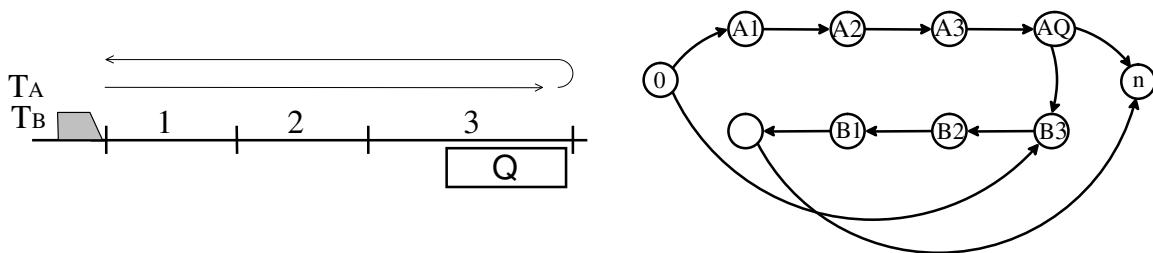


Figure 4.6: Rolling stock connections

Passenger connection constraints

In Figure 4.7, we present an example of a network with two trains T_A and T_B . The two trains have a scheduled stop Q at a station platform. In the graph, the two arcs (AQ, n) and

(BQ, n) are used to represent the delay contribution of the two trains at Q . Moreover, arcs $(0, A3)$ and $(0, B6)$ restrict the departure times of the two trains to be greater or equal to the ones in the timetable. Finally, the horizontal arcs $(AQ, A3)$ and $(BQ, B6)$ model the dwell times.

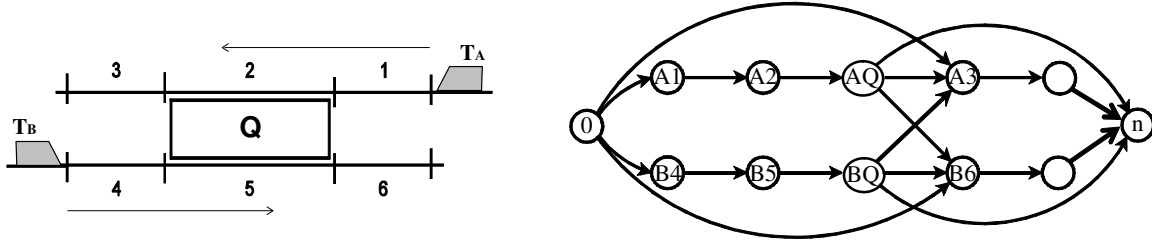


Figure 4.7: Passenger connections

On the right side of Figure 4.7, we report the alternative graph formulation of *passenger connection* constraints. To let passengers moving from one train to another, each train must depart sufficiently later with respect to the other. We model this constraint with two diagonal fixed arcs $(BQ, A3)$ and $(AQ, B6)$, which constrain the departure time of a train with respect to the arrival time at the platform of the other. In general, the arc length depends on the distance between the two stopping platforms. Of course, a unilateral passenger connection is also possible.

Route booking constraints

Further requirements could be necessary to model the traffic flow in a railway network. In some cases, if a railway line slopes up over a certain gradient there may be some freight trains that should not decrease their speed under a certain limit, due to limited traction power and heavy weight. These freight trains would not be able to climb up to the top, if they were forced to stop on the slope and to accelerate again. This situation can be modeled by using *route booking constraints*.

In the example of Figure 4.8, we address the formulation of route booking constraints for a train T_A running in a corridor before a scheduled stop. We introduce a maximum allowed time for T_A to run on a sequence of block sections until a scheduled stop Q , i.e., a maximum time to process the consecutive operations o_i , o_j and o_h in a job. Considering the pair of consecutive operations o_i and o_j , we model that o_j must start its processing within f_i time units after the completion of o_i ($f_i \geq 0$). In scheduling theory, this is known as a *perishability constraint*, since it represents the fact that a job deteriorates when stored for more than f_i time units between the two consecutive operations. We can represent this constraint with a pair of fixed arcs (i, j) and (j, i) having length p_i and $-p_i - f_i$, respectively.

We now present the interesting case in which $f_i = f_j = f_h = 0$ (no-wait constraints), i.e., T_A is not allowed to change its running time in each block section of Figure 4.8. We indicate this specific route booking formulation as a *green wave strategy* since T_A must run at its scheduled speed in the corridor and (eventually) wait for other trains only at its scheduled

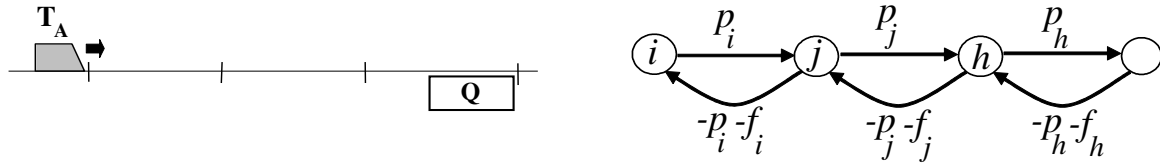


Figure 4.8: Route booking constraint

stop Q . In other words, the usual traffic management strategy is to solve train conflicts by modifying both running times and dwell times, while the green wave strategy permits only dwell time modifications.

4.3 CDR problem formulation

An alternative graph represents all the scheduling alternatives when train routes are defined. As far as the CDR problem is concerned, we observe that changing a route implies changing the set F and A in \mathcal{G} . If a train can be assigned to different routes, then F is a decision variable and A depends on the choice of F . Note that each arc of F associated with the traversing of a block section has a length equal to the running time of the associated train on the associated block section. In fact, this depends also on the train speeds on the previous/following block sections. Changing a part of a route for a train may therefore cause adjusting its running time on several block sections that have not been modified. In such cases, new fixed arcs replace all the arcs of F that are associated to a different block section or to the same block section but with a different running time. Once a route-set F is defined, we call $S(F)$ a complete consistent selection obtained from A in \mathcal{G} , and $\mathcal{G}(F, S)$ indicates the alternative graph solution $(N, F \cup S)$. A solution to the CDR problem is therefore a pair $(F, S(F))$.

We recall $\mathcal{C}(F, S)$ the set of all critical nodes, and the length $l^{F, S(F)}(0, n)$ of $\mathcal{C}(F, S)$ in $\mathcal{G}(F, S)$ is the maximum consecutive delay. This value can be reduced by either changing the train sequencing (i.e., the set $S(F)$) or modifying the train routes (i.e., the set F). An optimal CDR solution $(F^*, S^*(F^*))$ is therefore to find an optimal route-set F^* and a corresponding optimal graph selection $S^*(F^*)$.

4.3.1 CDR illustrative example

Figure 4.9 shows a railway network composed of 14 block sections and three trains T_A , T_B and T_C . For the sake of clarity, we only show the location of the most relevant block signals. In the timetable, the default route of T_A is given by the sequence of operations $A1$, $A2$, $A3$, $A9$, $A12$, $A13$, $A14$, even if also the passable routes $A1$, $A2$, $A3$, $A9$, $A10$, $A5$, $A13$, $A14$ and $A1$, $A2$, $A3$, $A4$, $A5$, $A13$, $A14$ are available for this train in order to reach the exit at block section 14. The route of T_B is $B7$, $B8$, $B9$, $B10$, $B5$, $B6$ and the route of T_C is $C11$, $C8$, $C9$, $C10$, $C5$, $C6$. T_B and T_C share the same path from block section 8 to

6, which is crossed by the timetable default route of T_A on block section 9. T_B is a slow train entering block section 7 at release time 0. Its running time on each block section is 20 time units. T_A and T_C are fast trains requiring a running time of 10 time units on each block section. Their release time is 60 and 40, respectively. There are only two points that are relevant for the timetable, namely the exit of block sections 6 and 14. The timetable requires that T_A , T_B and T_C exit the network within time 131, 160 and 122, respectively. We observe that each train, individually, would be able to exit the network on time, i.e., the initial delay of each train at its relevant point is zero.

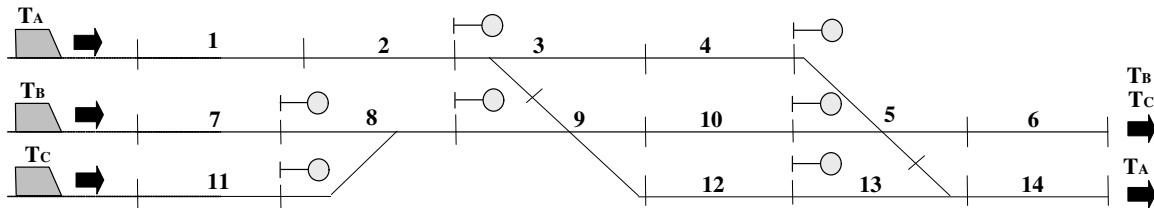


Figure 4.9: A small network with three trains

Figure 4.10 shows the alternative graph model of the CDRFR problem from the example of Figure 4.9. We denote a node with the pair (train, block section) of the associated operation or with the pair (train, exit point), except for dummy nodes. Alternative pairs are depicted using dashed arcs and for simplicity we assume their length (i.e., the setup time) always equal to zero. The three fixed arcs departing from node 0 model the release time of each train, whereas the arcs entering node n model the objective function. Note that the current alternative graph $\mathcal{G} = (N, F, A)$ has three isolated nodes (i.e., the nodes $A4$, $A5$ and $A10$) that can be used in order to implement alternative routes.

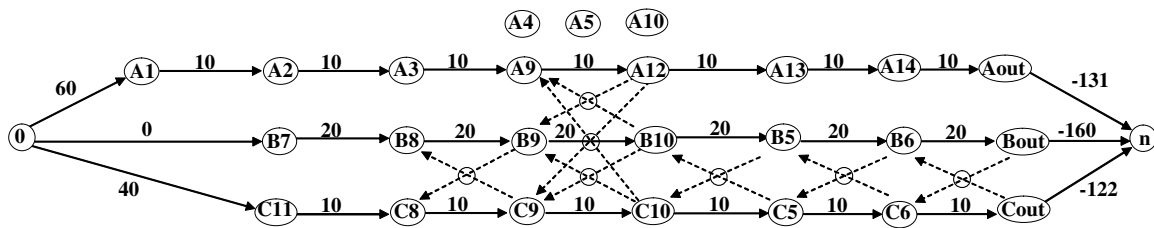


Figure 4.10: Alternative graph formulation of the proposed example

Figure 4.11 shows the optimal solution to the alternative graph of Figure 4.10. In this schedule T_B always precedes T_C , while T_A follows the other two trains on block section 9. The longest path passes through the nodes $0, B7, B8, B9, B10, B5, B6, Bout, C6, Cout$ and n . The optimal value $l^{F,S(F)}(0, n)$ is therefore 8.

Consider now the CDR problem, in which train routes (i.e., the set F of \mathcal{G}) can be changed, and let us choose the new route $A1, A2, A3, A4, A5, A13, A14$ for T_A (see the network of Figure 4.9). Clearly, a different alternative graph $\mathcal{G}' = (N, F', A')$ has to be adopted to represent the new CDRFR problem, in which T_A crosses the routes of T_B and T_C at block section 5. In the new alternative graph, the isolated nodes are $A9, A10$ and $A12$. Figure 4.12 shows the optimal solution to \mathcal{G}' , in which T_C precedes T_B , while T_A follows T_C and

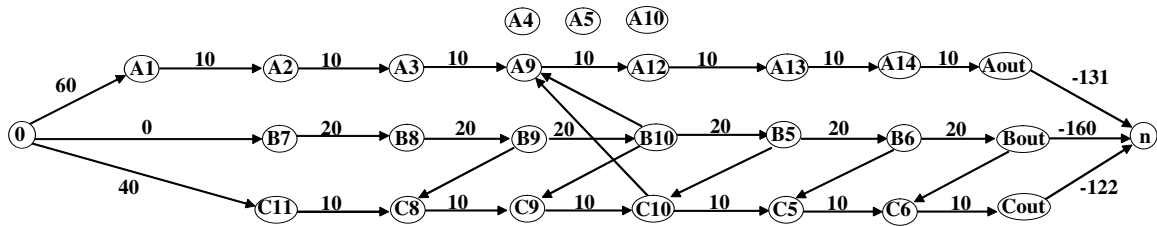


Figure 4.11: Optimal rescheduling solution of the proposed example

precedes T_B . The longest path, of length 0, passes through the nodes 0, C_{11} , C_8 , C_9 , B_8 , B_9 , B_{10} , B_5 , B_6 , B_{out} and n . This solution is also optimal for the compound CDR problem, i.e., $(F^*, S^*(F^*))$.

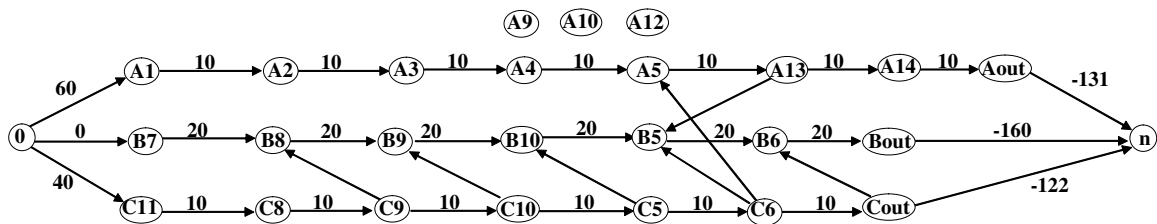


Figure 4.12: Optimal rescheduling and rerouting solution of the proposed example

4.4 Conclusions

In the traffic management of railway networks the problem of finding conflict-free and deadlock-free schedules is faced by railway practitioners both in the timetable design and during train operations. This chapter focuses on real-time traffic management aspects of the problem. When train operations are perturbed, a new timetable of feasible arrival and departure times is computed, such that deviations from the original one are minimized.

We propose a detailed discrete optimization model that can also be used to design optimal schedules. The problem is viewed as a job shop scheduling problem with no-store and other additional constraints. We make use of a careful estimation of space and time separation among trains, and model the overall scheduling problem (global conflict resolution) with the alternative graph formulation. This is able to efficiently treat the no-store aspect of the train scheduling problem for trains traveling in the same or opposite directions while ensuring the signaling and safety constraints.

It should be interesting to include further additional railway constraints in the traffic optimization model, such as constraints to the maximum allowed consecutive delay, stochastic running and setup times, different block signaling technologies, and others. In general, adding new constraints to the alternative graph model is not difficult but their effects still have to be evaluated in terms of quality of heuristic solutions, computational effort of branch and bound algorithms, practical consequences of flexible operations and so on.

Chapter 5

Train scheduling algorithms

This chapter describes the scheduling algorithms developed for this thesis and implemented in the ROMA dispatching support system. Since we intend to evaluate the potential benefit of effective procedures for solving CDRFR problems, we present three different classes of scheduling algorithms: simple dispatching heuristics, greedy heuristics and a specific branch and bound procedure. Dispatching heuristics simulate real-time control systems based on commonly used local rules while the other two classes of algorithms make use of specific properties of the alternative graph formulation of the CDRFR problem. The objective function is the minimization of the maximum consecutive delay at stations and at a set of relevant points of the dispatching area under study.

It is worthwhile noting that the alternative graph formulation of a practical size instance may include hundreds of machines (block sections) and tens of jobs (trains), resulting in a complex job shop problem with no-store and other additional constraints to be solved within the strict time limits imposed by the real-time nature of the problem. To this end, we exploit new properties of a feasible CDRFR solution, which allow the design of efficient implication rules. Such rules are used by the proposed scheduling algorithms to speed up the computation. A large set of computational experiments, based on a heavily congested area of the Dutch railway network, shows that a truncated version of the branch and bound algorithm is able to provide near-optimal solutions within a short time.

We also assess the effects of using flexible departure times in combination with various CDRFR algorithms, based on local and global rules. In what follows, we refer to a *rigid timetable* to mean the traditional one, in which all conflicts among trains are solved off-line and time reserves are used to absorb minor perturbations. We then refer to a *flexible timetable* to denote a timetable in which less details are fixed in the plan and more control decisions are left to the dispatcher. More flexibility to real-time traffic control should allow to improve the timetable reliability while enhancing the capacity of the lines. On the other hand, a computerized traffic management system is required to support the skills of human dispatchers. We therefore investigate the concept of flexible timetable, and the relations between flexibility, timetable robustness and delay minimization. An extensive analysis requires effective tools to manage real-time planned and unplanned traffic condi-

tions. We carry out an extensive computational study to analyze the influence of timetable flexibility starting from a rigid timetable. In order to make the comparison independent from the particular CDRFR algorithm, we use our branch and bound procedure to compare the optimal solutions for different delay scenarios and for varying the arrival and departure times of trains at stations. Finally, we evaluate the effects of timetable flexibility in terms of throughput, i.e., the number of trains per time interval.

The chapter is organized as follows. Section 5.1 presents CDRFR problem properties that can be used to reduce the computation time of the scheduling algorithms. Section 5.2 describes a lower bound algorithm to the CDRFR problem while Sections 5.3 and 5.4 deal with CDRFR heuristics. Section 5.5 gives a detailed description of the branch and bound algorithm. Moreover, Section 5.6 illustrates the formulation and optimal resolution of CDRFR problems with and without flexible departure times. In Section 5.7, computational experiments are reported to compare the different train scheduling algorithms and to evaluate the effectiveness of flexible timetables.

5.1 Static and dynamic implications

A key in the reduction of the computational effort of CDRFR problems based on the job shop scheduling formulation is the concept of *implication rule* (or immediate selection (Brucker et al., 1994)). We identified two kinds of implication rules: *dynamic* and *static*. Dynamic rules are computed during the execution of the solution procedure. Given an upper bound (UB) to the value of the current best solution, a partial selection S and an unselected pair of alternative arcs $((i, j), (h, k)) \in A$, the rationale behind dynamic rules is that if a lower bound to any extension of $S \cup \{(h, k)\}$ is larger than or equal to UB, then arc (i, j) can be added to S . We therefore say that arc (i, j) is *implied* by S , i.e., arc (h, k) is forbidden and arc (i, j) is added to S . In other words, the idea is to prove that given a partial selection S , no improvement to the current best solution is possible if arc (h, k) is selected. On the other hand, static rules can be computed off-line on the basis of the physical track topology of the railway network. We next give a formal description of the implication rules considered in this thesis.

We first present four dynamic implication rules. The first two rules are specific for the alternative graph model (Mascis & Pacciarelli, 2002), while the two latter rules were designed by Carlier and Pinson (1989) for the disjunctive graph but apply as well to the alternative graph model. In what follows, tails q_i , heads r_i , processing times p_i and the values $l^S(0, i)$ and $l^S(i, n)$ are computed for all nodes $i \in N(F)$, as described in Chapter 4.

Proposition 5.1.1 *Given a selection S , if $((i, j), (h, k))$ is an unselected pair of alternative arcs and $l^S(k, h) + a_{hk} > 0$, then arc (h, k) is forbidden and arc (i, j) is implied by S .*

Proposition 5.1.2 *Given a selection S , if $((i, j), (h, k))$ is an unselected pair of alternative arcs and $l^S(i, j) \geq a_{ij}$, then arc (i, j) is called redundant and this is implied by S .*

Proposition 5.1.3 *Given a selection S , if $((i, j), (h, k))$ is an unselected pair of alternative arcs and $r_h + a_{hk} + q_k \geq UB$, then arc (h, k) is forbidden and arc (i, j) is implied by S .*

The fourth implication rule for detecting implied arcs requires some additional definition. Given a set K of operations to be processed on a specified machine, a subset J of K is called an *ascendant* set of an operation $o_c \in K \setminus J$, if:

$$\min_{o_j \in J \cup \{o_c\}} r_j + \sum_{o_j \in J \cup \{o_c\}} p_j + \min_{o_j \in J} q_j \geq UB.$$

Symmetrically, J is a *descendant* set of $o_c \in K \setminus J$ if:

$$\min_{o_j \in J} r_j + \sum_{o_j \in J \cup \{o_c\}} p_j + \min_{o_j \in J \cup \{o_c\}} q_j \geq UB.$$

Proposition 5.1.4 *If J is an ascendant set of o_c , all arcs $(\sigma(c), j), \forall o_j \in J$, are forbidden. If J is a descendant set of o_c , all arcs $(\sigma(j), c), \forall o_j \in J$, are forbidden.*

In general, dynamic implication rules are a key property in the solution of the job shop scheduling problem at optimality, and have been successfully used to solve both the case with infinite capacity buffers (see e.g. Carlier and Pinson (1989)) as well as that with blocking (no-store) constraints (see e.g. Mascis and Pacciarelli (2002)). However, some rules for the computation of dynamic implications may require excessive time when dealing with a large number of jobs and machines, as in practical size instances of the CDRFR problem. Specifically, rules 5.1.1 and 5.1.2 may require a significant computational effort when using the complete matrix of longest paths $l^S(i, j)$ for each pair of nodes $i, j \in N(F)$ (Mascis & Pacciarelli, 2002). On the other hand, Carlier and Pinson (1994) show that rules 5.1.3 and 5.1.4 can be adopted to provide a fast algorithm for finding a maximal set of arcs that can be immediately selected. In all the train scheduling algorithms implemented in this thesis, we therefore include only the latter rules for the computation of dynamic implications, both of which can be computed very efficiently.

In this thesis, we also design and implement static implication rules. These are effective for CDRFR problems, since they contribute significantly to the reduction of the computation time of our solution algorithms. This speed up can be computed in a pre-processing step, given the physical track topology of the railway network and the train patterns.

The following propositions are introduced to establish a link between the selection of arcs from different alternative pairs.

Proposition 5.1.5 *Consider a selection S and two unselected alternative pairs $((a, b), (c, d))$ and $((i, j), (h, k))$. If $a_{ab} + l^S(b, i) + a_{ij} + l^S(j, a) \geq 0$, then arc (h, k) is implied by selection $S \cup \{(a, b)\}$ and arc (c, d) is implied by selection $S \cup \{(i, j)\}$.*

Proof. The result immediately follows from the observation that $S \cup \{(a, b), (i, j)\}$ contains at least a positive length cycle. \square

Proposition 5.1.5 can be applied particularly with the empty selection S^\emptyset , when the graph $\mathcal{G}(\emptyset)$ is composed only of fixed arcs associated with the train routes. During the execution of the solution procedure, the selection of an arc causes the selection of the entire set of arcs implied by this arc. The next proposition expresses rule 5.1.5 in terms of the railway physical configuration.

Proposition 5.1.6 Consider two alternative pairs $((a, b), (c, d))$ and $((i, j), (h, k))$. Then, $a_{ab} + l^\emptyset(b, i) + a_{ij} + l^\emptyset(j, a) \geq 0$ if the following conditions hold.

1. Nodes b and i are associated with a train T_A and are connected by a directed path of fixed arcs, i.e., T_A executes o_b before o_i .
2. Nodes j and a are associated with a train T_B and are connected by a directed path of fixed arcs, i.e., T_B executes o_j before o_a .
3. T_A and T_B pass through the block sections $((a, b), (c, d))$ and $((i, j), (h, k))$ refer to.

Proof. The result immediately follows from Proposition 5.1.5. \square

For each alternative arc (i, j) , we therefore associate a list of implied alternative arcs $Stat(i, j)$ such that Propositions 5.1.5 and 5.1.6 are satisfied.

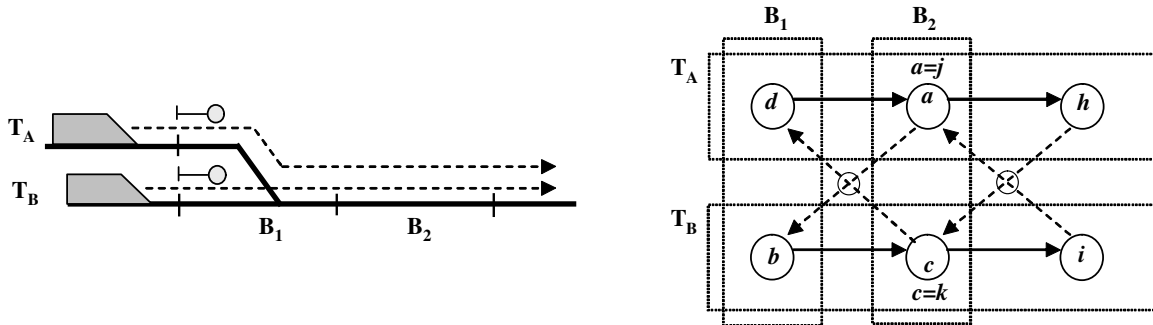


Figure 5.1: Trains traveling in the same direction

We now specify how the two static implication rules are used in our CDRFR algorithms. Let us use B_1 and B_2 to denote the two block sections associated with pairs $((a, b), (c, d))$ and $((i, j), (h, k))$, respectively. The conditions of Proposition 5.1.6 hold in the two situations:

- B_1 and B_2 are two adjacent block sections traversed by two trains, T_A and T_B , in the same order (see Figure 5.1). In this case, nodes a and j coincide, as well as nodes c and k . Arcs (a, b) and (h, k) imply each other ($Stat(a, b)$ contains (h, k) and $Stat(h, k)$ contains (a, b)), and arcs (c, d) and (i, j) imply each other ($Stat(c, d)$ contains (i, j) and $Stat(i, j)$ contains (c, d)).

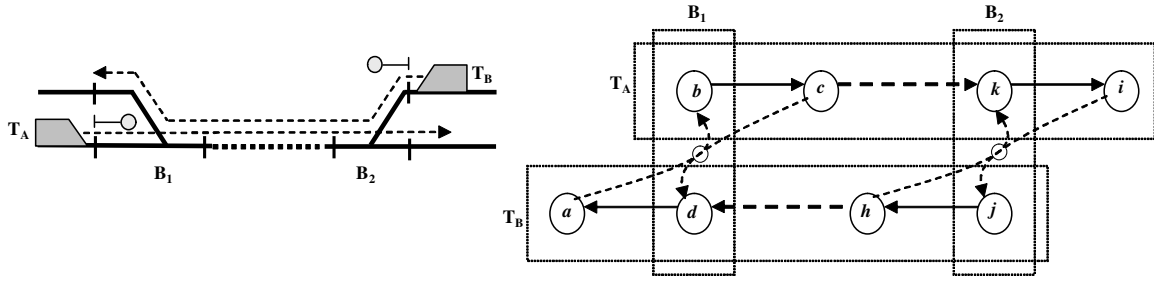


Figure 5.2: Trains traveling in opposite directions

- T_A and T_B pass both through B_1 and B_2 in opposite orders (see Figure 5.2). In this case, arc (a, b) implies arc (h, k) ($Stat(a, b)$ contains (h, k)) and arc (i, j) implies arc (c, d) ($Stat(i, j)$ contains (c, d)). Note that arc (h, k) does not imply arc (a, b) , and arc (c, d) does not imply arc (i, j) .

During the resolution of the CDRFR problem, if an alternative arc $(i, j) \in Stat(i, j)$ is selected in the current graph selection S then all the alternative arcs in the list $Stat(i, j)$ have also to be selected in S .

5.2 Lower bound algorithm

Carlier and Pinson (1994) describe an algorithm for computing a maximal set of implied arcs with complexity $O(|K| \log |K|)$, where K is the set of operations to be processed on a given machine. Their algorithm is based on a single machine scheduling problem with heads, tails and preemption, and computes the so called *Jackson Preemptive Schedule* (JPS) (Jackson, 1955). For the interested reader, a detailed description of the single machine JPS can be found e.g. in Carlier (1982) and Pinson (1990). The single machine JPS permits efficient adjustments of heads and tails, and returns a lower bound to the length of the longest path from node 0 to node n for any extension of S . This lower bound has also been successfully adopted in Mascis and Pacciarelli (2002) for the resolution of job shop scheduling problems with no-store and no-wait constraints.

In this thesis, we adapt the single machine JPS to deal with the alternative graph formulation of CDRFR problems. The idea is to view a block section as a machine, which can process at most one train at a time for a given processing time. Each train cannot enter the block section before a known release time, and the maximum consecutive delay cannot be smaller than the block section exit time, plus a known tail. Release times, processing times and tails are evaluated on a partially selected graph $\mathcal{G}(S)$. In general, a lower bound to the maximum consecutive delay is computed in $O(z \log z)$ time units for each block section, with z equal to the number of trains traversing the current block section.

5.3 Dispatching heuristics

In this thesis, we consider dispatching rules that may be adopted by traffic controllers. These heuristics order the running trains at each railway junction by means of simple and local decision criteria. We next present some examples among several possible alternatives.

The First Come First Served (FCFS) dispatching rule, often used in the railway practice, solves train conflict situations by assigning each block section to the first train requiring it, which is simply to give precedence to the train arriving first at a block section. This rule requires no dispatching action since trains pass at merging or crossing points on the basis of their actual order of arrival and not necessarily as planned in the timetable.

The second dispatching rule, called First Leave First Served (FLFS), is as follows. When two trains claim the same block section, we first compute the time required for each train to enter and traverse it. Precedence is then given to the train that would leave the block section first. FLFS is a compromise between two commonly used dispatching rules of (i) giving priority to the fast trains over the slow ones and (ii) giving precedence to the train arriving first. Note that the proposed rule may lead to a different train sequence with respect to the one specified by the timetable.

We have also implemented two priority rules, called PRULE1 and PRULE2. PRULE1 is based on two priority levels as follows. A first level gives precedence to the train type: first intercity, then local and finally freight. In case of a tie, a second level gives precedence to the train with the smallest number of scheduled stops after the conflicting point. PRULE2 is based on the entrance train delays, i.e., gives precedence to the train with the larger train delay recorded at the entrance of the studied railway area. It has to be observed that the latter rule is usually only adopted in practice if strongly delayed trains are of a significantly higher priority than trains running on time. Besides, in case of a conflict between trains having equal priority values, both PRULE1 and PRULE2 apply the FCFS rule.

We finally describe a simple dispatching algorithm that simulates the practice of railway traffic management adopted in the Netherlands. This is based on the Dutch automatic route setting system called ARI (Berends & Ouburg, 2005). When train delays are confined in a predefined time-window (threshold), ARI detects and solves automatically train conflicts according to the following precedence rules:

- If the conflicting trains claim the same route segment at the same time, then the scheduled order is kept;
- If the conflicting trains claim different but incompatible track segments, then precedence is given to the train that requested its route first.

When the delay of an approaching train exceeds the threshold, the traffic controllers must set up the routes of the conflicting trains manually on the basis of a list of what-if scenarios, and on their intuition and experience.

In order to evaluate the effectiveness of a completely automated decision support system, we implemented an automated version of the ARI system that simulates the dispatcher behavior as follows. When the entrance train delays are within the predefined time-window, the two above ARI rules are used. Otherwise, PRULE1 is applied.

5.4 Greedy heuristics

Greedy heuristics based on global information have recently been developed to find a feasible schedule for job shop scheduling problems modeled with alternative graphs (see e.g. Mascis and Pacciarelli (2002) and Meloni et al. (2004)). In this thesis, we develop a greedy heuristic similar to the AMCC (Avoid Most Critical Completion Time) algorithm described in Mascis and Pacciarelli (2002). The idea is to forbid one alternative arc at a time, the one which would cause the largest increase in consecutive delay, and to select its companion (see Figure 5.3).

Algorithm AMCC

Input: An initial selection $S = S^I$,

Begin

While $A \neq \emptyset$ **do**

Begin

 Let $C_{\max}(S) = \max_{(u,v) \in A} \{l^S(0, u) + a_{uv} + l^S(v, n)\}$,

 Choose a pair $((i, j), (h, k)) \in A$ such that $l^S(0, h) + a_{hk} + l^S(k, n) = C_{\max}(S)$,

 Let $S' := S \cup \{(i, j)\}$, $A' := A \setminus \{((i, j), (h, k))\}$,

 Select all arcs in the set $Stat(i, j)$,

If (there is a cycle in $\mathcal{G}(S')$) **or** (an arc from $Stat(i, j)$ is forbidden) **then**

Begin

 Let $S' := S \cup \{(h, k)\}$, $A' := A \setminus \{((i, j), (h, k))\}$,

 Select all arcs in the set $Stat(h, k)$,

If (there is a cycle in $\mathcal{G}(S')$) **or** (an arc from $Stat(h, k)$ is forbidden) **then**

Begin

 The procedure failed in computing a feasible solution, **exit**

End

End

$S := S'$, $A := A'$.

End

End

Figure 5.3: Sketch of the AMCC heuristic

At each step of the solution procedure, AMCC enlarges a partial selection S , initially set equal to S^\emptyset , by choosing a pair of unselected alternative arcs $((i, j), (h, k))$ such that the quantity $l^S(0, h) + a_{hk} + l^S(k, n)$ is maximum among all the unselected alternative arcs.

Selecting arc (h, k) would therefore increase $l^S(0, n)$ more than any other unselected alternative arc. For this reason, AMCC forbids this arc and selects its alternative, i.e., arc (i, j) and all arcs in its set of static implications $Stat(i, j)$ are added to S . The difference with the AMCC presented in Mascis and Pacciarelli (2002) is that here we consider only static implications and not dynamic ones. This results in a significant reduction of the computation time of the heuristic. Therefore, large rescheduling problems can be solved by AMCC within a few seconds (see Section 5.7).

5.5 Branch and bound algorithm

Branch and bound search has been known for a long time and has been widely used in solving a variety of problems, such as integer linear programming and combinatorial optimization problems. In general, the goal in such problems is to systematically search a very large space to find a globally optimal solution. Since the search space is intractably large, some implicit method is required in order to rule out regions of the search space that contain no interesting solutions. Branch and bound algorithms work by the divide and conquer principle: the search space (enumeration tree) is subdivided into smaller subregions (this subdivision is referred to as branching), and bounds are found on all the solutions contained in each subregion under consideration. The strength of the branch and bound approach comes when bounds on a large subregion show that it contains only inferior solutions, and so the entire subregion can be discarded (pruned) without further examination. Ideally the procedure stops when all the search space is explored. At that point, all non-pruned subregions will have their bounds equal to the global minimum of the function (proven optimum). In practice the procedure is often terminated after a given time (near-optimum).

In this thesis, a novel BB algorithm has been designed and implemented to compute near-optimal solutions to CDRFR problems. This algorithm, sketched in Figure 5.4, is an advanced version of the one described in Mascis and Pacciarelli (2002), including a number of variants, particularly effective for solving CDRFR problems. These variants enable a substantial reduction of the computation time. However, due to the inherent complexity of the CDRFR problem, a truncated version of the BB algorithm is adopted in our experiments. We next describe the main algorithmic components while the computational results will be discussed in Section 5.7.

At each node of the enumeration tree, the BB algorithm uses the single machine JPS (introduced in Section 5.2) for computing a lower bound and possibly pruning the enumeration tree, as well as static and dynamic implication rules (described in Section 5.1) for speeding up the computation. The lower bound is computed for each block section and is used to evaluate partial graph selections. In Figure 5.4, the resulting value for a selection S is indicated as $JPS(S)$.

The implication rules are another key factor to speed up the BB algorithm, since they reduce the number of branches needed to prove the optimality of a solution. Specifically, Proposition 5.1.6 is used in a preprocessing phase of the BB algorithm to compute static

Algorithm Branch&Bound**Input:**

An alternative graph $\mathcal{G} = (N, F, A)$,

An initial selection S^I ,

Begin

Compute sets $Stat(i, j)$ for all alternative arcs,

For all $(i, j) \in S^I$ **do** $S^I = S^I \cup Stat(i, j)$,

$UB = BestSol$ found by FCFS, FLFS and AMCC,

$L = \{S^I\}$,

While $(L \neq \emptyset)$ **and** (time limit not reached) **do**

Begin

Extract a selection S from L ,

Choose an unselected pair $((i, j), (h, k))$ from $\mathcal{G}(S)$,

$S' = S \cup \{(i, j)\} \cup Stat(i, j)$,

While $(JPS(S') < UB)$ **and** (there are arcs dynamically implied by S') **do**

Begin

Add to S' all arcs implied by $JPS(S')$ and their static implications,

End

If S' is complete **then**

Begin

If $l^{S'}(0, n) < UB$ **then** $UB = l^{S'}(0, n)$,

End

Else add S' to L ,

$S'' = S \cup \{(h, k)\} \cup Stat(h, k)$,

While $(JPS(S'') < UB)$ **and** (there are arcs dynamically implied by S'') **do**

Begin

Add to S'' all arcs implied by $JPS(S'')$ and their static implications,

End

If S'' is complete **then**

Begin

If $l^{S''}(0, n) < UB$ **then** $UB = l^{S''}(0, n)$,

End

Else add S'' to L ,

End

If time limit not reached **then** $OptSol = UB$.

End

Figure 5.4: Sketch of the branch and bound algorithm

implications $Stat(i, j)$ for each unselected alternative arc $(i, j) \in A$. We therefore associate to each alternative arc the set of all arcs implied statically. Static implications are then utilized during the execution of initial heuristics, as well as during the search procedure.

The BB initial solution (upper bound) is obtained by running some of the heuristics described in Sections 5.3 and 5.4. In our computational experience, we test dispatching rules FCFS and FLFS and greedy heuristic AMCC as initial solutions. The solutions found by AMCC are almost always better than those of the two dispatching rules. On the other hand, while the computation time of the two dispatching rules is negligible, AMCC is significantly slower and fails to find feasible solutions more frequently. Hence, none of the three heuristics outperforms the others overall. For these reasons, the initial value of the upper bound (UB in Figure 5.4) is always set equal to the best solution ($BestSol$) obtained by the three heuristics.

Given an initial selection S^I , possibly empty, the BB algorithm builds an extension of S^I by minimizing the length of the longest path from node 0 to node n . A list L , initially equal to S^I , contains the set of all partial selections. At each step of the BB algorithm, a selection S is chosen from L . A partial selection S is added to L if an extension of S improving the current optimum UB exists. An unselected alternative pair $((i, j), (h, k))$ is then picked up from S in order to produce the two selections $S' = S \cup \{(i, j) \cup Stat(i, j)\}$ and $S'' = S \cup \{(h, k) \cup Stat(h, k)\}$. Moreover, Propositions 5.1.3 and 5.1.4 are exploited to compute dynamic implications and to enlarge the selections S' and S'' . This step is repeated until it is not possible to enlarge them anymore. Whenever some arcs are selected with this procedure, new values of heads and tails of all operations are computed on the graphs $\mathcal{G}(S')$ and $\mathcal{G}(S'')$, and a further set of arcs might be implied. After this, the single machine JPS returns a lower bound (LB) on any extension of each of the selections S' and S'' . Each incomplete selection with LB smaller than the current optimum (UB) is finally added to L . The procedure terminates when a time limit of computation is reached or when $L = \emptyset$. In the latter case, UB corresponds to the proven optimum ($OptSol$).

We study four binary branching schemes to choose an unselected pair $((i, j), (h, k))$ from $\mathcal{G}(S)$ and use the following evaluation criteria: (i) the AMCC rule, (ii) the LB computed with the single machine JPS and (iii) the number of static implications. Specifically, we tested four branching rules.

1. Choose the unselected pair $((i, j), (h, k))$ such that $l^S(0, i) + a_{ij} + l^S(j, n)$ is maximum and branch on this pair.
2. Among all unselected alternative arcs, consider those with the value $|Stat(i, j)|$ maximum, and choose among them the one maximizing $l^S(0, i) + a_{ij} + l^S(j, n)$. Branch on the pair (i, j) belongs to.
3. Given the current selection S , consider only those unselected arcs with the value $|Stat(i, j)| \geq 10$ (empirically chosen), and compute $JPS(S \cup (i, j) \cup Stat(i, j))$. Choose the arc (i, j) that minimizes such value. Branch on the pair (i, j) belongs to. If $|Stat(i, j)| < 10$ holds for all the unselected arcs, use the first branching rule.

4. As the previous one, but choose the arc (i, j) that maximizes LB.

We also analyze eight search strategies generated by the BB algorithm for exploring the enumeration tree, i.e., for extracting a selection S from L .

DepthFirst. Extract first the selection S inserted last in L . If the last branch of the searching procedure inserted both the resulting selections (S' and S'') in L , then if $JPS(S') \leq JPS(S'')$ extract first S' else extract S'' .

BreadthFirst. Extract the selection added first to L .

BestFirst. Extract the selection S with the smallest value of $JPS(S)$.

WorstFirst. Choose the open node with the largest value of $JPS(S)$.

Hybrid1. Alternate three repetitions of the depth-first visit with the choice of a selection S with smallest value of $JPS(S)$ among the last nine selections inserted in L .

Hybrid2. Alternate three repetitions of the depth-first visit with the choice of a selection S with smallest value of $JPS(S)$ among the last nine selections inserted in L .

Hybrid3. Perform depth first until at least one of the two selections, S' and S'' , generated in the last branch of the searching procedure is added to L , otherwise choose a selection S according to strategy BestFirst.

Hybrid4. Perform depth first until at least one of the two selections, S' and S'' , generated in the last branch of the searching procedure is added to L , otherwise choose a selection S according to strategy WorstFirst.

We then performed a preliminary set of experiments on six hard instances of the CDRFR problem. Each instance corresponds to an alternative graph with a number of alternative pairs between 8060 and 33451. We tested all the 32 combinations of branching schemes and search strategies of the enumeration tree. The best BB configuration for this set of experiments, in terms of computation time and number of branches, proved to be the first branching rule and search strategy Hybrid1. In our computational experiments, we therefore choose this BB configuration to select the most promising node of the enumeration tree.

5.6 Illustrative example of optimal CDRFR solutions

We present optimal CDRFR solutions for a timetable with flexible departure times. The effects of flexible departure times are reported for a small railway network with three trains, depicted on the top of Figure 5.5. In this network, there are seven block sections and a station. For clearness, we only show the location of the most relevant block signals. The timetable is shown in Table 5.1. At time $t = 0$, the trains are already in the network. T_A

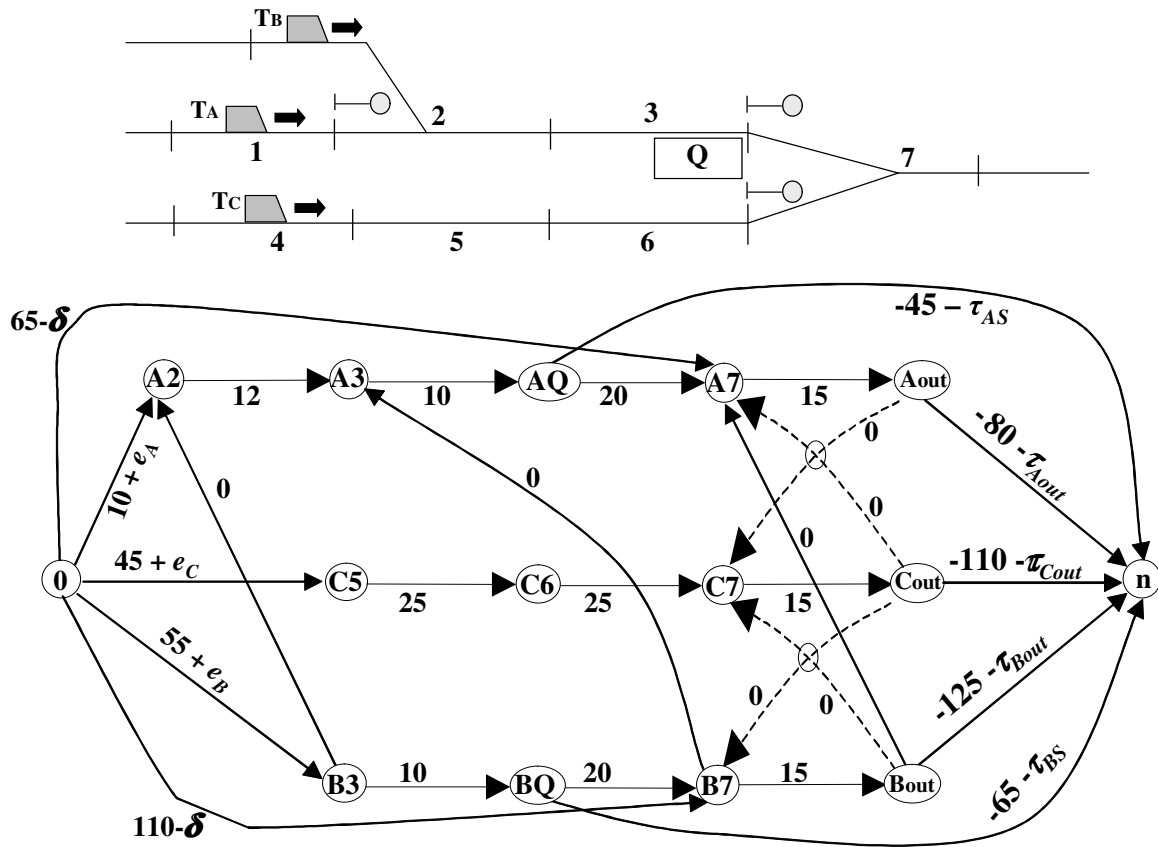


Figure 5.5: A railway network and the alternative graph formulation

is a passenger train, running from block section 1 to block section 7 with a scheduled stop Q in a station. T_B is a passenger train running from block section 2 to block section 7 with the scheduled stop Q . T_C is a freight train running from block section 4 to block section 7.

The alternative graph formulation is shown at the bottom of Figure 5.5. We recall that a train entering a block section is modeled by a node of the graph. For example, $A3$ and $B3$ are different nodes referring to block section 3. T_A and T_B share block sections 2, 3 and 7. T_C shares only block section 7 with T_A and T_B . For each couple of trains sharing a block section, there is a pair of alternative arcs representing the decision to be taken on the precedence relation between the involved trains. Arcs $(0, A2)$, $(0, B3)$ and $(0, C5)$ represent the time needed for each train to reach the end of the current block section, which is the off-line scheduled entrance in the next block section plus the entrance delay e_A , e_B , and e_C , respectively. Arcs $(0, A7)$ and $(0, B7)$ represent the scheduled departure time from Q . Finally, arcs (AQ, n) , $(Aout, n)$, (BQ, n) , $(Bout, n)$ and $(Cout, n)$ represent the modified due dates of the trains at the station and at the exit point of the network (the end of block section 7). The length of all horizontal arcs corresponds to the running time of a train on a block section. For the sake of simplicity, all the setup times (alternative arcs lengths) are set equal to zero in this example. Note that the initial position of T_B implies that T_A is not allowed to precede T_B on block sections 2, 3 and 7, and we therefore have the selected alternative arcs $(B3, A2)$, $(B7, A3)$ and $(Bout, A7)$.

The optimal solutions of the CDRFR problem are described in Table 5.1 as follows. The

Passing Time	Timetable			OptSol $\delta = 0$			OptSol $\delta = 10$			OptSol $\delta = 25$		
	T_A	T_B	T_C	T_A	T_B	T_C	T_A	T_B	T_C	T_A	T_B	T_C
t_1	0			47			47			47		
t_2	10	45		57	45		57	45		57	45	
t_3	35	55		110	55		100	55		85	55	
t_Q	45	65		120	65		110	65		95	65	
t_4			30			47			47			47
t_5			45			62			62			62
t_6			70			87			87			87
t_7	$65-\delta$	$110-\delta$	95	140	110	125	130	100	115	115	85	130
t_{out}	80	125	110	155	125	140	145	115	130	130	100	145
Exit Delay												
Initial	0	0	0	34	0	17	34	0	17	34	0	17
Consecutive	0	0	0	41	0	13	31	0	3	16	0	18
Total	0	0	0	75	0	30	65	0	20	50	0	35
Station Delay												
Initial	0	0	0	34	0		34	0		34	0	
Consecutive	0	0	0	41	0		31	0		16	0	
Total	0	0	0	75	0		65	0		50	0	

Table 5.1: Optimal solutions of the CDRFR problem for three values of δ

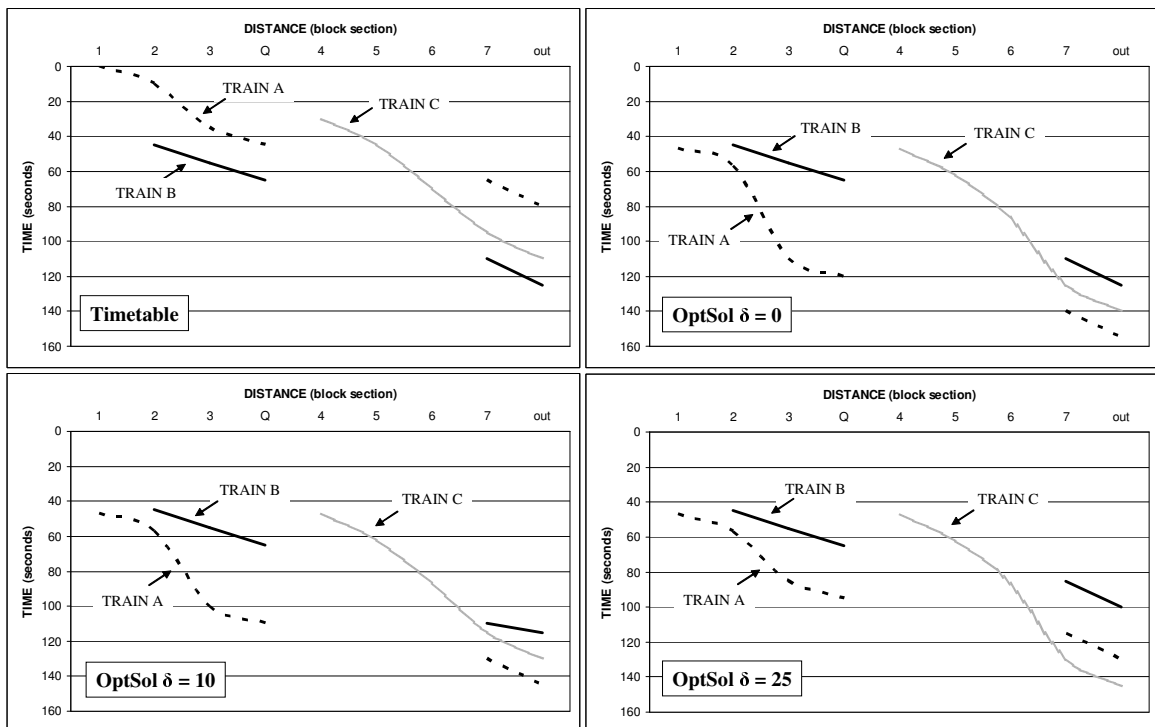


Figure 5.6: Graphical timetables for each of the proposed CDRFR solutions

values t_u , for $u = 1, \dots, Q, \dots, out$, are associated with the time a train should enter a block section B_u or start the dwell time at Q . In columns 2, 3 and 4, the timetable for the three trains is reported. These are the off-line scheduled times at which each train should enter each block section. The off-line exit order is $T_A-T_C-T_B$. In columns 5 to 13, the optimal solutions of the CDRFR problem are reported for three different values of δ . The flexibility varies from $\delta = 0$, i.e., the rigid timetable, to $\delta = 10$ and $\delta = 25$. In the three cases, T_A enters the network with a delay $e_A = 47$, T_B is on time ($e_B = 0$) and T_C enters the network with a delay $e_C = 17$. The optimal output sequence is $T_B-T_C-T_A$ for $\delta = 0$ and $\delta = 10$, while this is $T_B-T_A-T_C$ for $\delta = 25$. The initial delay for each train at each relevant point is computed as the shortest path in the graph of Figure 5.5, disregarding the presence of other trains (i.e., without considering the alternative arcs). Precisely, $\tau_{AQ} = \tau_{Aout} = 34$, $\tau_{BQ} = \tau_{Bout} = 0$ and $\tau_{Cout} = 17$. In the three cases, the maximum [average] consecutive delay is 41, 31 and 18 [is 19, 13 and 10] for $\delta = 0$, $\delta = 10$ and $\delta = 25$, respectively. Figure 5.6 illustrates the graphical timetables for this example.

The main effect of timetable flexibility is to allow T_B (that is on time) to depart earlier from the station, and T_A (that is late) to recover until its scheduled stop and somewhat to leave the station with a reduced delay. For increasing values of timetable flexibility, T_A is able to recover a bigger part of its delay, thus enabling a faster restoration of the scheduled order between T_A and T_C .

This example shows that flexible departure times cannot reduce initial delays. The scheduled running time reserves and dwelling buffer times of preceding trains can be used to reduce consecutive delays. A necessary condition for flexible timetables to be effective in practice is that sufficient recovery times are included. If the timetable is designed in such a way that no train can reach or leave a station earlier than originally scheduled, then a larger flexibility of arrival and departure times cannot generate significant effects.

5.7 Computational experiments

This section presents computational experiments on disturbed train traffic situations in order to allow a comparison between local and global CDRFR strategies. These experiments are based on a large sample of practical size instances derived from the Schiphol dispatching area (Appendix A). This is a 20 km long bottleneck area of the Dutch railway network that includes the underground station of Schiphol, beneath the international airport of Amsterdam, and the neighbor stations of Nieuw Vennep, Hoofddorp, Amsterdam Lelylaan and Amsterdam Zuid. We study the network by simulating real-time traffic conditions for different timetables and entrance perturbations, and computing a rescheduling of train operations each time. We do not consider other perturbations, such as temporary unavailability of some resources (trains, stations, block sections, etc.). Scheduling algorithms are executed on a laptop equipped with an Intel Pentium M processor (1.6 GHz), 512 MB Ram and Linux operating system.

5.7.1 Evaluation of CDRFR algorithms

Given a provisional hourly timetable for the year 2007 (Hemelrijk et al., 2003), we consider a first set of 60 instances that are divided in six groups of ten instances each. Each group differs for the sets of delayed trains and values of the delays. The number of delayed trains in each group is the same, varying between 7 and 27. The delayed trains are randomly chosen within the first half hour of the hourly timetable in order to study the propagation of train delays within one and two hours of traffic prediction. The value of the entrance delay is generated on the basis of Gaussian and uniform distributions and is randomly chosen in a time window of typical train delays.

Starting from the first set of 60 instances, we generate a second set of 240 more difficult instances to better evaluate the performance of the BB algorithm. For each perturbation scheme, we introduce four variations of the timetable by using flexible departure times for all trains at each station stop (as formulated in Chapter 4). This is obtained simply by anticipating the scheduled departure times of the same amount, chosen in the set $\{30, 60, 90, 120\}$ seconds, and keeping constant the scheduled arrival times. Such timetable modifications enlarge the set of feasible solutions to the CDRFR problem, since the constraints on the scheduled departure time are relaxed, thus making the instances potentially more difficult to solve for the BB algorithm.

We denote each instance with a four field code $A - B - C - D$ as follows: Under A we denote the number of delayed trains; B indicates the distribution type: Uniform (U) or Gaussian (G); C represents the range $[0, max]$ in which we choose the delay values, with max selected from the set $\{200, 300, 400, 600, 800, 1000, 1400, 1800\}$ seconds; D indicates the type of timetable that varies from 1 to 5, with 1 being the original timetable.

In the first two sets of instances (300 instances), the number of trains is 54 corresponding to an alternative graph with more than 8000 pairs of alternative arcs. In order to further evaluate the limits of the BB algorithm, we also generated a third set of 300 instances by enlarging the traffic prediction up to two hours (two hour timetable) for each instance of the first two sets. This means to solve a graph with more than 30000 pairs of alternative arcs, i.e., a CDRFR instance with 108 trains.

For all the three sets of instances (600 instances), we compute the performance of the three initial heuristics and of the BB algorithm, truncated after 120 seconds of computation. This limit of maximum computation time assures the algorithms compatibility with the needs of real-time railway applications.

The experiments of this subsection are partitioned into four test phases. The first two phases report on the performance of the three initial heuristics (i.e., FCFS, FLFS and AMCC) and of our BB algorithm. The third phase demonstrates the benefits of static implications while the last phase describes elaborately the most time consuming instances.

Effects of the AMCC heuristic

We first discuss the time performance of the three initial heuristics in our BB procedure. FCFS and FLFS require a negligible computation time, whereas AMCC needs on average 0.5 seconds and 14.6 seconds on one hour and two hour instances, respectively.

Since AMCC is the most time consuming heuristic, we now analyze its influence and actual benefits as initial heuristic of the BB algorithm. Table 5.2 describes the BB solutions obtained with and without using AMCC. Each row of this table gives the average behavior over 300 instances. The time horizon of traffic prediction is given in column two (i.e., one hour or two hour instances). The average number of iterations required to obtain the best solution and the average total number of iterations are shown in columns three and four, respectively. The two subsequent columns show the average computation time required to reach the best solution and to prove optimality. Finally, the last column reports the number of proven optimal solutions computed by the BB algorithm within the time limit of computation.

AMCC Used	Time Horizon	Iterations		Time (sec.)		# Optimal Solutions
		Best	Total	Best	Total	
Yes	1h	9.49	116.63	0.69	1.93	297/300
Yes	2h	31.77	92.20	17.17	20.13	294/300
No	1h	149.45	387.49	2.58	4.39	297/300
No	2h	358.58	794.34	26.74	36.70	294/300

Table 5.2: Effects of the AMCC heuristic

In general, a good initial upper bound is very beneficial for the BB algorithm. As reported in Table 5.2, the use of AMCC as initial heuristic considerably reduces the computation time for both sets of one hour and two hour instances. In addition, when using AMCC the number of iterations to compute the best solution is strongly reduced and the BB algorithm found the same number of proven optimal solutions within less total iterations.

Branch and bound algorithm versus initial heuristics

In the second test phase, we evaluate the 300 instances with the one hour timetable. For each instance, we compute the performance of the three initial heuristics and the BB algorithm. In Figure 5.7, we present the solutions obtained in terms of maximum and average consecutive delays. In fact, we do not consider initial delays since they depend only on the entrance delay and cannot be avoided by dispatching measures. For each of the four considered algorithms, a point (X, Y) in the curve corresponds to the number of instances Y with a consecutive delay smaller or equal to X over the 300 instances.

As expected, BB outperforms clearly the other algorithms in terms of consecutive delay minimization. The improvement with respect to the FCFS and FLFS dispatching rules confirms the advantage of global conflict resolution in the considered network even within the

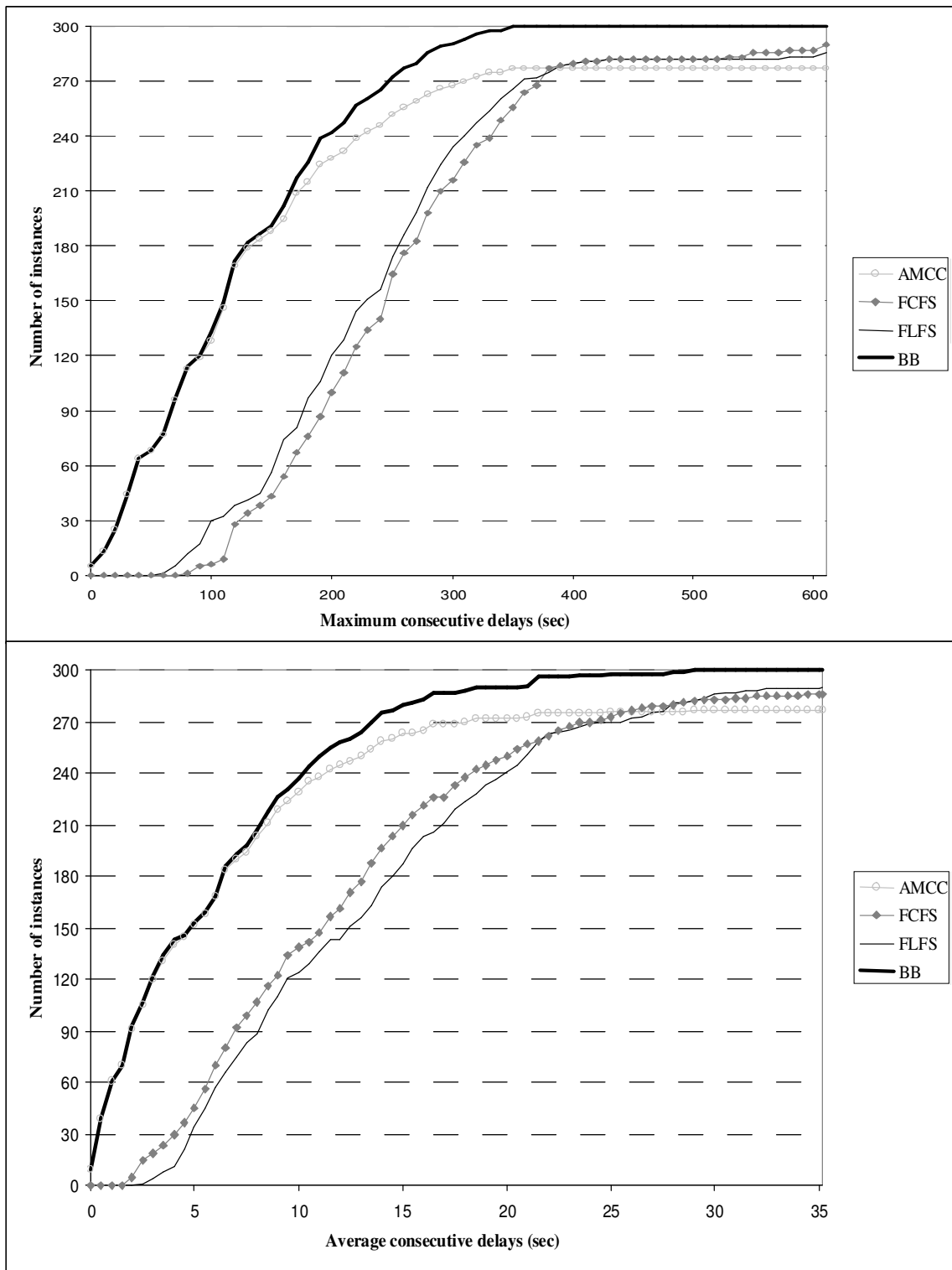


Figure 5.7: Cumulative plots of maximum and average consecutive delays

strict time limits imposed by real-time purposes. The BB algorithm reduces the amount of maximum consecutive delay by approximately 100 seconds compared to the FCFS and FLFS dispatching rules. Unexpectedly, the BB algorithm also exhibits better behavior when considering the average delays. The total amount of the average consecutive delay is approximately 50% less compared to FCFS and FLFS.

The BB algorithm finds proven optimal solutions in 297 cases out of 300 within the time limit of 120 seconds. The average number of nodes in the enumeration tree is 116, and the average computation time is 1.93 seconds. Also for the three remaining instances (near-optimal solutions), the BB algorithm is able to improve the solutions found by the initial heuristics. For all the studied heuristics, the deviation from the lower bound is more than 85%, while this deviation decreases to 26% when using the BB algorithm.

The FCFS and FLFS dispatching rules, being the most simple “local” rules, give the worst results. While the worst solution found by the BB algorithm exhibits a maximum consecutive delay of 350 seconds, the worst solutions calculated by the dispatching rules have more than 600 seconds of consecutive delay. Thus, simple dispatching rules may, in general, lead to poor results. Moreover, since the Schiphol network and train schedule may allow for deadlocks, the heuristics do not guarantee a feasible solution. FCFS finds a feasible solution 290 times out of 300, FLFS 286 times out of 300, AMCC 277 times out of 300 instances. However, when AMCC finds a feasible solution, this algorithm is very close to optimality and almost always better than the solutions found by FCFS and FLFS.

Effects of static implications

In the third test phase, we show the effects of static implications for the CDRFR problem, both in the initial heuristics as well as in the BB algorithm. The experiments are carried out on the first set of 60 instances with 54 trains and the hourly timetable, and on other 60 instances with 108 trains, obtained by replicating the timetable in the second hour.

Table 5.3 describes the performance of our algorithms when using static implications both in the initial heuristics and in the BB algorithm, compared to the case in which static implications are not adopted either in the initial heuristics or in the BB algorithm. The rows of this table give the average behavior over 60 instances. Columns 2-7 of Table 5.3 are as in Table 5.2 while the last column reports the number of feasible solutions obtained by the BB algorithm.

The advantage of using static implications in the resolution of the CDRFR problem is evident by comparing the rows of Table 5.3. Furthermore, the main differences between the one hour instances and the two hour instances are in the increased number of iterations and in the longer CPU time.

When using static implications, the BB algorithm is always able to attain and prove optimality, and only requires a few steps in order to terminate each run. When static implications are not used, the BB algorithm is unable to reach the optimal solution in more than 50% of the instances and, in some cases, is not even able to find a feasible solution. In fact, if the

Static Implications	Time Horizon	Iterations		Time (sec.)		# Optimal Solutions	# Feasible Solutions
		Best	Total	Best	Total		
Yes	1h	7.3	15.1	0.66	0.74	60/60	60/60
Yes	2h	17.2	34.6	16.15	18.11	60/60	60/60
No	1h	112.3	7316.9	25.45	67.76	28/60	50/60
No	2h	0.0	21.0	117.68	119.40	8/60	38/60

Table 5.3: Effects of static implications

three heuristics do not use static implications the number of times in which these heuristics find a feasible solution decreases significantly.

However, when a two hour timetable is considered the algorithm without dynamic implications is unable to find a feasible solution in about 40% of instances, and is only able to prove optimality in a few cases.

We finally report that the time limit of 120 seconds allows the exploration of a limited number of nodes (see columns “Iterations”) when solving the two hour instances. This is partially due to the longer time required by the three initial heuristics in order to compute a feasible solution without using static implications.

“Hard instances”

In the fourth test phase, we describe the most critical instances, in terms of computation time, over the three sets of experiments with one and two hours of traffic prediction.

We initially stated that the CDRFR problem is a very big no-store job shop problem, but real-world railway instances are, in most cases, easy to solve. In the vast majority of the instances, the BB procedure closes the problem at its root or after a single branching of the enumeration tree. The optimal solution almost always coincides with the lower bound. Moreover, AMCC is very effective in computing good solutions. However, there are also some difficult instances.

We classify an instance *hard* if the BB algorithm requires more than two nodes of the enumeration tree for finding the proven optimum. Specifically, there are only two hard instances from the first set of 60 instances (one hour timetable), and in both cases the optimal solution equals the lower bound. For these two instances, the optimal solution is proved by our branch and bound algorithm in less than five seconds. There are 12 hard instances from the second set of 240 instances (one hour timetable) and 24 hard instances in the third set of 300 instances (two hour timetable). We next restrict our attention to these 38 hard instances and enlarge the time limit of the BB algorithm up to 7200 seconds.

Table 5.4 reports the performance of the BB algorithm on the hard instances. In the first column, the code of the instance is given with the $A - B - C - D$ fields of Section 5.7.1. Columns two, three and four show the lower bound, the best solution found by AMCC,

Perturbation Identifier	Lower Bound	Initial Solution	Final Solution	Iterations		Time (sec.)	
				Best	Total	Best	Total
One Hour Instances							
7-G-800-3	125	165	125*	146	146	3.1	3.1
7-G-800-4	95	145	95*	144	144	3.0	3.0
7-G-800-5	65	145	65*	151	151	3.1	3.1
26-G-600-1	48	310	162*	199	387	3.9	4.8
11-U-1000-1	212	320	212*	239	239	4.6	4.6
11-U-1000-2	182	290	182*	232	232	4.4	4.4
11-U-1000-3	152	260	175*	229	58498	4.4	656.4
11-U-1000-4	122	248	155	236	494563	4.4	7200.0
11-U-1000-5	114	207	155*	207	46624	4.0	575.4
11-G-1000-5	151	215	151*	218	218	4.3	4.3
16-U-1800-2	237	343	237*	248	248	4.6	4.6
16-U-1800-3	207	262	207*	189	189	3.6	3.6
16-U-1800-4	177	262	177*	203	203	3.9	3.9
16-U-1800-5	177	262	177*	207	207	4.0	4.0
Two Hour Instances							
7-G-800-2	155	197	155*	361	361	39.9	39.9
7-G-800-3	125	236	125*	439	439	45.3	45.3
7-G-800-4	95	212	95*	430	430	45.0	45.0
7-G-800-5	65	212	70*	446	873	46.0	54.8
26-G-600-1	151	310	162*	494	961	50.5	60.4
26-G-600-2	121	280	196	465	144578	49.3	7200.0
26-G-600-3	91	185	185	0	156045	14.8	7200.0
26-G-600-4	61	155	155*	0	31	14.9	16.6
11-U-1000-1	212	320	212*	540	540	57.0	57.0
11-U-1000-2	182	290	182*	521	521	55.8	55.8
11-U-1000-3	152	260	175	497	146215	53.5	7200.0
11-U-1000-4	122	248	155	543	147400	57.0	7200.0
11-U-1000-5	114	207	155	472	145888	51.7	7200.0
11-U-1400-3	70	124	100*	287	566	33.9	41.5
11-U-1400-4	40	110	100*	278	546	33.3	41.4
11-U-1400-5	40	110	100*	287	564	33.9	42.1
11-U-1800-5	183	343	343	0	187610	16.0	7200.0
11-G-1000-3	211	236	211*	467	467	50.3	50.3
11-G-1000-4	181	215	181*	463	463	49.5	49.5
11-G-1000-5	151	215	151*	476	476	50.4	50.4
16-U-1800-2	237	343	237*	591	591	58.2	58.2
16-U-1800-3	207	262	207*	471	471	50.5	50.5
16-U-1800-4	177	262	177*	496	496	50.5	50.5
16-U-1800-5	177	262	177*	507	507	51.4	51.4

Table 5.4: Hard instances with one and two hour timetables

FCFS, FLFS and the final solution found by BB, respectively. The asterisk on the elements of column four indicates proven optimality of the current CDRFR solution. Finally, columns five and six [seven and eight] give the number of nodes in the enumeration tree [the computation time in seconds] that are needed to reach the best solution and the termination of the BB algorithm, respectively.

The BB algorithm is able to close 11 out of 14 hard instances of the one hour timetable, for which the proven optimal solution is always found within five seconds. The optimal solution is proven within 11 minutes for two of the three remaining hard instances. One open instance remains, for which the BB algorithm is unable to improve the solution found in the first 4.4 seconds. A similar behavior can be observed for the two hour instances. In Table 5.4, six instances remain unsolved within two hours of computation, but the best solution is always found in less than one minute. Moreover, the proven optimum equals the lower bound computed for the initial graph selection in 50% of instances.

5.7.2 Assessment of flexible timetables

In this subsection, we study our CDRFR algorithms when dealing with different timetables. We assess the impact of timetable flexibility by varying the departure time of trains at scheduled stops and the amount of buffer time scheduled in the timetable. We present an extensive computational study on the complex Schiphol dispatching area between Amsterdam and Leiden. We intend to demonstrate that timetable flexibility is a promising concept to reduce consecutive delays. Moreover, we will show that similar levels of train delay performance can be attained by increasing the flexibility of departure times while reducing the buffer time, which corresponds to increase the throughput of the railway network.

We study the network by simulating real-time traffic conditions, and by rescheduling trains for a large set of randomly generated entrance perturbations. We model an initial perturbation by assigning an entrance delay to some trains, i.e., by introducing some late trains in the network. Precisely, we study the first set of 60 perturbation schemes described in Section 5.7.1. For each instance, we compute four CDRFR problem solutions by using the three initial heuristics and the BB algorithm, truncated after 120 seconds of computation.

The experiments of this subsection are organized as follows. Initially, we evaluate, in terms of maximum and average consecutive delays, the solutions produced by CDRFR algorithms for different levels of flexibility. Then, we analyze the effects of four flexible timetables with different amounts of buffer time and throughput.

Analysis of CDRFR algorithms

In the first set of experiments, we consider the typical rolling stock characteristics and a provisional hourly timetable for the year 2007 (Hemelrijk et al., 2003). Starting from this (rigid) timetable, we construct a flexible timetable by replacing the scheduled arrival/departure times with maximum arrival/minimum departure times. To ensure fairness

when comparing the results of flexible timetables, we set the maximum arrival times constant and equal to the rigid arrival times in all experiments. In fact, increasing these values would immediately result in reduced train delays without changing their actual arrival times.

We evaluate the effectiveness of four CDRFR algorithms by varying the level of flexibility. Given a value of the flexibility δ , we set the minimum departure times equal to the rigid departure times minus δ . Five timetables are generated for $\delta = \{0, 30, 60, 90, 120\}$, where $\delta = 0$ corresponds to the rigid timetable. For each of the 60 perturbation schemes, we generate an instance of the CDRFR problem for $\delta = \{0, 30, 60, 90, 120\}$, thus yielding a total of 300 CDRFR instances.

Figure 5.8 shows the maximum and average consecutive delays obtained by the four algorithms for different values of the flexibility δ . Each value reported is the average over the 60 perturbation schemes. For some instances, one or more of the three initial heuristics fail to find a feasible solution. In order to compare the different results, we consider a failure penalty of 10 minutes for the maximum consecutive delay and 1 minute for the average consecutive delay. Precisely, we assume that finding no solution should be considered worse than finding a bad quality solution, and we thus impose a penalty double than the worst case for each failure.

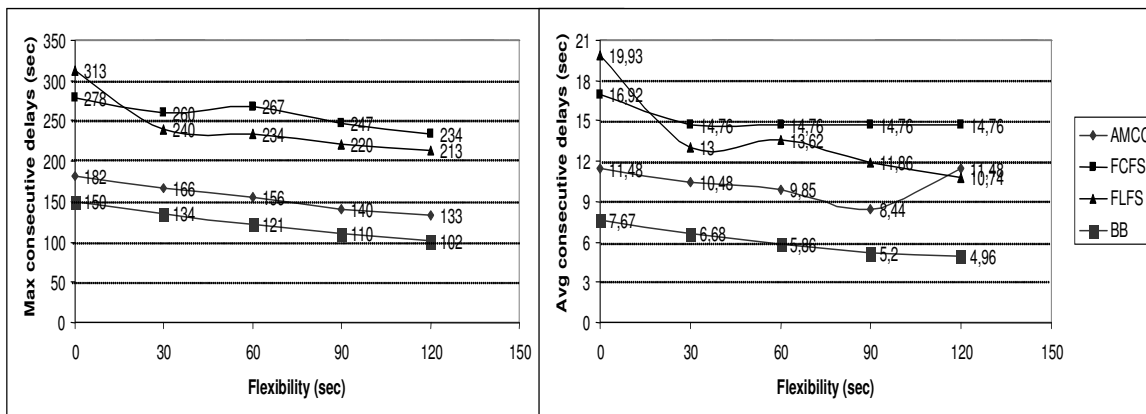


Figure 5.8: Maximum and average consecutive delays for the four algorithms

As expected, all the four algorithms take advantage of an increasing value of timetable flexibility, even if the three initial heuristics exhibit less consistent behaviors. The BB algorithm clearly outperforms the other algorithms in terms of delay minimization (see Figure 5.7). The maximum and average consecutive delays of the dispatching heuristics are more than double with respect to the values found by the BB algorithm.

The best BB solution is always found within the first 30 seconds of computation. However, the computation time of the three heuristics is significantly shorter than for the BB algorithm. FCFS only needs 0.01 seconds on average to compute a solution, FLFS requires 0.02 seconds and AMCC 0.5 seconds. These heuristics are therefore potentially useful even if very strict time limits are imposed to compute a CDRFR solution.

Figure 5.9 shows the maximum and average consecutive delays obtained by the BB algorithm at the stops in Hoofddorp and Schiphol platforms as well at the exit points of the

dispatching area. Clearly, Hoofddorp is a less critical station than Schiphol. Besides, some trains experience an additional consecutive delay from Schiphol to the exit points of the dispatching area.

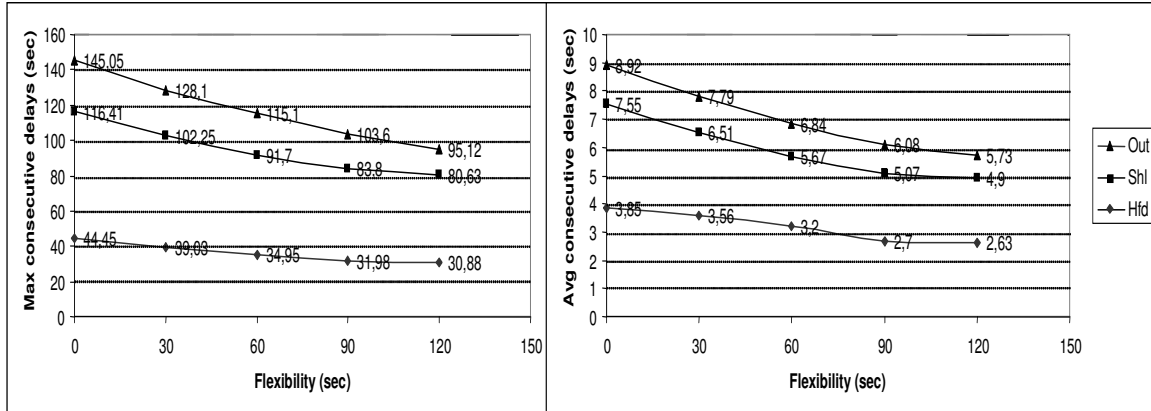


Figure 5.9: Maximum and average consecutive delays at the stations and exit points

Increasing the flexible departure times from 0 to 120 seconds causes a reduction of both the maximum and average consecutive delays at all stations/exit points of more than 30%. However, the effectiveness of the amount of flexibility is not linear. When increasing the flexibility from 0 to 30 seconds the maximum and average consecutive delays decrease of around 10%, while an increment from 90 to 120 seconds causes a reduction of around 5% for both values.

This set of experiments shows that the use of advanced optimization algorithms is important in order to obtain significant benefits in terms of maximum and average consecutive delays. In fact, all algorithms are able to exploit the benefits of flexibility in order to reduce train delays while the BB algorithm makes the best use of flexibility compared to the proposed heuristics.

Analysis of different timetables

In the second set of experiments, we compare the solutions obtained for different levels of flexibility and buffer time inserted in the timetable. Specifically, we study the effects of flexibility when varying the level of buffer time. We use the first set of 60 perturbations of Section 5.7.1, and discuss only the CDRFR solutions provided by the BB algorithm.

We consider 4 timetables (A, B, C and D) and refer to the original one as timetable B. Removing all the buffer time from timetable B gives a less robust timetable A in which the 27 trains per direction are compressed into a period of approximately 3200 seconds. This corresponds to a maximal theoretical throughput of 31 trains per hour per direction that can run in the network within an hour, if all trains would run at their scheduled speed. In fact, timetable A still contains a limited amount of time reserves. The largest recovery time is 240 seconds for a train which requires a minimum of 790 seconds to traverse the network

plus 240 seconds of cumulated minimum dwell time at the two intermediate stations. We also developed two new timetables, C and D, obtained from A by doubling and tripling the amount of buffer time of timetable B. This corresponds to a throughput of 25 and 23 trains per hour per direction in timetable C and D, respectively. In the four timetables, the speed profile (and the recovery time) of each train is kept as close as possible to the original timetable, so that the amount of initial delay associated to a certain entrance delay is approximately the same when varying the timetable and the level of flexibility. On average over the 60 instances, the maximum initial delay is 544, 534, 533, and 530 seconds for timetable A, B, C and D, respectively. The average initial delay is approximately 67 seconds for timetables A, B and C, and 66 seconds for timetable D.

To assess the timetable robustness on each perturbation scheme, we computed a solution by using the BB algorithm for varying flexible departure times from 0 to 180 seconds. The average results over the 60 perturbation schemes are shown in Figure 5.10, where the maximum and average total delay are reported for each timetable (A, B, C, D) and each value of flexibility ($\delta \in \{0, 30, 60, 90, 120, 150, 180\}$). Since the four timetables differ in the time separation between consecutive trains, we only compare these timetables in terms of total delays (i.e., initial delays are slightly different). As expected, the maximum and average total delays are significantly smaller in the case with larger buffer times. However, increasing the flexibility from 0 to 30 seconds is always more effective than from 150 to 180 seconds.

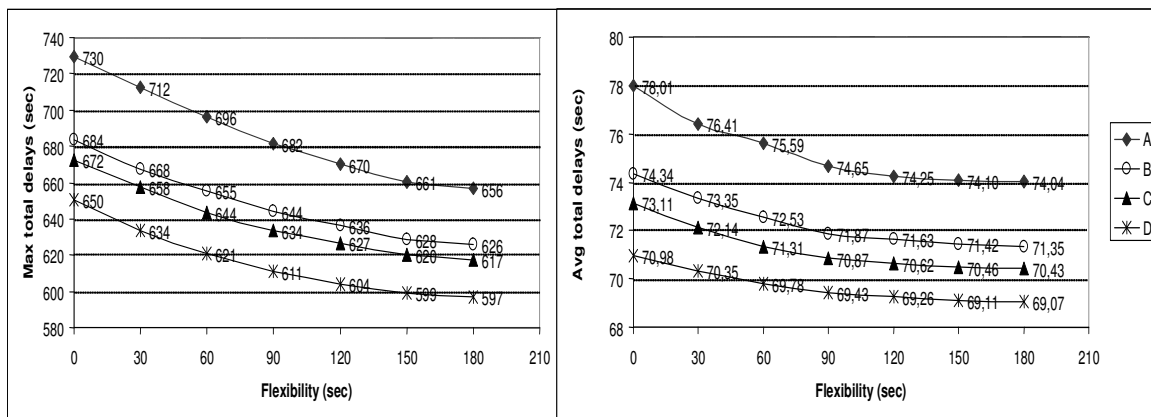


Figure 5.10: Maximum and average total delays for the four timetables

From Figure 5.10, it is also interesting to compare the effects of different levels of buffer time and flexibility against the total (maximum or average) delay and the throughput. In fact, timetables with different levels of buffer time and flexibility exhibit similar maximum or average total delays (see e.g. timetable A with $\delta = 90$ versus timetable B with $\delta = 0$, or timetable B with $\delta = 120$ versus timetable D with $\delta = 30$). Since varying the buffer time corresponds to a different throughput, we can evaluate the flexibility in terms of equivalent buffer time and increased line capacity. Similar values of maximum total delay are obtained by adding four trains per hour and increasing the flexibility to 90 seconds, while similar values of average total delay are obtained through adding two trains per hour and increasing the flexibility to 90 seconds.

No general conclusion can be drawn from this limited set of experiments, but the results show, clearly, that timetable flexibility combined with advanced CDRFR algorithms are promising tools that deserve further investigation in order to increase the robustness of railway lines while improving the punctuality of running trains.

5.8 Conclusions

This chapter describes different CDRFR algorithms for the real-time management of complex railway networks. We make use of the alternative graph formulation of the CDRFR problem (see Chapter 4) and deal with algorithms for global conflict resolution. Given an initial perturbation that requires timetable modifications, the objective is to minimize the maximum consecutive delay for all trains at all relevant points. The main achievement is the development and implementation of a branch and bound algorithm that is able to solve the problem within a short computation time.

We carried out computational experiments on the Dutch bottleneck of Schiphol, a congested area of the Dutch railway network, and considering multiple delayed trains. The results show a very promising performance of the BB algorithm in finding near-optimal solutions to practical-size instances of the CDRFR problem, truncating its execution after 120 seconds of computation. More importantly, from a practical point of view, the computational experience demonstrates that algorithms based on the alternative graph model outperform the proposed dispatching rules in terms of both maximum and average delays. Hence, the algorithms based on the alternative graph model may yield significant improvements in the quality of the railway service within a short computation time. We finally mention that the effectiveness of the remaining dispatching heuristics (described in this chapter) will be assessed and compared with the BB algorithm in Chapters 6 and 7.

In this chapter, we also study the concept of timetables with flexible arrival and departure times when dealing with the railway traffic management on a busy railway network. Specifically, three greedy heuristics and the BB algorithm are tested on the Schiphol bottleneck area and several timetable perturbations. From these computational experiments, several conclusions can be drawn:

1. Flexible timetables are preferable to rigid ones, since flexibility offers more freedom to solve conflicts and may reduce consecutive delays without decreasing the throughput (provided that the timetables contain sufficient time reserves).
2. When comparing different values of buffer time and flexibility with similar behavior in terms of maximum or average delay, flexible timetables enable a higher throughput, thus making a better use of existing capacity compared to rigid timetables.
3. The use of advanced CDRFR optimization algorithms improves the benefits of flexible timetables in terms of delay minimization.

However, the application of flexible arrival and departure times (and other RDTM principles such as flexible train routing in stations) in practice may be limited, too, by the ability of the passengers to understand and accept such a standard of changeability of train services.

Future research should be addressed to the process of designing flexible timetables: What is the best amount of flexibility required in practice? How to distribute time margins in a flexible timetable? Which is the best trade-off between the amount of buffer time, recovery time and flexibility? Where should buffer time be located? New concepts are necessary to answer such questions, and deeper analysis of these points will be useful from theoretical as well as from practical points of view.

Chapter 6

Train rerouting algorithms

This chapter describes the rerouting algorithms developed for this thesis and implemented in the ROMA dispatching support system. We thus address the conflict detection and resolution (CDR) problem, which is practiced every day by traffic controllers to adapt the timetable to delays and other unpredictable events occurring during operations.

We investigate mathematical properties of CDR solutions and evaluate the benefits of local routing modifications in order to improve train punctuality. We also take advantage from reliable rescheduling algorithms (see Chapter 5) and incorporate them in the ROMA traffic optimization procedure. Here we address the minimization of the maximum and average consecutive delays in lexicographic order. To be precise, we denote the *lexicographic comparison* as $[a; b] < [c; d]$ if $a < c$ or if $a = c$ and $b < d$.

Firstly, the branch and bound algorithm of Chapter 5 is embedded in a local search framework in which train routes can be changed when better solutions are achievable. Precisely, a move is to change one route and its evaluation is to solve the associated CDRFR problem. A train route is modified if the new solution improves the above-mentioned objective functions. The search for promising routes is based on the CDR properties.

Successively, we explore the effectiveness of extensive rerouting strategies by incorporating the search for new routes in a tabu search scheme. This allows to escape from local minima. We also investigate the effects of using various neighborhood structures for the CDR problem and report on several algorithmic improvements that allow to speed up the algorithm considerably. Specifically, our tabu search is based on a fast heuristic evaluation of a move and on a hybrid neighborhood scheme. We alternate the search for promising moves in neighborhoods of different size, similarly to the *variable neighborhood tabu search* developed by Moreno Pérez et al. (2003) who tested it on the median cycle problem.

We study the performance of the proposed algorithms when dealing with practical size instances of the Dutch dispatching area between Utrecht and Den Bosch (see Appendix A). We study complex timetable perturbations (i.e., several trains delayed at their entrance or within the studied network) and disruptions (i.e., some tracks are temporarily unavailable). We also analyze the relevant case in which a double track corridor is blocked in one of the

two directions and the trains traveling in opposite directions are forced to use the same track in both traffic directions. We observe that the presence of blocked zones and strong perturbations may require extensive real-time timetable modifications to reach feasible CDR solutions in a regional railway network of practical size. We compare tabu search CDR solutions with optimal CDRFR solutions, provided by the branch and bound algorithm, and local search CDR solutions.

This chapter is organized as follows. Section 6.1 provides the CDR properties while in Section 6.2 we insert them in a local search scheme. Section 6.3 deals with the tabu search algorithm development. Section 6.4 reports on the computational results for different level of disturbances and various configurations of the proposed algorithms. Section 6.5 presents some conclusions and subjects for further research.

6.1 Properties of the CDR problem

This section describes some properties of the CDR problem that can be used to direct the search for optimal train routes. Note that each train is associated with a passable route in the alternative graph $\mathcal{G} = (N, F, A)$ and a set of alternative passable routes are generated (see Chapter 3), and modeled in terms of the alternative graph formulation (see Chapter 4), before the resolution of the CDR problem starts.

Let $S(F)$ be the CDRFR solution computed by a train scheduling algorithm (for fixed F). We start from $\mathcal{G}(F, S)$ and search for a new set F' , i.e., we compute a new alternative graph $\mathcal{G}' = (N, F', A')$ and a new solution $\mathcal{G}'(F', S')$. We use the notation $l^{F', S(F)}(0, n)$ to denote the longest path in $\mathcal{G}'(F', S')$ such that all alternative pairs in $A \cap A'$ are selected as in $S(F)$. With this notation, F' is preferred to F if $l^{F', S(F)}(0, n) < l^{F, S(F)}(0, n)$.

Given a set F and two consistent selections $S(F)$ and $S(F')$, let $P^C \subseteq S(F)$ be the set of arcs of $S(F)$ belonging to $\mathcal{C}(F, S)$. A well known property of the job shop scheduling problem (Balas, 1969) states that if $P^C \subseteq S(F')$ then $l^{F, S(F')} (0, n) \geq l^{F, S(F)}(0, n)$. We next restate this property for the problem in which set $S(F)$ is fixed and set F can be modified. Given a *rerouting option* ω , i.e., a chain of fixed arcs of F , we denote with $l(\omega)$ its length.

Property 6.1.1 *Consider a set F and the corresponding critical path set $\mathcal{C}(F, S)$. Let F' be a new arc set obtained from F by replacing a rerouting option $\omega_1 \subseteq F$ for some train with a different rerouting option ω_2 with the same terminal nodes as ω_1 . If $\omega_1 \subseteq \mathcal{C}(F, S)$ and $l(\omega_1) \leq l(\omega_2)$, then $l^{F', S(F)}(0, n) \geq l^{F, S(F)}(0, n)$.*

Proof. Simply observe that the new solution $\mathcal{G}'(F', S')$ contains the path $\mathcal{C}'(F', S')$ obtained by $\mathcal{C}(F, S)$ substituting rerouting option ω_1 with rerouting option ω_2 . Denoting with $l(\mathcal{C}'(F', S'))$ the length of $\mathcal{C}'(F', S')$, and since the length of ω_1 is smaller or equal to the length of ω_2 , we have $l^{F', S(F)}(0, n) \geq l(\mathcal{C}'(F', S')) \geq l^{F, S(F)}(0, n)$. \square

When a new set F' is built, the solution can be further improved by rescheduling train operations for fixed routings. However, changing a route which is not contained in the critical path and then computing an optimal schedule may also lead to better solutions. The proposed properties focus on a set of promising route modifications, strictly containing the changes on the critical path.

To this aim, let us introduce the following notation. Given a node $i \in \mathcal{C}(F, S)$, we call *ramification* $R(i)$ all the sequences of fixed arcs from node 0 to any node $j \in N(F)$ such that there is a path from node j to node i in $\mathcal{G}(F, S)$. We call *ramified critical path set* the set $\mathcal{R}(F, S) = \bigcup_{i \in \mathcal{C}(F, S)} [R(i)] \cup \mathcal{C}(F, S)$, i.e., the critical path plus the sets $R(i)$ computed for all the nodes of the critical path.

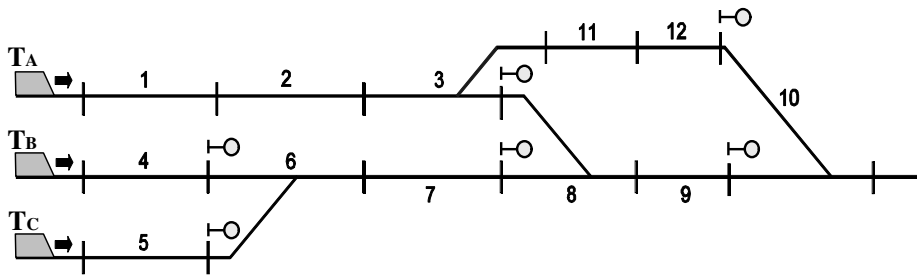


Figure 6.1: A small example with two paths for train T_A

Figure 6.1 shows a small railway network with three trains, denoted as T_A , T_B and T_C . In this example, we only report the location of the most relevant block signals. We assume that T_A can pass through block sections 1, 2, 3, 8, 9, 10 or through block sections 1, 2, 3, 11, 12, 10. Two alternative graphs are therefore associated with the different routes of T_A , which implies different sets of fixed and alternative arcs (see Figure 6.2 and Figure 6.3). In each graph, the set of arcs belonging to the critical path and the ramified critical path are depicted with bold black and bold grey arrows, respectively. The selected alternative arcs are depicted with dotted arrows.

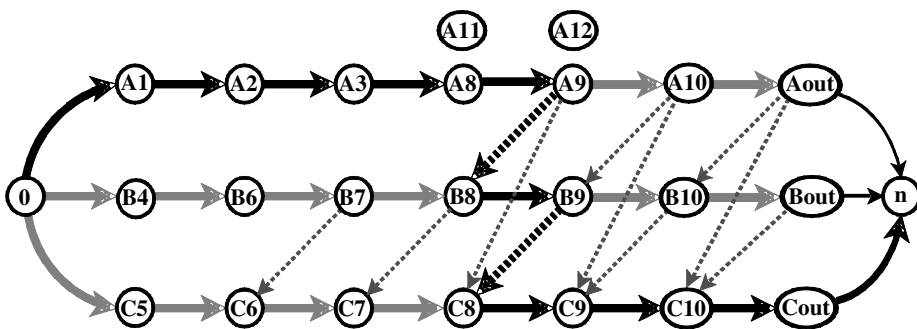


Figure 6.2: Alternative graph with default route for train T_A

In the graph of Figure 6.2, the trains are sequenced in the order: $T_A-T_B-T_C$. The graph of Figure 6.3 is obtained by changing the route of T_A . In the latter graph, the new train order is $T_B-T_C-T_A$ while T_A has no longer the conflicts with T_B and T_C on block sections 8 and 9. Consequently, the two graphs have different ramified critical paths. The whole route of

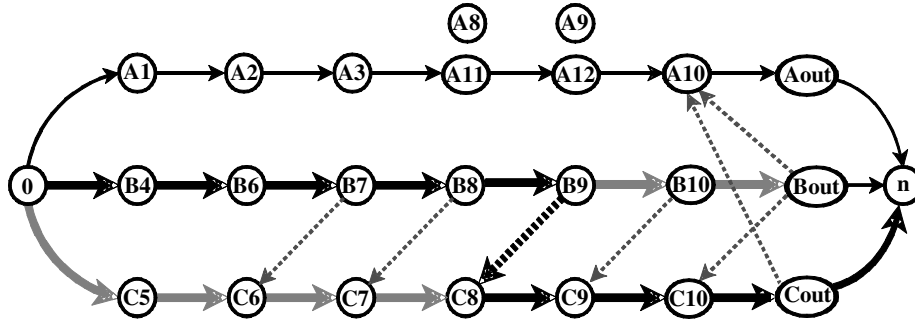


Figure 6.3: Alternative graph with alternative route for train T_A

T_A no longer belongs to the ramified critical path. For this reason, it may be possible that a longer route for T_A yields a smaller value for the maximum consecutive delay.

The following property motivates the search for better routes among the routes of the ramified critical path. We denote by P^R the set of arcs of F belonging to the ramified critical path set $\mathcal{R}(F, S)$.

Property 6.1.2 Given two sets F and F' , if $P^R \subseteq F'$, then $l^{F', S(F)}(0, n) \geq l^{F, S(F)}(0, n)$.

Proof. Simply observe that the new solution $\mathcal{G}'(F', S')$ contains the set $\mathcal{C}(F, S)$. \square

6.2 A local search algorithm

Since the compound CDR problem for a railway network is not tractable by exact methods, we exploit the CDR properties of the previous section to design a novel local search algorithm. The pseudo-code of this algorithm is given in Figure 6.4.

In general, a local search technique starts from an incumbent solution (*IncSol*) and then iteratively moves to a neighbor solution. This is typically an *incomplete* algorithm, as the search may stop even if the best solution found (*BestSol*) by the algorithm is not optimal. This can happen even if termination is due to the impossibility of improving the current local best solution (*LocalBestSol*), as the optimal solution (*OptSol*) can lie far from the neighborhood of the solutions crossed by the algorithm.

The neighborhood of a solution $\mathcal{G}(F, S)$ contains all the solutions $\mathcal{G}'(F', S')$ in which F' is obtained from F by changing a single route on the ramified critical path. If the route is contained in the critical path, this can be replaced only with a shorter route, as stated by Property 6.1.1. Note that computing $\mathcal{G}'(F', S')$ requires the execution of a CDRFR algorithm, which can be computationally expensive. We therefore limit the search for better solutions to a subset of promising neighbors only, as follows.

Given a rerouting option ω , let u and v be its first and last nodes, and $l(\omega)$ be the length of route ω in $\mathcal{G}(F, S)$. We estimate the potential of a new local rerouting option ω as the

Algorithm LocalSearch**Input:** An *IncSol* $\mathcal{G}(F, S)$,**begin****While** (Rerouting options available) & (time limit not reached) & (max consecutive delay > 0) **do** **begin** Build neighborhood for $\mathcal{G}(F, S)$ (using Properties 6.1.1 and 6.1.2), Insert all ω_i in L_1 , $LocalBestSol = +\infty$, **While** $L_1 \neq \emptyset$ **do** **begin** Add to L_2 the ψ routing options with the highest potential $\Pi(\omega_i)$, $L_1 = L_1 \setminus L_2$, **For all** rerouting options in L_2 **do** **begin** Create the alternative graph $\mathcal{G}' = (N, F', A')$, Reschedule trains for graph \mathcal{G} according to a chosen CDRFR algorithm, **If** $LocalBestSol >$ current rerouting option in L_2 **then** **begin** $LocalBestSol =$ current rerouting option in L_2 , **End** **End** **If** $LocalBestSol <$ $IncSol$ **then** **begin** $IncSol = LocalBestSol$, $L_1 = \emptyset$, **End** **Else** $L_2 = \emptyset$, **End** **End****Return** $BestSol = IncSol$.**End****Figure 6.4:** Sketch of the local search algorithm

quantity $\Pi(\omega) = l^{F,S(F)}(0, v) - l^{F,S(F)}(0, u) - l(\omega)$, since the length $l^{F,S(F)}(0, v)$ might decrease to $l^{F,S(F)}(0, u) + l(\omega)$ when changing route. We restrict the neighborhood to the ψ routes with highest potential, where ψ is a parameter of our local search procedure. If none of the ψ routes improves the solution, we evaluate the next ψ routes with highest potential. The procedure continues as long as an improvement is possible or no rerouting option is available or the time limit is reached.

For each of the ψ best routes we define a neighbor F' by replacing an old route of F from node u to node v with a new route, including all the fixed arcs required to implement the new route and its railway constraints, and then execute the train rescheduling procedure. Among the ψ new solutions $\mathcal{G}'(F', S')$, we choose the one having the shortest critical path, i.e., the one minimizing the maximum consecutive delay. In case of tie, we choose the solution with minimum average consecutive delay.

6.3 A tabu search algorithm

This section deals with the development of more sophisticated algorithms for solving the CDR problem. We propose a tabu search approach to this problem, since its combinatorial structure is similar to that of the job shop scheduling problem with routing flexibility and tabu search algorithms achieved very good results with the latter problem (see, e.g. Mastrolilli and Gambardella (2000)).

The tabu search (TS) is a deterministic metaheuristic based on local search (see, e.g. Glover (1986) and Glover and Laguna (1997)), which makes extensive use of memory for guiding the search. Basic ingredients of a tabu search are the concepts of move and tabu list, which restrict the set of solutions to explore. From the incumbent solution, non-tabu moves define a set of solutions, called the incumbent solution neighborhood. At each step, the best solution in this set is chosen as the new incumbent solution. Then, some attributes of the former incumbent are stored in a tabu list, used by the algorithm to avoid being trapped in local optima and to avoid revisiting the same solution. The moves in the tabu list are forbidden as long as these are in the list, unless an aspiration criterion is satisfied. The tabu list length can remain constant or be dynamically modified during the search.

We notice that, despite their similarity, there are significant differences between the CDR problem and the job shop scheduling problem, such as the absence of inter-machine buffers in the CDR problem, called no-store or blocking constraint (see, e.g. Hall and Sriskandarajah (1996) and Grabowski and Pempera (2000)). As a result, most of the properties that are used in the job shop scheduling problem to design effective neighborhood structures do not hold for the CDR problem. We observe that computing reliable estimates of the global impact of a local change, either train rerouting or reordering, may require a significant amount of time. Even the feasibility of a solution after a local change cannot be ensured as it occurs, e.g. in the job shop scheduling problem this happens when reordering two consecutive operations laying on the critical path (Balas, 1969). For these reasons, our

tabu search adopts a different searching scheme with respect to those mostly used for the flexible job shop problem (see, e.g. Mastrolilli and Gambardella (2000)).

In our tabu search algorithm, the basic move consists of locally changing the route of a single train in order to avoid passing on a specific block section. Given a route-set F , we evaluate the quality of a solution by computing a new solution $S(F)$ to the CDRFR problem. If no feasible solution $S(F)$ can be computed for a route-set F then the move is not allowed, which occurs e.g. when changing a train route leads to a deadlock situation. Specifically, we analyze three methods for computing $S(F)$ as described in Section 6.3.1. The first method produces a near-optimal solution to the CDRFR problem by the branch and bound algorithm. The other two methods aim at reducing the computation time and produce upper and lower bounds to the optimum of the CDRFR problem. Since in all the three cases the evaluation of a single move is still computationally expensive, we restrict the number of solutions that have to be evaluated by reducing the neighborhood size. In this section, we avoid the use of aspiration criteria, which in our computational experience do not improve the solution quality (as reported in Section 6.4).

The remaining part of this section is organized as follows. Section 6.3.1 briefly describes the different scheduling algorithms adopted to compute upper and lower bounds to the CDRFR problem, i.e., when the route-set F is given. Section 6.3.2 introduces three neighborhood structures of different size while Section 6.3.3 gives the tabu search scheme.

6.3.1 Upper and lower bounds to the CDRFR problem

We adopt a lower bound (LB), as described in Chapter 5, that is based on a specialized version of JPS to deal with the CDRFR problem. As for the upper bound, we use a branch and bound algorithm (BB) to compute near-optimal CDRFR solutions (see Chapter 5) and a faster algorithm that takes into account the modification of train routes as follows. We use a heuristic strategy for computing $S(F')$ for a given incumbent solution $(F, S(F))$ and a new route-set F' , differing from F for the route of a single train T_y . We first exclude from $\mathcal{G}(F, S)$ all the nodes associated to the old route of T_y , all the fixed arcs in F and all the alternative arcs in $S(F)$ incident in a node $i \in N(F)$, thus obtaining a reduced solution $(F'', S(F''))$. We then add to $(F'', S(F''))$ the fixed arcs associated to the new route of T_y , thus obtaining the route-set F' , and the associated pairs of alternative arcs $A(T_y)$. The resulting alternative graph can be indicated as $(F', S(F''), A(T_y))$. A new solution is then computed on this graph by using the greedy algorithm AMCC which selects one alternative arc at a time (see Chapter 5). The idea is to forbid at each iteration the alternative arc which would introduce the largest consecutive delay in the current selection, thus selecting its paired alternative arc. In this chapter, we refer to the latter algorithm as UB.

6.3.2 Routing neighborhoods

We propose three neighborhood structures of different size and combine two of them to form hybrid neighborhood structures. To this aim, we need to introduce further notations.

Given a solution $S(F)$ and a node $i \in N(F) \setminus \{0, n\}$, we say that i is a *critical node* in $S(F)$ if $l^{F,S(F)}(0, i) + l^{F,S(F)}(i, n) = l^{F,S(F)}(0, n)$. A critical node i is a *waiting node* if $l^{F,S(F)}(0, i) > l^{F,S(F)}(0, \mu(i)) + b_{\mu(i)i}$, where $\mu(i)$ is the node which precedes node i on its route. For each waiting node i , there is always a node g_i in $\mathcal{G}(F, S)$, different from node $\mu(i)$, such that $l^{F,S(F)}(0, i) = l^{F,S(F)}(0, g_i) + b_{g_i i}$ and we call g_i the *hindering node* of o_i . Notice that for each waiting node $i \in N(F) \setminus \{0, n\}$ there is exactly one hindering node.

Given a node $i \in N(F) \setminus \{0, n\}$, we recursively define its *backward ramification* $R_B(i)$ as follows. If node i is waiting, then $R_B(i) = R_B(\mu(i)) \cup R_B(g_i) \cup \{i\}$, otherwise $R_B(i) = R_B(\mu(i)) \cup \{i\}$. Similarly, we recursively define the *forward ramification* $R_F(i)$ as follows. If node i is the hindering of a waiting node k (i.e., if $o_i = o_{g_k}$), then $R_F(i) = R_F(\sigma(i)) \cup R_F(k) \cup \{i\}$, otherwise $R_F(i) = R_F(\sigma(i)) \cup \{i\}$. Moreover, by definition, $R_B(0) = R_F(0) = \{0\}$ and $R_B(n) = R_F(n) = \{n\}$.

Given the critical path set $\mathcal{C}(F, S)$, we call *backward ramified critical path set* (BRCP) the set $\mathcal{B}(F, S) = \bigcup_{i \in \mathcal{C}(F, S)} [R_B(i)]$ and *forward backward ramified critical path set* (FBRCP) the set $\mathcal{F}(F, S) = \bigcup_{i \in \mathcal{C}(F, S)} [R_B(i) \cup R_F(i)]$. Observe that Properties 6.1.1 and 6.1.2 hold for the set of arcs belonging to both $\mathcal{B}(F, S)$ and $\mathcal{F}(F, S)$.

For example, in the graph of Figure 4.11:

$$\mathcal{C}(F, S) = \{0, B7, B8, B9, B10, B5, B6, Bout, C6, Cout, n\};$$

$$\mathcal{B}(F, S) = \{0, B7, B8, B9, B10, B5, B6, Bout, C11, C8, C9, C10, C5, C6, Cout, n\};$$

$$\mathcal{F}(F, S) = \{0, B7, B8, B9, B10, B5, B6, Bout, C11, C8, C9, C10, C5, C6, Cout, n\}.$$

In the graph of Figure 4.12:

$$\mathcal{C}(F, S) = \{0, C11, C8, C9, B8, B9, B10, B5, B6, Bout, n\},$$

$$\mathcal{B}(F, S) = \{0, B7, B8, B9, B10, B5, B6, Bout, C11, C8, C9, n\};$$

$$\mathcal{F}(F, S) = \{0, B7, B8, B9, B10, B5, B6, Bout, C11, C8, C9, C10, C5, C6, Cout, n\}.$$

In this section, we study the three neighborhood structures listed below. The two latter neighborhoods can be viewed as restricted versions of the first one.

\mathcal{N}_C . The *complete neighborhood* contains all the feasible solutions to the CDR problem in which one train follows a different route compared to the incumbent solution. This is the largest neighborhood we consider. To limit the number of neighbors to be evaluated, \mathcal{N}_C is only partially explored as follows. A move is obtained by choosing a non-tabu train and a route different from the current one at random (i.e., all trains and alternative routes having the same probability), until a number ψ of alternative route-sets is obtained, where ψ is a parameter of the tabu search algorithm.

\mathcal{N}_{BRCP} . The *backward ramified critical path neighborhood* considers only the operations in $\mathcal{B}(F, S)$ and the associated trains. The idea is that the maximum consecutive delay of an optimal solution to the CDRFR problem can be reduced by removing one of the conflicts causing it. This requires either removing an operation from the critical path set (i.e., rerouting the associated train through a different block section) or anticipating its arrival at the conflict point. The latter result can be obtained by removing an operation from $\mathcal{B}(F, S)$ and then rescheduling train movements. This neighborhood structure has been studied in the previous section within a local search scheme.

\mathcal{N}_{FBRCP} . The *forward backward ramified critical path neighborhood* extends the previous neighborhood by considering also the forward ramifications, i.e., all the operations in $\mathcal{F}(F, S)$ and the associated trains. This neighborhood contains all the solutions to the CDR problem, in which one train associated to $\mathcal{F}(F, S)$ is rerouted. \mathcal{N}_{FBRCP} is explored by alternating the rerouting of a train in the forward ramification to the rerouting of a train in the backward ramification.

All the proposed neighborhoods enlarge the common neighborhood structures for the job shop problem, in order to avoid empty neighborhoods as far as possible. However, the adopted search strategy requires defining the sets F and $S(F)$ in succession. It is thus worthwhile studying specific properties of the different neighborhood structures.

A neighborhood structure \mathcal{N} is called *opt-connected* if, starting from any feasible solution $(F, S(F))$ to the CDR problem, an optimal solution $(F^*, S^*(F^*))$ can be reached after a finite number of moves chosen in \mathcal{N} . In our case, \mathcal{N} is explored by first defining a new route-set F and then computing $S(F)$. We must therefore take into account this assumption when studying the different neighborhood structures.

It can be proved that \mathcal{N}_C is opt-connected if for any route-set F' there exists a feasible schedule $S(F')$, which is always the case when the network is deadlock-free. If $F \neq F^*$ then at least one train can be rerouted according to its route in F^* , thus leading from F to F^* after a number of moves smaller or equal to the number of trains. In order to obtain $(F^*, S^*(F^*))$, it is sufficient to choose the selection $S^*(F^*)$ after the route-set F^* is reached. However, if deadlock situations may arise, opt-connectedness is not guaranteed.

An example of this situation is shown in Figure 6.5, in which two trains, T_A and T_B , run in opposite directions on a single track line. Suppose that the optimal CDR solution requires T_A passing through block sections 1, 2 and 4, and T_B passing through block sections 4, 3 and 1. Consider now a feasible solution in which T_A and T_B are routed through block sections 3 and 2, respectively. Clearly, changing only one route for T_A or T_B leads to a deadlock situation for which no feasible schedule exists. Thus, it is not possible to reach the optimal solution passing through a sequence of feasible intermediate solutions each differing from the previous one for at most one train route.

We now show that even if for any route-set F' there exists a feasible schedule $S(F')$, the two restricted neighborhoods \mathcal{N}_{BRCP} and \mathcal{N}_{FBRCP} are not opt-connected. To show this fact, consider the example described in Section 4.3.1. Analyzing the network in Figure 4.9,

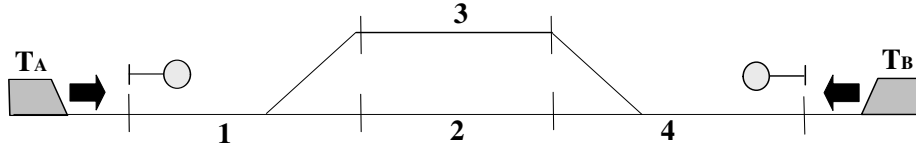


Figure 6.5: A small example in which \mathcal{N}_C is not opt-connected

it is straightforward to state that only one route exists for T_B and T_C , while three routes are possible for T_A (through block section 4, 10 or 12), all leading to feasible schedules. The solution in Figure 4.11 can be improved only by changing the route of T_A but no node of this train belongs to $\mathcal{F}(F, S)$, thus implying that the two restricted neighborhoods are empty. In other words, starting from the solution in Figure 4.11, no move is allowed in \mathcal{N}_{BRCP} or \mathcal{N}_{FBRCP} , and the optimal solution in Figure 4.12 cannot be reached.

The above discussion suggests new research directions to design effective CDR algorithms. In what follows, we assume that for any route-set F' there exists a feasible schedule $S(F')$ and the length of the tabu list is smaller than the number of trains that can be rerouted. The former assumptions imply that there is always a non-tabu move in \mathcal{N}_C .

For each route-set F , an optimal or near-optimal schedule $S(F)$ has therefore to be computed in order to avoid missing the optimal CDR solution, i.e., choosing the route-set F^* . For this reason, independently from the neighborhood structure and move evaluation strategy adopted, whenever a route-set F is changed the algorithm produces a near-optimal solution to the CDRFR problem by using the truncated branch and bound method. The computational experiments of Section 6.4 will show that this time limit is rarely reached. This is therefore the solution we start from in the next iteration.

In order to achieve the opt-connected property, we implement the **Complete** neighborhood search strategy, that explores the complete neighborhood \mathcal{N}_C , and the four restricted neighborhood search strategies described below.

Restart. The best candidate solution is searched in \mathcal{N}_{FBRCP} unless this is empty. Otherwise, $\gamma \geq 1$ consecutive moves are performed in \mathcal{N}_C before searching again in the restricted neighborhood, where γ is a parameter of the tabu search algorithm. A new schedule is computed with the BB algorithm after all γ routes are changed.

Hybrid1. When \mathcal{N}_{FBRCP} is empty, $\gamma > 1$ moves are performed in \mathcal{N}_C before searching again in \mathcal{N}_{FBRCP} . The move is chosen by evaluating ψ candidate solutions in \mathcal{N}_C and selecting the best one with the two objective functions in lexicographic order (i.e., the minimization of the [maximum; average] consecutive delays). Each solution is obtained by first rerouting a single, randomly chosen, non-tabu train and then computing a near-optimal schedule with the BB algorithm.

Hybrid2. When \mathcal{N}_{FBRCP} is empty, $\gamma > 1$ moves are performed in \mathcal{N}_C . Each move is carried out by evaluating ψ candidate solutions in \mathcal{N}_C and selecting the one with

the smallest average delay. Each solution is obtained by first rerouting a single, randomly chosen, non-tabu train and then computing a near-optimal schedule with the BB algorithm.

Hybrid3. When a local minima is reached in the restricted neighborhood, i.e., all moves in \mathcal{N}_{FBRCP} are non-improving, ψ candidate solutions are evaluated in \mathcal{N}_C and the best solution found among those in $\mathcal{N}_{FBRCP} \cup \mathcal{N}_C$ is implemented. The two objective functions are considered in lexicographic order. Each solution in $\mathcal{N}_{FBRCP} \cup \mathcal{N}_C$ is obtained by first rerouting a single, randomly chosen, non-tabu train and then computing a near-optimal schedule with the BB algorithm.

The four restricted strategies are combinations of \mathcal{N}_C and \mathcal{N}_{FBRCP} . Moreover, strategy Restart is a simple restart strategy, i.e., the γ moves are chosen at random without evaluating their quality. This is a commonly used diversification strategy in tabu search algorithms. Among the hybrid strategies, Hybrid1 and Hybrid3 minimize the two objective functions in lexicographic order. The choice of a different objective function for strategy Hybrid2 is motivated by two observations. First, changing the objective function improves diversification, which is often the aim of the restart actions in tabu search algorithms. Second, this choice allows to address directly the secondary objective function, which is useful when the primary objective function cannot be improved (this is often the case when the restricted neighborhood is empty). The tabu search algorithm with strategy Hybrid3 can also be viewed as a variable neighborhood tabu search technique (Moreno Pérez et al., 2003).

6.3.3 The tabu search scheme

Our tabu search algorithm is based on hybrid neighborhood strategies, which explore two neighborhoods of different size (see Section 6.3.2). Figure 6.6 describes the pseudo-code of strategy Hybrid2 while similar schemes apply to the other search strategies.

We indicate with $\mathcal{N}_{FBRCP}(IncSol, \psi)$ the neighborhood of the incumbent solution $IncSol = (F, S(F))$ containing at most ψ elements (ψ rerouting options). The sum of the consecutive delays of all the running trains at their relevant points is $\sum \max\{0, l^{F, S(F)}(0, n)\}$, i.e., the sum of the values $l^{F, S(F)}(0, i) + b_{in}$ for all arcs entering node n . Each solution is evaluated in $\mathcal{N}_{FBRCP}(IncSol, \psi)$ by considering the maximum and average consecutive delays, in lexicographic order, while each solution is evaluated in $\mathcal{N}_C(IncSol, \psi)$ by only considering the average consecutive delay. The value of a solution is therefore the pair $[l^{F, S(F)}(0, n); \sum \max\{l^{F, S(F)}(0, n)\}]$.

The search process is halted when a time limit (bounding the computational effort) is exceeded. For all iterations, each neighbor $(F', S(F')) \in \mathcal{N}_{FBRCP}(IncSol, \psi)$ is generated by changing a train route of $IncSol$. If \mathcal{N}_{FBRCP} is empty, the algorithm applies strategy Hybrid2 for γ iterations. The value of a neighbor $(F', S(F')) \in \mathcal{N}_{FBRCP}(IncSol, \psi)$ is computed by using one of the three strategies described in Section 6.3.1. Moreover, when the JPS lower bound is used, the primary objective function is evaluated and the average

Algorithm TabuSearch**Input:**

An initial solution $(F, S(F))$,

begin

$IncSol = (F, S(F))$,

$BestSol = (F, S(F))$,

$BestValue = [l^{F,S(F)}(0, n); \sum \max\{l^{F,S(F)}(0, n)\}]$,

while time limit is not reached **do****begin**

if $\mathcal{N}_{FBRCP}(IncSol, \psi) = \emptyset$ **then** (explore the neighborhood \mathcal{N}_C)

begin

Starting from $IncSol$, execute γ moves in \mathcal{N}_C and generate $(F'', S(F''))$,

$IncSol = (F'', S(F''))$,

$NextValue = [l^{F'',S(F'')}(0, n); \sum \max\{l^{F'',S(F'')}(0, n)\}]$,

Update TL,

end

else (execute a move in the neighborhood \mathcal{N}_{FBRCP})

begin

$NextValue = [+∞; +∞]$,

while $\mathcal{N}_{FBRCP}(IncSol, \psi) \neq \emptyset$ **do**

begin

Choose a solution $(F', S(F')) \in \mathcal{N}_{FBRCP}(IncSol, \psi)$,

$\mathcal{N}_{FBRCP}(IncSol, \psi) = \mathcal{N}_{FBRCP}(IncSol, \psi) \setminus (F', S(F'))$,

if $[l^{F',S(F')}(0, n); \sum \max\{l^{F',S(F')}(0, n)\}] < NextValue$ **then**

begin

$NextSol = (F', S(F'))$,

$NextValue = [l^{F',S(F')}(0, n); \sum \max\{l^{F',S(F')}(0, n)\}]$,

end**end**

Insert in TL the train that has a different route in $IncSol$ and $NextSol$,

$IncSol = NextSol$,

end

if $NextValue < BestValue$ **then**

begin

$BestSol = IncSol$,

$BestValue = NextValue$.

end**end****end**

Figure 6.6: Sketch of the tabu search algorithm using strategy Hybrid2

consecutive delay is not considered. Once the most promising route-set has been selected, a new incumbent solution is computed by solving the CDRFR problem with the truncated branch and bound algorithm. The best solution in $\mathcal{N}_{FBRCP}(IncSol, \psi)$ is the new incumbent solution, which replaces, possibly, the current optimal solution. The rerouted train (that has a different route from the old to the new incumbent solution) is added to the tabu list (TL) for λ iterations, where λ is the length of the tabu list. A tabu-train is not allowed to be rerouted, even if its rerouting would lead to an unexplored solution. In our experiments, we found it beneficial to use a small tabu list and to forbid all the routes associated to a tabu train. In the next section, we will discuss in detail the effects of this choice against the alternative choice of inserting a single train route in the tabu list.

6.4 Computational experiments

The computational experiments of this chapter are based on the dispatching area of the Dutch rail network between Utrecht and Den Bosch (see Appendix A). We study the network simulating traffic conditions under various kinds of disturbance schemes, i.e., entrance delays and blocked tracks. A new timetable containing feasible arrival and departure times is computed for each disturbance, with the objective of minimizing maximum and average consecutive delays. Given a disturbance at the border or within the area, we generate and solve CDR instances of one hour traffic prediction. Each of these instances corresponds to an alternative graph of around 3600 pairs of alternative arcs, the exact value depending on the route chosen for each train. We incorporate in the model and evaluate the effects of relevant real-world railway constraints, such as a minimum transfer time between connected train services, in order to estimate the quality of railway services.

Table 6.1 describes the large sample of practical size instances evaluated in this chapter. Each row presents average results on 24 configurations of entrance delay, with a maximum entrance delay varying from 1000 up to 1800 seconds and an average entrance delay of around 320 seconds. The values of the entrance delay are randomly chosen in a time window of typical train delays. In order to evaluate the effects of each perturbation on the remaining part of the schedule, the delayed trains are chosen among those entering the network in the first 30 minutes of the timetable. We consider 13 configurations of randomly generated blocked tracks. However, we do not study emergency situations for which lines are completely blocked (see Figure 3.3). Each disruption is obtained by making unavailable a set of block sections, as reported in the second column. Note that the railway infrastructure with the block section numbers is illustrated and described in Appendix A. In total, we consider 312 instances with passenger and rolling stock connection constraints plus 312 instances with rolling stock connection constraints but without passenger connection constraints.

The third column of Table 6.1 shows the percentage of unavailable routes, due to the presence of blocked zones, while the fourth column gives the percentage of routes changed by the disruption recovery procedure, which restores a feasible routing with respect to the

Disturbance	Unavailable Block Sections	%Unavailable Routes	%Changed Routes
Perturbation		0.0	0.0
Disruption 1	83	40.4	12.8
Disruption 2	131 132	30.9	5.1
Disruption 3	163 167	20.2	10.3
Disruption 4	122 107	30.3	17.9
Disruption 5	67 145 167	65.2	38.8
Disruption 6	67 175 186	67.4	41.0
Disruption 7	168 164 67 175	66.6	25.6
Disruption 8	5 186 163 66	69.1	43.6
Disruption 9	135 136 122 110 67 159 186 167	82.3	41.0
Disruption 10	158 66 110 186 167 163	69.4	41.0
Disruption 11	67	40.4	15.4
Disruption 12	125 106	30.3	5.1

Table 6.1: Description of the test cases

original timetable. Specifically, the percentage of unavailable routes is obtained by checking how many of the 356 original routes pass through an unavailable block section. In our test cases this number varies up to 82% of the total number available, as shown in the third column. Similarly, the percentage of changed routes is obtained as the number of default routings (one for each train) including an unavailable block section, divided by the total number of trains. On average, a moderate amount of rerouting actions, about 22%, is sufficient to handle the timetable disturbances in the reference test cases.

The set of rerouting options used by the disruption recovery procedure is larger than that of the routing optimization procedure (see Chapter 3). For some blocked tracks the only feasible rerouting option may force a train to miss a scheduled stop. This option is allowed for disruption recovery and forbidden for routing optimization. In such cases, the delay at that stop is not considered in the experiments. Due to this fact, it may happen that the disrupted schedule exhibits a smaller delay than the undisrupted schedule.

We next present our computational results by using the local search algorithm. The most difficult instances are then solved again by using the tabu search algorithm. Each run of the compound routing and scheduling procedure is terminated after 180 seconds of computation. This choice makes the code compatible with the time constraints of railway operations. We note that computation times and train delays are always expressed in seconds.

6.4.1 Local search algorithm

In this subsection, we evaluate all the instances of Section 6.4 using the local search algorithm for the routing optimization procedure combined with two scheduling algorithms of Chapter 5, i.e., the ARI-like dispatching support system and the BB algorithm. Routing and scheduling algorithms are executed on a laptop equipped with an Intel Pentium M processor (1.6 GHz), 512 MB Ram and Linux operating system.

We next report on different configurations of the local search algorithm. We compare different conflict resolution strategies based on train reordering and rerouting, and evaluate the effects of using passenger connection constraints in the timetable. We also analyze the CDR solutions in case of heavy disturbances and discuss the performance of the local search algorithm at different time limits of computation.

Parameters analysis

The real-time purpose of ROMA imposes strict time limits to produce a new feasible timetable, which is attained by limiting the execution of the scheduling algorithm and the number of rerouting possibilities. Note that reducing the time limit of the scheduling algorithm allows the evaluation of a larger number of rerouting possibilities, at the price of less accurate schedules.

In Table 6.2, we present a comparison between two configurations, which give different importance to rescheduling and rerouting strategies. In “Config. 1”, we let $\psi = \infty$ and allow only 10 seconds of maximum computation time for BB. In “Config. 2”, we let $\psi = 5$ and allow 30 seconds to BB. The main difference between the two configurations is that “Config. 2” relies more on the scheduling procedure, since this allows larger computation times to the scheduling algorithm and reduces the number of rerouting options explored by the local search algorithm ($\psi = 5$). Each row in Table 6.2 describes average results on the 624 instances of Section 6.4. Specifically, “Default Routing” reports on the solutions provided by the disruption recovery procedure, whereas “Routing Optimization” reports on the solutions given by the iterative rerouting and rescheduling procedure based on the local search algorithm. Columns 2–4 and 5–9 show the results with the default routing and the optimal routing, respectively. All the values are expressed in seconds. Columns 2 and 5 give the maximum consecutive delays. Columns 3 and 6 indicate the average consecutive delays computed at the borders of the dispatching area. Columns 4 and 7 indicate the average computation time of the two procedures over the 624 instances. The percentage of changed routes (column 8) consists of counting how many trains have been rerouted in the overall process with respect to the original timetable. Being a simple counter, this indicator may be very sensitive to the number of trains running in the timetable. Finally, column 9 (“Time Limits”) shows the number of times the compound problem is terminated after 180 seconds of computation.

Average Results	Default Routing			Routing Optimization				
	Delay Max	Delay Avg	Time Tot	Delay Max	Delay Avg	Time Tot	%Changed Routes	Time Limits
Config. 1	279.8	50.4	2.1	245.3	44.8	33.9	23.8	76
Config. 2	279.1	50.4	5.2	251.6	45.3	33.2	23.6	64

Table 6.2: Results for varying ψ and the time allowed to BB

As expected, the scheduling solutions obtained for the default routing are slightly better for “Config. 2”, but the time needed to produce the first schedule is more than double

with respect to “Config. 1”. However, “Config. 1” gives better results after the routing optimization procedure with similar computation time limits. For this reason, in the rest of this subsection “Config. 1” will be set as the CDR system configuration.

Scheduling versus routing strategies

We compare the performance of the BB and ARI scheduling algorithms on the 624 instances of Section 6.4 without and with routing optimization. Each row of Table 6.3 thus reports the average performance of the two CDRFR algorithms on all the proposed instances.

In terms of maximum and average consecutive delays, BB provides much better solutions compared to ARI but requires more computation time. In fact, ARI always finds a solution in less than 10 seconds, whereas BB requires on average more than 30 seconds. On the other hand, the improvement of the maximum consecutive delay is about 40% when using BB over ARI. The average delay for the two CDRFR algorithms exhibits a similar behavior, but the improvement when passing from ARI to BB is about 25%. When comparing the percentage of rerouted trains similar values are obtained (around 23% of the running trains).

Average Results	Default Routing			Routing Optimization			
	Delay Max	Delay Avg	Time Tot	Delay Max	Delay Avg	Time Tot	%Changed Routes
ARI	489.4	66.9	0.6	417.0	60.5	8.1	23.4
BB	279.8	50.4	2.1	245.3	44.8	33.9	23.8

Table 6.3: BB versus ARI

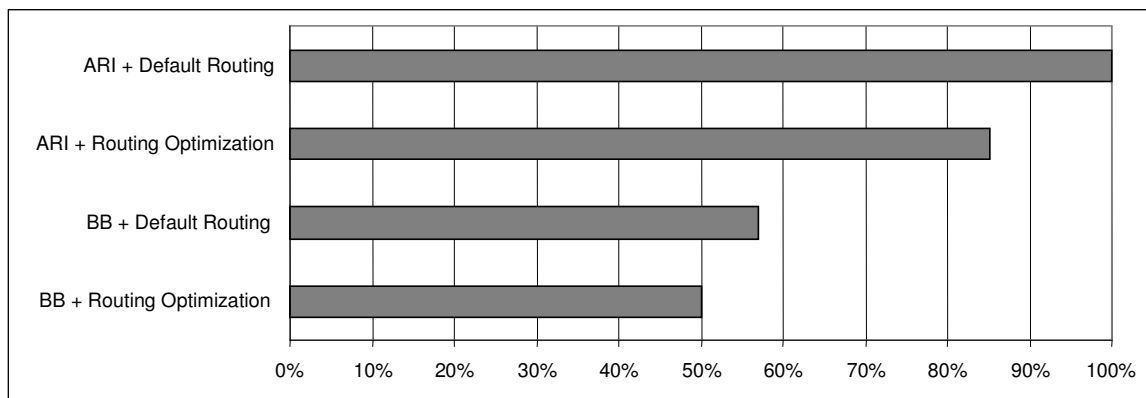


Figure 6.7: Maximum consecutive delay with four configurations of ROMA

Figure 6.7 shows the reduction of the maximum consecutive delay from the worst configuration, obtained with ARI and the default routing, up to the best configuration, obtained with BB and routing optimization. The overall reduction is about 50%, about 43% of which is due to the scheduling algorithm. The routing optimization procedure contributes with an additional reduction of about 7%. The overall improvement is therefore largely due to the use of an advanced scheduling algorithm.

Scheduling	Routing	Out	Htn	Cl	Gdm	Zbm	Hc	Ht
ARI	Default	67.0	17.1	21.0	24.4	94.1	20.7	97.5
ARI	Optimization	59.8	14.0	19.9	26.3	88.7	23.1	84.7
BB	Default	50.4	16.9	20.8	6.0	78.6	9.6	66.9
BB	Optimization	44.8	15.1	19.4	6.6	73.7	8.7	56.5

Table 6.4: Average consecutive delays at stations

Table 6.4 shows the average consecutive delays at each station and at the borders of the dispatching area (“Out”). Each row shows the results obtained with the different scheduling/routing strategies shown in the first two columns. When comparing the two scheduling algorithms using the default routing, although the objective function is to minimize the maximum consecutive delay, BB is also able to produce better solutions in terms of average consecutive delays at all stations. When comparing the solutions using the local rerouting optimization strategy the average delay is more effectively reduced. In fact, the combined strategy of routing optimization and algorithm ARI [resp. BB] decreases the average delay in four [resp. five] stations.

Effects of passenger connections

Table 6.5 evaluates the effects of adding/removing passenger connection constraints. Here we compare the different cases in which all connections are maintained or no connection is enforced. This approach differs from that of other works on delay management (see, e.g. Schöbel (2001)), in which one aims to decide which connections have to be maintained or canceled in order to minimize the inconvenience for the passengers. Each row of Table 6.5 shows the average results for the 312 instances of Section 6.4 without (“Off”) or with (“On”) passenger connections and using ARI or BB. When passenger connections are in use the maximum and average consecutive delays increase of about 10%. This is the cost of taking into account such constraints in the presence of disturbances. The computation times required to solve the problem, with or without passenger connection constraints, do not vary significantly. Moreover, the use of BB with passenger connections gives better results even when compared to ARI without passenger connections.

Average Results	Passenger Connections	Default Routing			Routing Optimization		
		Delay Max	Delay Avg	Time Tot	Delay Max	Delay Avg	Time Tot
ARI	Off	456.9	62.6	0.6	391.0	57.1	11.0
ARI	On	521.9	71.3	0.6	427.6	62.4	11.2
BB	Off	266.2	48.5	2.2	234.7	43.2	34.4
BB	On	293.5	52.2	2.1	255.9	46.3	33.3

Table 6.5: Effects of passenger connections

Table 6.6 shows the effects of passenger connections at all stations. We show the average consecutive delay measured at each station and at the borders of the dispatching area

Passenger Connections	Out	Htn	Cl	Gdm	Zbm	Hc	Ht
Off	43.2	14.6	19.1	7.2	71.8	8.6	53.8
On	46.3	15.5	19.6	5.9	75.6	8.7	59.2

Table 6.6: Influence of passenger connections at stations

(“Out”). All these solutions are computed using BB and the routing optimization procedure. Each row refers to the average results on the same instances of Table 6.5. In our tests, the passenger connection constraints are active only in Zaltbommel (Zbm) and Den Bosch (Ht) stations, but delays may propagate among the trains. When the passenger connections are “On” the average consecutive delay increases mainly in the stations where the passenger connections are active. On average, the average consecutive delays increase more than 5 seconds at Den Bosch and almost 4 seconds at Zaltbommel, whereas this increase is less evident at the other stations. However, such a small increase seems negligible compared with the benefit for passengers by assuring the scheduled connections.

Effects of timetable disruptions

We evaluate the ability of the proposed CDR system to attain feasible solutions when a large part of the network is unavailable. Precisely, we study the influence of each disruption scheme described in Table 6.1.

Average Results	Default Routing			Routing Optimization			
	Delay Max	Delay Avg	Time Tot	Delay Max	Delay Avg	Time Tot	%Changed Routes
Perturbation	364.5	45.0	0.6	267.0	35.8	22.4	2.9
Disruption 1	495.8	66.8	0.6	398.2	57.7	15.7	15.6
Disruption 2	338.5	40.4	0.6	254.6	33.2	8.8	7.2
Disruption 3	378.6	49.7	0.5	273.7	39.7	17.0	12.6
Disruption 4	420.3	49.3	0.5	339.3	43.7	9.2	18.4
Disruption 5	572.4	88.8	0.6	517.0	81.1	9.5	31.4
Disruption 6	570.1	84.8	0.8	486.5	78.8	11.2	41.1
Disruption 7	569.1	75.3	0.7	484.6	69.2	9.2	26.8
Disruption 8	566.1	81.7	0.7	489.4	75.3	6.9	44.1
Disruption 9	590.9	86.8	0.7	552.2	83.2	2.0	41.5
Disruption 10	607.2	88.4	0.7	513.3	78.7	6.9	41.5
Disruption 11	524.6	69.0	0.6	464.8	62.2	15.0	16.6
Disruption 12	365.2	45.1	0.5	280.8	38.2	11.2	6.8

Table 6.7: Effects of timetable disruptions using ARI

Table 6.7 presents the solution obtained using the ARI dispatching heuristic while Table 6.8 illustrates the BB solutions. Each row refers to average results over 48 instances, which correspond to the 24 perturbations of Section 6.4 and timetables with and without passenger connection constraints.

Average Results	Default Routing			Routing Optimization			
	Delay Max	Delay Avg	Time Tot	Delay Max	Delay Avg	Time Tot	%Changed Routes
Perturbation	183.8	20.8	0.5	127.0	15.2	17.7	3.3
Disruption 1	294.9	49.7	0.8	271.3	46.0	36.4	15.1
Disruption 2	173.6	18.7	0.5	124.5	13.7	8.4	7.8
Disruption 3	196.0	24.4	0.4	148.0	18.6	18.4	12.3
Disruption 4	184.0	21.3	0.4	125.0	15.2	9.9	19.2
Disruption 5	376.9	77.7	4.1	353.5	73.4	61.2	31.5
Disruption 6	345.4	79.1	3.9	321.7	73.0	55.9	41.3
Disruption 7	352.6	67.8	4.5	320.5	61.1	69.3	26.7
Disruption 8	331.7	69.2	3.4	312.8	62.4	29.9	44.3
Disruption 9	337.4	71.9	3.2	323.6	67.7	11.5	41.7
Disruption 10	351.0	78.7	4.2	326.2	71.0	44.7	41.8
Disruption 11	327.5	55.5	2.3	308.0	50.0	68.7	17.1
Disruption 12	183.9	20.9	0.4	127.5	15.4	8.9	7.2

Table 6.8: Effects of timetable disruptions using BB

When a large part of the network is unavailable the maximum consecutive delay increases. In this case, due to the larger number of trains running over the same block sections, the network becomes increasingly congested. When comparing the two situations with and without blocked tracks (rows “Disruption” against rows “Perturbation”), the delay may even be double (e.g. Disruption 5 with BB) and the percentage of rerouted trains ranges from 3% up to more than 40%. Moreover, for some strong disruptions (named 5, 7 and 11) the average computation time exceeds one minute of CPU time (as shown in Table 6.8).

Computation time versus solution quality

We show the ROMA performance when varying the time limit allowed to compute optimal CDR solutions. In Figure 6.8, we plot the maximum and average consecutive delays of the best solutions found under different time limits for the 48 perturbations and the 144 disruptions classified as 5, 7 and 11 in Table 6.1. The former case allows the maximum routing flexibility, being available all the 356 original routes, while the three latter cases are those requiring on average the longest computation time (see Table 6.8). The consecutive delays are depicted after each half minute of computation, up to 5 minutes. For each case, we show the average results over 48 instances corresponding to the 24 perturbations of Section 6.4 with and without passenger connection constraints.

At time 0, the consecutive delays are shown when using the routes prescribed by the disruption recovery procedure. For real-time purposes the algorithm is quite effective, especially in the perturbation case in which limited rerouting actions allow to obtain good solutions already after 30 seconds of computation. As far as the disruptions 5, 7 and 11 are concerned, a more extensive search among the available rerouting options is needed in order to find a

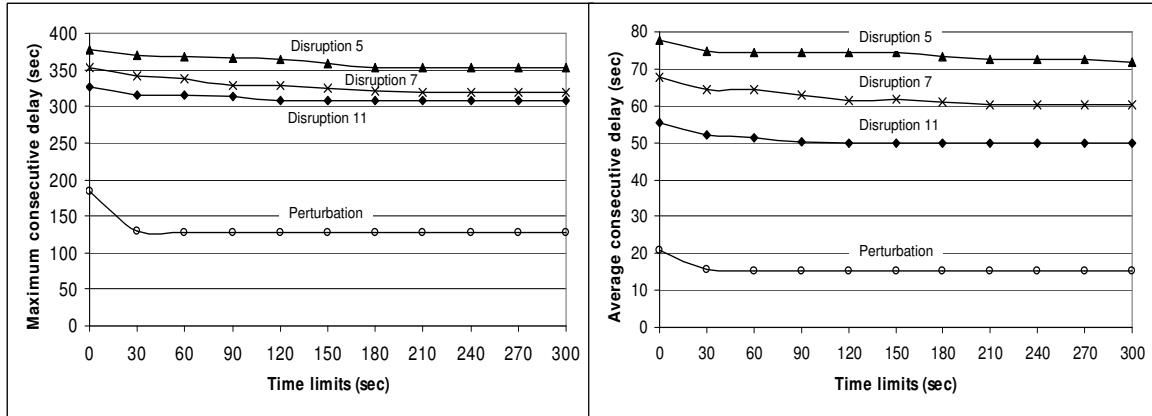


Figure 6.8: Maximum and average consecutive delays at different time limits

local minimum. However, in all cases, increasing the computation time over 180 seconds does not improve the solution quality significantly.

6.4.2 Tabu search algorithm

This subsection presents our computational experiments on the tabu search algorithm. We consider a part of the timetable disturbances tested in Section 6.4.1. Precisely, we refer to the 48 timetable perturbations with multiple late trains (i.e., the 24 configurations of entrance delay and timetables with and without passenger connection constraints) and to the three most time consuming disruptions (i.e., the blocked track configurations classified as 5, 7 and 11 in Table 6.1).

In all these experiments, the parameters (ψ, λ, γ) are fixed equal to $(8, 3, 5)$ respectively. These values are defined by running a preliminary set of experiments with 18 pilot instances and for several triples (ψ, λ, γ) . The above values achieve the best results on average for all the algorithmic components addressed in this subsection. Routing and scheduling algorithms are executed on a PC equipped with an Intel Pentium D processor (3 Ghz), 1 GB Ram and Linux operating system.

We next report on the advantages of using alternative algorithmic components in our tabu search. We also present the performance of all the proposed neighborhood search strategies.

Assessment of tabu search components

We illustrate different alternative choices to configure our tabu search algorithm. Precisely, we next analyze: (i) the implementation of an aspiration criterion, (ii) the tabu list structure, (iii) the method for neighbors evaluation, i.e., for choosing the best candidate in the neighborhood. For each point, we report the average results on all the proposed disturbances, in terms of maximum consecutive delay. Note that each comparison is based on strategies Restart and Complete.

As for point (i), we only explore two possibilities, i.e., using or not using this feature. We consider only a simple aspiration criterion, in which tabu moves are explored and implemented when leading to an improvement of the current best solution. The results are compared in Figure 6.9 to the case without aspiration criterion. The value of the best solution found at time t is shown for the two cases, for $t = 0, \dots, 180$ seconds. We conclude that the computational cost of the aspiration criterion is not compensated by significant improvements to the best solution. This criterion is therefore not used in the following experiments.

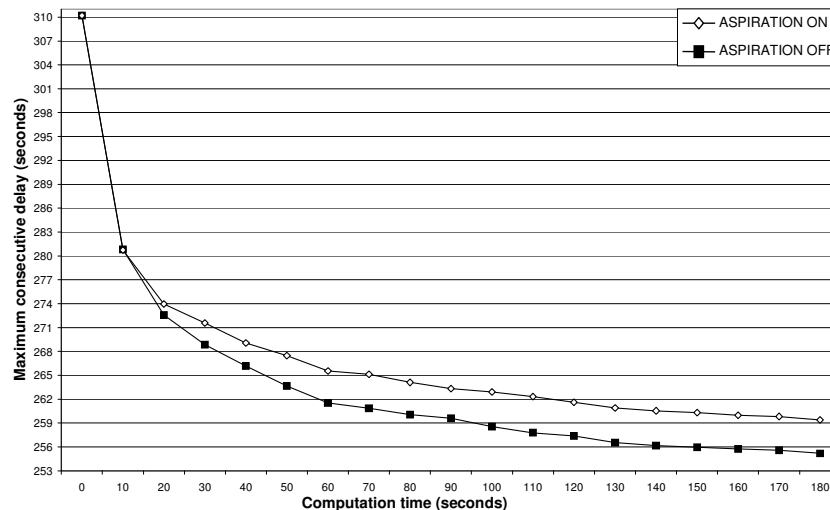


Figure 6.9: Comparison between using or not using an aspiration criterion

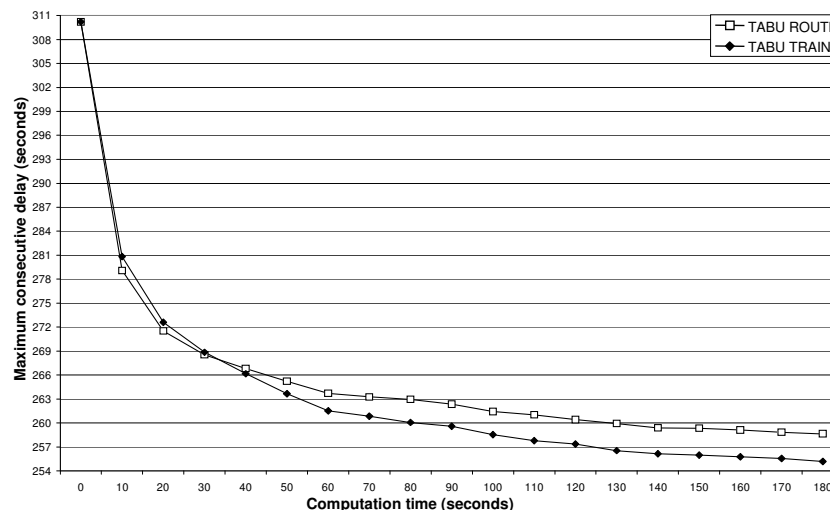


Figure 6.10: Comparison between two tabu list structures

As for point (ii), we explore the two possibilities of inserting a train in the tabu list or inserting a train route in the tabu list, the former case being more restrictive. The results of this comparison are shown in Figure 6.10. The value of the best solution found at time t for the two cases shows, clearly, that the former choice is preferable. This is thus the option used in the following experiments.

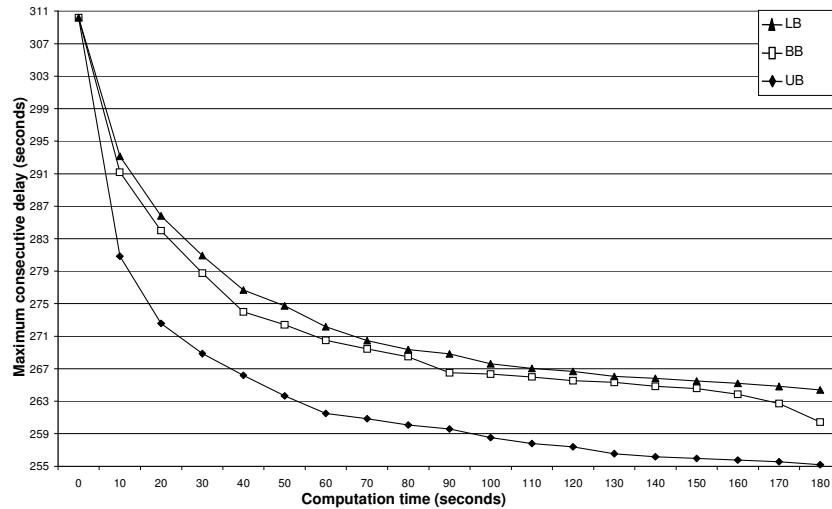


Figure 6.11: Assessment of different methods for neighbors evaluation

As for point (iii), we evaluate three algorithms for choosing the best candidate in the neighborhood of the incumbent solution. These algorithms are denoted as LB, UB and BB (as in Section 6.3.1). Figure 6.11 shows the values of the best solution found at time $t = 0, \dots, 180$ seconds for the three cases. LB is a fast but not accurate evaluation criterion while UB is more fast and accurate, even if the latter sometimes fails in finding a feasible solution. In case of infeasibility, the neighbor is discarded from the neighborhood. From the obtained results, BB produces the most accurate estimates but, being the most time consuming method, causes the tabu search being approximately three times slower than when using algorithm UB. The described behavior motivates our choice of using UB for the purpose of neighbors evaluation.

Disturbance management

We present computational experiments on severe traffic disturbances and considering relevant real-world railway constraints. Precisely, we analyze separately the management of the 48 perturbations, and the joint effects of these 48 perturbations and the three heavy disruptions proposed at the beginning of this subsection. Except for the search strategy, the configuration of the tabu search algorithm is the best performing in the previous set of experiments (assessment of tabu search components). The computational results are described for all the neighborhood search strategies of Section 6.3.2.

Table 6.9 reports on the general behavior of different tabu search strategies. Column 2 shows the average number of times the algorithm improves the current best solution over the 48 perturbations and 144 disruptions, while column 3 indicates the average number of train routes that are changed in the overall best solution compared to the original (default) routes prescribed in the timetable. In case of disruptions, if a default route is not available this is initially replaced with its shortest available alternative route. It is interesting to underline that the best solution is improved more frequently using strategies Hybrid2 and Hybrid3. In

fact, their final solutions also present a smaller number of changed routes compared to the default routes and should therefore be easier to implement by railway traffic operators.

Neighborhood Search Strategy	Timetable Perturbations					
	Improving Moves	Changed Routes	Time Limits BB	Moves in \mathcal{N}_C	Moves in \mathcal{N}_{FBRCP}	Time Spent by BB
Complete	6.9	14.5	0.2	131	0	16.4
Restart	5.1	15.1	0.9	280	148.3	34.9
Hybrid1	6.3	13.7	0.6	45	20	39.5
Hybrid2	7.1	12.7	0.5	44	20.2	41.4
Hybrid3	7.1	12.7	0.6	74	4.6	37.9
Neighborhood Search Strategy	Timetable Disruptions					
	Improving Moves	Changed Routes	Time Limits BB	Moves in \mathcal{N}_C	Moves in \mathcal{N}_{FBRCP}	Time Spent by BB
Complete	5.6	12.1	3.8	76	0	63.1
Restart	4.8	9.8	4.7	110	91.5	91.7
Hybrid1	5.4	9.4	4.3	19	33.5	99.7
Hybrid2	6.0	9.3	4.5	19	33	101.4
Hybrid3	6.0	9.3	5.1	22	3.1	102.6

Table 6.9: Comparison of tabu search neighborhood search strategies

Column 4 of Table 6.9 gives the number of time limits reached by the BB algorithm which is truncated after 10 seconds of computation. When dealing with timetable perturbations this time limit is only rarely reached. On the other hand, the instances with disruptions are significantly harder, as also indicated in the last column of Table 6.9 for algorithm BB. This is probably due to the need of scheduling trains on a single track in both directions between the two station areas of Zaltbommel and Den Bosch, which makes these CDR instances particularly hard to solve to optimality.

Columns 5 and 6 of Table 6.9 report the average number of moves executed in the neighborhoods \mathcal{N}_C and \mathcal{N}_{FBRCP} , respectively. We consider under strategy Hybrid3 and \mathcal{N}_{FBRCP} only those moves that do not require searching in \mathcal{N}_C . The higher number of moves carried out by strategy Restart is due to the fact that searching in \mathcal{N}_C is particularly fast with this strategy and leaves more time for evaluating \mathcal{N}_{FBRCP} . The last column shows the average time spent by the tabu search configurations for running algorithm BB, within 180 seconds of computation. Quite surprisingly, this time is shorter with strategy Complete, despite the higher number of BB executions. When rerouting a train with the latter strategy, the resulting instances of the CDRFR problem are thus easier to solve compared to the other strategies. On the other hand, the other four strategies require similar amount of time for executing algorithm BB.

We now compare the results obtained by our tabu search algorithm, for each of the five search strategies, with the performance of the local search algorithm (based on the \mathcal{N}_{BRCP} neighborhood) and of the proven optimal CDRFR solutions with default routes.

Figure 6.12 illustrates the average results, in terms of the maximum and average consecutive

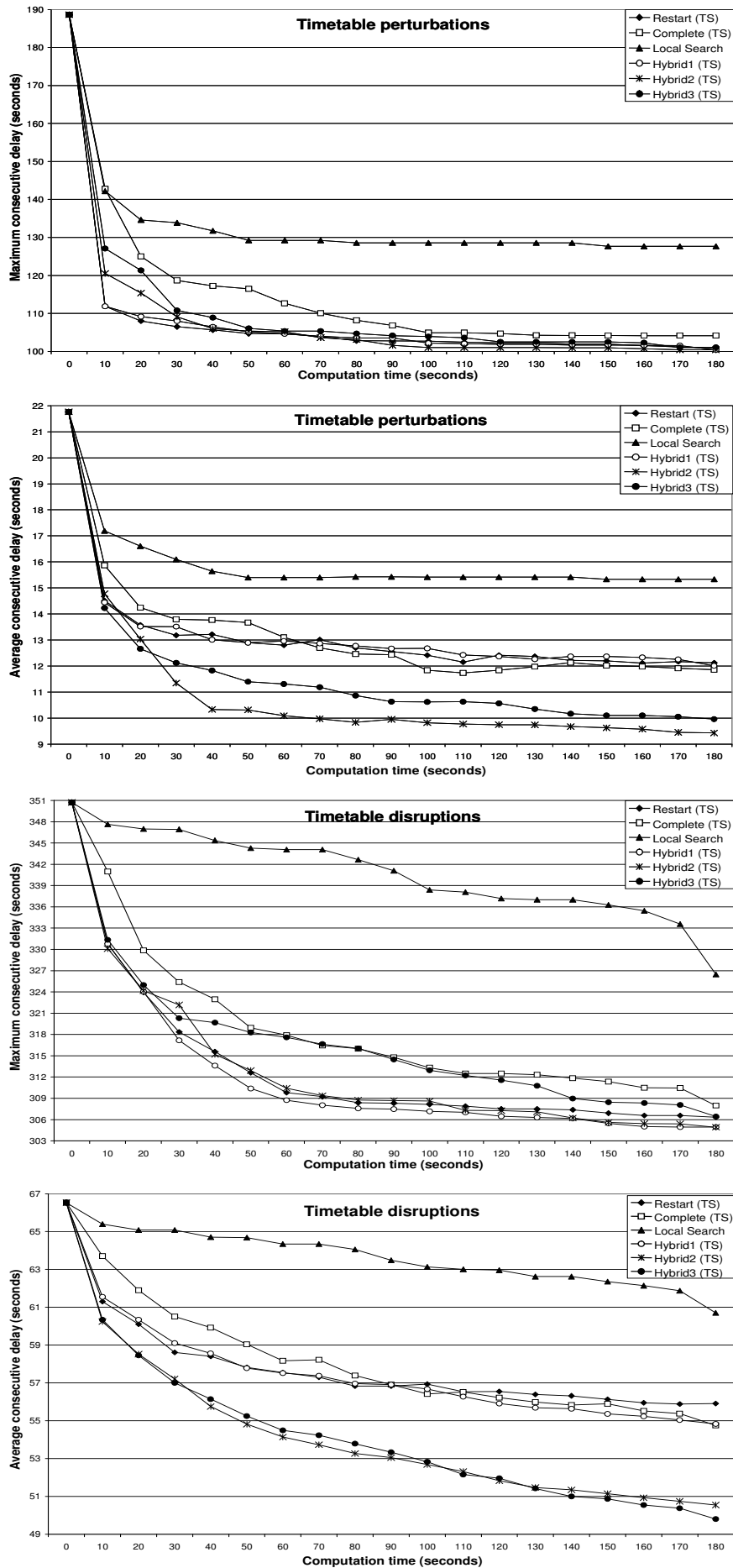


Figure 6.12: Comparison of the six algorithms in case of perturbations and disruptions

delays, over the 48 perturbation schemes for the two cases without and with disruptions. At time $t = 0$, we report the average results on the proven optimal CDRFR solutions with default routes. The average solutions of the rerouting algorithms are depicted each 10 seconds of computation, up to three minutes.

The latter figures present, clearly, the benefits of rescheduling and rerouting trains to recover delays. Also evident is the delay reduction when passing from the optimal CDRFR solutions to the achieved CDR solutions. The drop is more than one third for maximum consecutive delay and more than one half for the average consecutive delay.

The hybrid strategies give the best overall results, outperforming strategies Restart and Complete, thus demonstrating the advantage of using a hybrid search scheme. However, the delay reduction has a price in terms of computation time. For the perturbation instances, the proven optima to the CDRFR problem are obtained, on average, within two seconds of computation, while the tabu search algorithm needs at least 10 seconds to achieve a 30% reduction of maximum and average consecutive delays. The local search algorithm needs more than 20 seconds to obtain similar results. Moreover, the strategies Hybrid2 and Hybrid3 achieve a 40% reduction of both delays after 20 seconds of computation, and strategy Hybrid2 needs 40 seconds to halve the average consecutive delay.

The disruption instances are significantly more time consuming compared to the perturbations instances, despite the smaller number of rerouting options. The local search algorithm improves slowly the initial CDRFR solutions during the 180 seconds of computation, while strategies Hybrid1 and Hybrid2 attain a 10% reduction of the maximum average delay only after 40 seconds. Furthermore, strategies Hybrid2 and Hybrid3 need around 90 seconds to obtain a 20% average consecutive delay reduction. However, evidently, the human dispatchers need more extensive support to manage severe disruptions. A restricted line capacity also limits the delay reduction that can be achieved.

To summarize, our tabu search algorithm, with hybrid search strategies, is able to provide an effective support for the traffic management in case of timetable perturbations and disruptions within a time limit compatible with rail operations.

6.5 Conclusions

This chapter describes a number of algorithmic improvements implemented in the ROMA dispatching support system. We investigate advanced properties and algorithms to solve the compound train rerouting and rescheduling problem. The computational results, performed on real-world railway instances, demonstrate the high potential of combined rerouting and rescheduling strategies in order to reduce the maximum and average consecutive delays and to improve the dynamic use of train routes.

The results obtained with the local search algorithm for the CDR problem achieve a significant delay reduction, even though the benefit is mainly due to the sequencing optimization rather than to rerouting, particularly when dealing with heavy disruptions in the network.

While advanced scheduling algorithms are able to solve large instances within a short computation time, the local search algorithm does not exploit all the potential offered by routing flexibility. A limited attempt to investigate larger neighborhoods, in which several routes are changed simultaneously, shows that there are further margins for improving upon the local minima found by our local search algorithm.

Further research has therefore addressed the analysis of larger neighborhoods within a short computation time as well as to the development of more sophisticated rerouting metaheuristics. Novel tabu search neighborhood structures and search strategies are studied in detail. Focused neighborhoods yield promising results within a short computation time when dealing with perturbations, while more sophisticated hybrid strategies are required to effectively manage disruptions in real-time. The computational results show, clearly, that hybrid strategies outperform the local search and the stand-alone neighborhoods.

From theoretical and practical points of view, it is worthwhile to search for non-dominated solutions (with respect to the minimization of maximum and average consecutive delays) within a shorter computation time, as well as to implement exact procedures and effective lower bounds to the CDR problem. This would allow to further improve the quality of the solutions found by the proposed tabu search algorithm.

Chapter 7

Train speed coordination

The alternative graph formulation of the train traffic optimization problem, described in Chapter 4, corresponds to a fixed-speed model that does not allow a dynamic adjustment of running and setup times. Train conflicts are solved by delaying the starting time of the operations involved in each conflict. In other words, the fixed-speed model does not consider the impact of braking and re-acceleration of trains facing the yellow and red signal aspects of the Dutch signaling system NS54. Issues such as schedule robustness and variability of train dynamics are now approached.

This chapter presents the variable-speed model adopted in the ROMA dispatching support system for train scheduling and train speed coordination (TSC) problems that acts as follows: once the fixed-speed model for train scheduling has been solved and a graph selection has been obtained (see Chapters 4, 5 and 6), the scheduled (off-line) train speed profiles might no longer be admitted due to conflicts by delayed trains. In other words, the considered train speed profiles might not respect the change of the signaling system constraints. A feasibility check is thus applied to ascertain that the minimum distance headway between all pairs of consecutive trains is respected over any common route. In case the feasibility check fails, some running trains need to adapt their velocity, and the alternative graph has to be updated accordingly. The speed updating procedure is thus applied to ensure that the train speed profiles fit to real operations. The resulting alternative graph is used in the successive iterations of the TSC procedure until an admissible train speed profile has been computed for each train running in the railway network.

The ROMA solution from a network point of view can be improved by modifying the speed profiles locally for the individual train routes. Optimal train running profiles can be designed in order to further reduce train delays and energy consumption at the same time. A constructive heuristic algorithm for the dynamic modification of running times during operations is proposed that satisfies the schedule constraints of train orders and routes, and that guarantees the feasibility of the running profile, while taking into account the properties of the signaling and train protection systems in use.

This chapter is organized as follows. The feasibility check, the train speed updating and the adjustment of train speed profiles and orders in the alternative graph are formalized as part

of the process to compute a feasible dispatching solution. A small real-world example is given to show the alternative graph formulation of the TSC problem. In a following section dedicated to computational experiments, the proposed procedures of the ROMA dispatching support system are adopted to compute feasible dispatching solutions for perturbed traffic conditions at the Schiphol railway bottleneck and for the dispatching area between Utrecht and Den Bosch. A further real-world example for the rail corridor between Utrecht and Geldermalsen is presented to demonstrate the benefits of optimal train speed profiles at conflict points. Conclusions and suggestions for future improvements of the proposed dispatching procedures are finally discussed.

7.1 Feasibility check

The feasibility check is implemented in the form of blocking times, which is an efficient and precise technique to check if minimum safety distances are maintained and can be used to easily interact with the dispatcher (see, e.g. Pachl (2002) and Jacobs (2004)).

The scheduled headway between two consecutive trains on a block section consists of the buffer time (t_β) and the mandatory minimal time headway, which represents the minimum time interval between the blocking times of two consecutive trains along a line between two stations. Buffer time is usually adopted to reduce consecutive delays in case of initial delays during operations, i.e., to prevent delay propagation. In practice, if a schedule is free of overlaps then $t_\beta \geq 0$, otherwise the minimum braking distance between trains traveling at operating speed in each block section is not respected.

Let T_1 and T_2 be two consecutive trains traveling on a certain block section, where T_1 precedes T_2 . We next model three main traffic situations and compute their corresponding buffer time values: the case in which there are no scheduled stops and the cases in which the preceding or following train performs a scheduled stop.

Figure 7.1 (a) shows how the buffer time is computed in a situation without scheduled stops, and in Figure 7.1 (b) we show its alternative graph formulation. Nodes i and j represent T_1 and T_2 traveling on a given block section and nodes $\mu(i)$, i and $\sigma(i)$ represent three consecutive block sections to be crossed by T_1 . Likewise, nodes $\mu(j)$, j and $\sigma(j)$ represent three consecutive block sections to be crossed by T_2 . Note that the block section of operations o_i and o_j coincides whereas $o_{\mu(i)}$ and $o_{\mu(j)}$ ($o_{\sigma(i)}$ and $o_{\sigma(j)}$) do not necessarily represent operations on the same block section. In Figure 7.1, $t_{\mu(i)}$, t_i , $t_{\sigma(i)}$, $t_{\mu(j)}$, t_j and $t_{\sigma(j)}$ represent the time at which a train enters the corresponding block section. The buffer time, between the blocking times of T_1 and T_2 , is given by the expression:

$$t_\beta = t_j - p_{\mu(j)} - a_{ij} - p_i - t_i \quad (7.1)$$

where p_i represents the running time of T_1 , $p_{\mu(j)}$ is the approaching time of T_2 and a_{ij} is the sum of the clearing and switching time of T_1 , and the sight and reaction time of

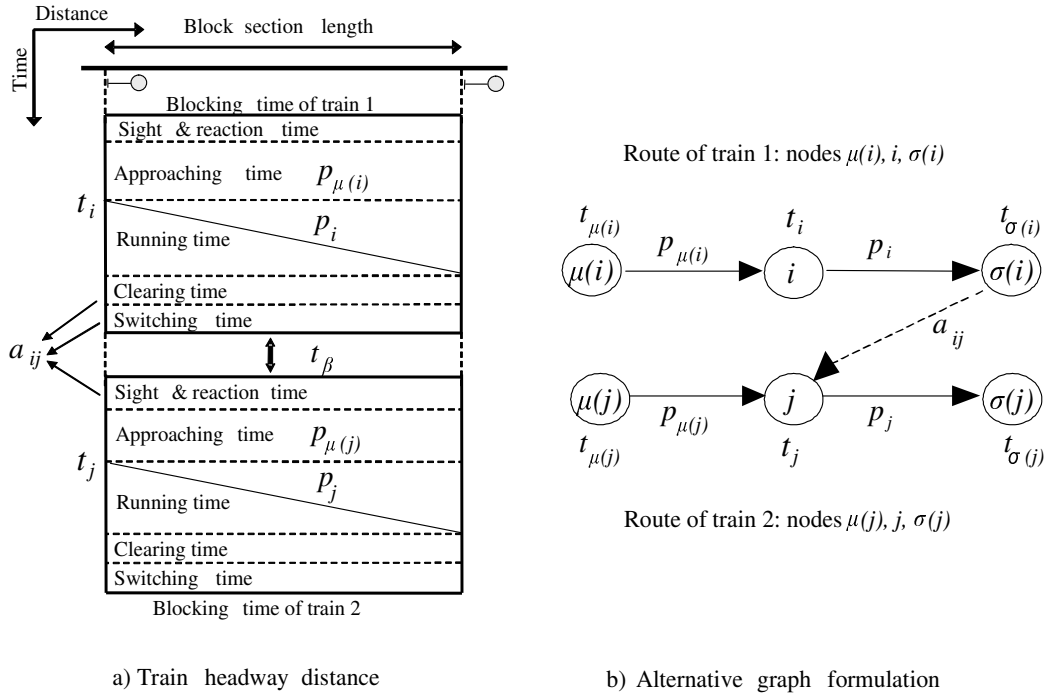


Figure 7.1: Model of buffer time in the alternative graph

T_2 . Running times, approaching times and clearing times depend on the train speed profiles, whereas sight times, reaction times and switching times are considered constants. A detailed description of these terms can be found in Pachl (2002).

In Figure 7.2 (a), the preceding train (T_1) performs a scheduled stop. In this case, the blocking time is composed of the approaching time, sight time, reaction time, running and dwell times (also defined as time between block signals), clearing time and switching time. Since both running and dwell times have to be considered in the block section where the stop is located, the buffer time can be calculated as:

$$t_{\beta} = t_j - p_{\mu(j)} - a_{ij} - s_i - p_i - t_i \quad (7.2)$$

where p_i represents the running time in the block section in which the platform is located and s_i is the dwell time.

In the case in which the following train (T_2) starts from a scheduled stop (Figure 7.2 (b)), t_{β} can be calculated as:

$$t_{\beta} = t_j - a_{ij} - p_i - t_i \quad (7.3)$$

Let $v_{operating}$ be the current operating speed on an open track, and let $v_{approach}$ be the maximum allowed speed at which a train approaches a red signal aspect. The approaching time then is the time needed to decelerate a train from $v_{operating}$ to $v_{approach}$. The approaching time collapses to zero (case of Equation 7.3) when a train has performed a complete stop

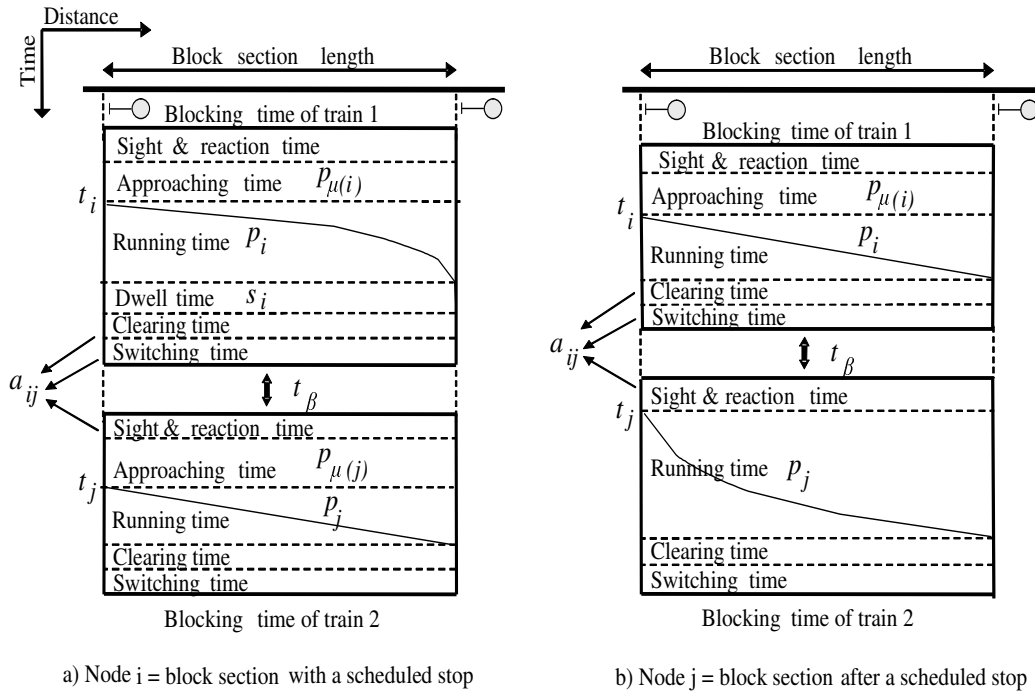


Figure 7.2: Buffer time in case of a scheduled stop

and its speed is zero (Pachl, 2002). In this case, the safe distance headway between the concerned trains may be less than two unoccupied block sections.

Figure 7.3 shows the track occupation in case of different signal aspects. Given two consecutive trains traveling on a track, three principal possibilities of train running can be distinguished for the Dutch signaling system NS54 depending on the actual occupation of the preceding block sections:

Feasible : As shown in Figure 7.3 (a), there are at least two empty block sections between T_1 and T_2 . This is sufficient as minimum distance headway between the two trains. In this case, the buffer time is: $t_\beta \geq 0$.

Overlap : T_2 occupies the block section that corresponds to node $\mu(j)$ and T_1 still occupies the block section that corresponds to node $\sigma(j)$. An overlap of blocking times would arise (Figure 7.3 (b)) and the buffer time between the two trains would become: $-p_{\mu(j)} \leq t_\beta < 0$ (in case of Equation 7.1) or $-p_{\mu(j)} - p_i \leq t_\beta < 0$ (in case of Equation 7.2).

Conflict : T_2 occupies the block section that corresponds to node $\mu(j)$ and T_1 still occupies the block section that corresponds to node i . A conflict arises (Figure 7.3 (c)) since both trains are asking for the same resource. The minimum distance headway at maximum speed would not be respected, which means that T_2 must have already decelerated at the preceding signal in order to stop safely before the red signal aspect. The buffer time between the two trains would be negative: $t_\beta < -p_{\mu(j)}$ (in case of Equation 7.1) or $t_\beta < -p_{\mu(j)} - p_i$ (in case of Equation 7.2) or $t_\beta < 0$ (in case of Equation 7.3).

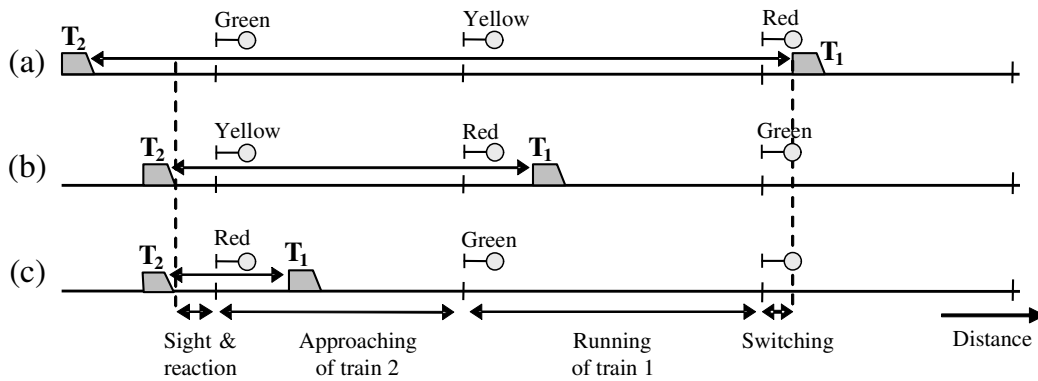


Figure 7.3: Three-aspect fixed block signaling system at different train positions

The feasibility check between two consecutive trains identifies a virtual overlap of blocking times when the minimum time headway between two trains is not respected. In this case, the signal aspect at the entrance of the preceding block section becomes yellow and the actual running time of the following train on the track would need to be modified.

7.2 Speed updating

In this section, we describe the standard driver's behavior in case of the three-aspect fixed block signaling system adopted in this thesis. However, our speed updating procedure is very general and various driver behaviors can be implemented if necessary.

Given two consecutive trains running on a track (block sections $n - 1$, n and $n + 1$), we assume that a yellow signal aspect is faced by the following train at block section $n - 1$. Figure 7.4 shows the three corresponding principally different speed profiles (driver behaviors) of the following train in case it passes a yellow signal. According to the current Dutch railway operations rules, we suppose that the driver of the following train decreases velocity from the operating speed, $v_{operating}$, to the signal approach speed, $v_{approach}$ (40 km/h), which is maintained until the signal of block section n becomes visible and the train has to stop if the block signal aspect does not change in the meantime from red to yellow or green. In the first scenario (S1), the signal aspect shows red until the train completely stops (position P1), and only after a certain waiting time the signal aspect becomes yellow. Only then the driver may accelerate the train up to $v_{approach}$. In the second scenario (S2), during the second braking phase the red signal aspect switches to yellow, while the train is in position P2. In this situation, we assume that after a short reaction time the driver increases the train velocity up to $v_{approach}$. In the remaining scenario (S3), when the train reaches the sight and braking distance (P3) at $v_{approach}$, the signal aspect becomes green. Finally, the driver is allowed to accelerate the train from $v_{approach}$ to $v_{operating}$.

We now consider a set of possible situations for the following train running on block section n when the signal aspect is yellow. We suppose that this block's signal aspect becomes green when the preceding train has cleared block section $n+2$ and the following train can accelerate from $v_{approach}$ to $v_{operating}$. Figure 7.4 shows the minimum distance for starting

of the re-acceleration in case of scenario S2 (min S2) and scenario S1 (min S1), depending on the regulation of railway operations. If the signal aspect has not become green when the following train reaches position P3' at $v_{approach}$ then the three possible train movements presented for block section $n-1$ are adopted in block section n again.

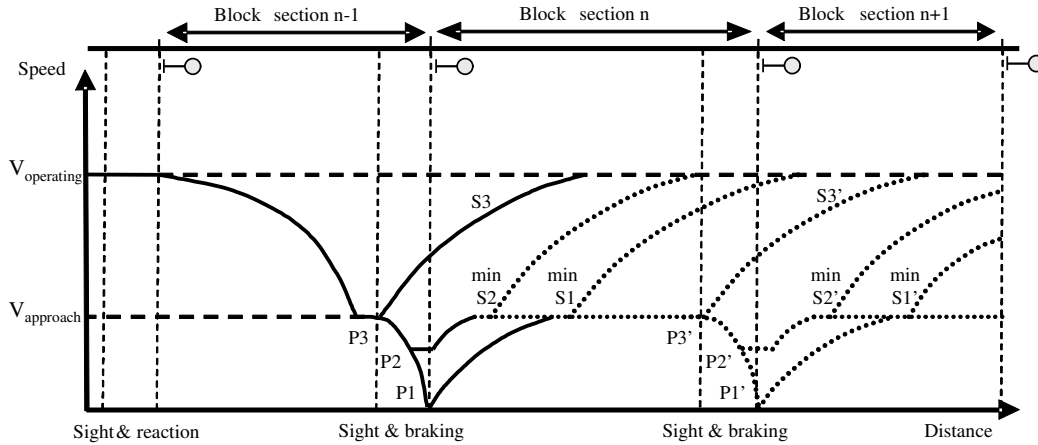


Figure 7.4: Driver behavior in case of the three-aspect fixed block signaling system

We have shown the approach to a distant signal within a block. In dense networks some signal blocks, mostly close to stations, are shorter than the braking distance required for $v_{operating}$. In these cases, the signal block must be extended over two blocks to ensure safe braking to reach the red aspect at $v_{approach}$. This is done by including speed limitations at yellow signal aspects when the signals delimit short block sections. We take into account these situations adopting suitable speed profiles for the involved trains.

7.3 Modeling the adjustment of train speed profiles

This section explains how the train dynamics are updated in the alternative graph, when a complete and consistent selection is already available. We only address in detail the case of a complete stop in the presence of a red signal aspect and a re-acceleration to reach $v_{operating}$ within two block sections (we consider the combination of scenarios S1 and S3' depicted in Figure 7.4), since the case in which the red signal aspect turns green before the complete stop and the case in which a train requires more than one block section to stop or to reach $v_{operating}$ are straightforward extensions. The solid line in Figure 7.5 (top) shows the speed profile followed by the train in case of a yellow signal aspect, where the approaching time begins. The dashed line shows the train dynamic behavior modeled by the alternative graph. This behavior does not take into account a speed restriction in case of a yellow signal aspect and the train would have to stop instantaneously when arriving in front of the red signal aspect. Furthermore, when the red signal aspect is released, in first instance, it is assumed that the train immediately starts moving at the operating speed, without considering its acceleration curve. In other words, if virtual overlaps of blocking times occur, the speed profiles of the following trains become unacceptable, since speed adjustments generated by a yellow signal aspect are not considered in the fixed-speed model.

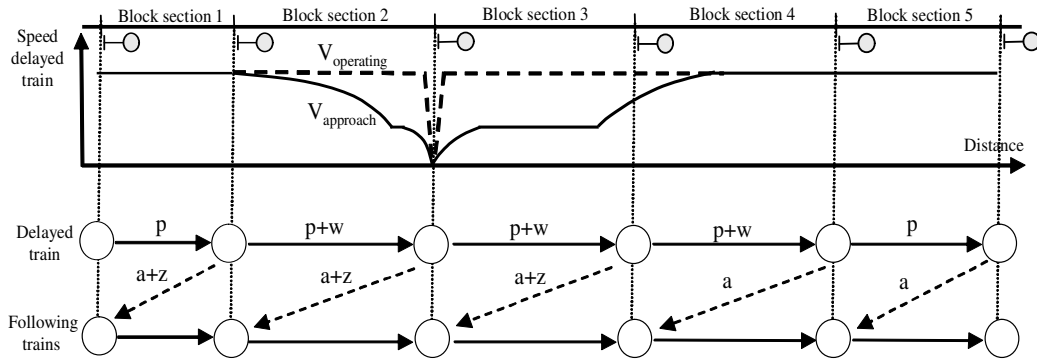


Figure 7.5: Modeling train speed for the three-aspect fixed block signaling system

Each time a train passes a yellow signal aspect its actual speed profile has to be reexamined with respect to the scheduled speed. Updating the speed of running trains in the alternative graph is equal to modifying the arc lengths that correspond to the block sections where the train dynamics change due to a yellow or red signal aspect. The running times (the length of fixed arcs) and the clearing times (part of the length of alternative arcs) of these block sections are thus increased by the difference between their values at $v_{approach}$ and $v_{operating}$. In block sections 2, 3 and 4 of Figure 7.5 (bottom), the updated running times are obtained on the basis of the modified speed profiles of Figure 7.5 (top). The fixed arc lengths of the delayed train increase to $p' = p + w$, where $w > 0$ is the difference between the running time at $v_{approach}$ and the running time at $v_{operating}$. Moreover, the clearing times have to be increased on all the alternative arcs from the delayed train to all the following trains with respect to a given block section. Consequently, the alternative arcs increase to $a' = a + z$, where $z > 0$ is the difference between the clearing times at $v_{approach}$ and the clearing times at $v_{operating}$. Note that the values of p , a , w and z are not constants since they depend on the corresponding train and block section, i.e., they are related to a node of the graph.

7.4 Train speed coordination procedure

The TSC procedure is in charge of computing a feasible solution for the variable-speed train dispatching problem and basically consists of the procedures shown in Figure 7.6.

After collecting all the necessary data, the alternative graph formulation with fixed routes is used to model the current train scheduling problem that represents the traffic flow of all trains operating in the railway network during the intended time horizon. The traffic flow is therefore computed on the basis of the off-line and real-time information available at the starting time of traffic prediction t_0 . Besides, since no dispatching measure has been taken yet, the resulting alternative graph has a starting selection S^I possibly empty, i.e., all the alternative pairs are still unselected. A CDRFR algorithm (see Chapter 5) is then applied to compute a complete graph selection. If the scheduling solution found is feasible and presents an acceptable speed profile for each train, the iterative procedure stops. Otherwise, a new train scheduling problem is defined and solved again according to a single or iterative

Procedure FeasibilityCheck**Begin** $t_{best} = +\infty,$ **For all** the block sections **do****Begin**Calculate t_β and $t_{update},$ **If** $t_\beta \leq 0$ and $t_{update} \leq t_{best}$ **then** $t_{best} = t_{update},$ **End****Return** $t_{best}.$ **End**

Procedure SpeedUpdating**Input:** T_y and $t_{current},$ **Begin**Find the signal position showing the yellow aspect at $t_{current},$ Find the block section in which the braking of T_y must start,

Increase the length of fixed and alternative arcs according to the braking curve,

Increase the length of fixed and alternative arcs to include re-acceleration.

End

Procedure GraphUpdating**Input:** $S, t_{current}$ and $(k, j),$ **Begin**SpeedUpdating($T_y, t_{current},$ **For all** the alternative pairs **do****Begin**Calculate t_{update} of the pair,**If** $t_{update} \geq t_{current}$ **then** deselect the pair,**End**Select the alternative arc (k, j) and increase (if necessary) the modified scheduled times,**Return** $S.$ **End**

Procedure TrainSpeedCoordination**Input:** $S = S^I$ and $t_{current} = t_0,$ **Begin****While** (1) **do****Begin****If** iterative scheduling **then** $S =$ Solution (re)computed by a given CDRFR algorithm,**If** S is not a complete and consistent selection **then exit** $t_{current} =$ FeasibilityCheck,Let (k, j) be the alternative arc of the first overlap,**If** $t_{current} = +\infty$ **then exit else** $S =$ GraphUpdating($S, t_{current}, (k, j)$).**End****End**

Figure 7.6: Sketch of the train speed coordination procedure

scheduling strategy (see the dispatching system architecture described in Chapter 3 and the pseudo-code of the TSC procedure described in Figure 7.6).

Let $t_{current}$ be the current time of the TSC procedure, and at the beginning let us set this time to t_0 . The first train scheduling problem is defined by letting all trains travel at their maximum speed in the absence of conflicts, i.e., free-net running. Precisely, the resulting solutions consider only ordering constraints due to the position and time of each train at its entrance of the dispatching area. After the CDRFR algorithm with fixed-speed profiles has computed a scheduling solution, the feasibility check is performed to verify whether the train dynamics are acceptable or not. If the check fails, then a new train scheduling problem with fixed-speed profiles is generated while the speed profile of the “first” unacceptable train is updated and the new current time $t_{current}$ is set. Moreover, the alternative graph is modified by keeping in the graph selection all the alternative pairs corresponding to the events which occurred “before” the increased current time. Precisely, the new selection is obtained starting from the previous selection S by keeping the already selected pairs as in S and by selecting the other alternative pairs, that are scheduling events “after” $t_{current}$.

Given a virtual overlap, let t_{update} be the time in which the driver of the following train has to start braking according to the signaling system constraints. In other words, t_{update} represents the time instant in which the train starts deviating from the scheduled speed profile due to the driver’s braking action (Figure 7.4). The feasibility checking procedure selects a virtual overlap such that its t_{update} is the minimum with respect to all the unsolved overlaps of blocking times, and the new $t_{current}$ is set to the minimum value of t_{update} . In the graph updating procedure, the alternative pairs for which the minimum t_{update} of the involved trains is greater than the new $t_{current}$ are unselected. In other words, at each iteration the new problem to be solved is smaller than the previous problem. Moreover, since the alternative graph is selected up to time $t_{current}$, consecutive delays happening before $t_{current}$ become unavoidable, and therefore they are removed from the alternative graph. The removal of unavoidable consecutive delays allows the CDRFR algorithm to focus only on those delays that can be avoided by train rescheduling. Clearly, the unavoidable consecutive delays removed during the TSC procedure contribute to the final dispatching solution.

During the iterative procedure, the number of unselected alternative pairs decreases until either a solution with acceptable speed profiles is found, or the CDRFR algorithm is unable to compute a feasible schedule, i.e., the algorithm returns an alternative graph containing a positive length cycle. In the latter case, the dispatcher has to carry out other types of timetable modifications such as changing train platform tracks and inserting new routing options or even canceling running trains. On the other hand, if the final solution is feasible, the distance headways between trains are always respected due to the use of blocking times. Moreover, the timetable produced by the variable-speed model is similar to the original timetable since only train rescheduling operations are used to restore the feasibility.

The above description of the TSC procedure motivates the following theorem:

Theorem 1 *The TSC procedure of Figure 7.6 terminates in a finite number of iterations.*

Proof. Let $(N, F \cup S)$ be a feasible alternative graph solution but an inadmissible TSC solution obtained by the CDRFR algorithm, i.e., this represents a solution with unsolved overlaps of blocking times. During the graph updating procedure a new selection S' is obtained. By construction, S' is a feasible partial selection up to time $t_{current}$. By applying the CDRFR algorithm starting from the partial selection S' , either the algorithm is able to find an extension S'' of the partial selection S' , or the algorithm fails to find a feasible schedule. In the latter case, the iterative procedure stops immediately. Otherwise, since S'' is an extension of S' , S'' will be a deadlock-free and conflict-free selection at least up to time $t_{current}$. Therefore, the first time in which a train starts braking (the new $t_{current}$) is greater or equal to the one calculated in the previous iteration, and S' contains at least one more alternative arc selected (the one fixing the overlap). Since the number of alternative pairs is finite, the TSC procedure executes at most $|A|$ iterations. \square

The overall approach is a heuristic, because at each step of the TSC procedure we solve an overlap and update the speed profile of a train definitively. During the next iterations the procedure takes into account only the train speed adjustments required by the virtual overlaps of blocking times and not yet considered. Once a speed profile adjustment has been performed, this cannot be altered anymore.

The system solution is a feasible schedule with admissible train speed profiles, obtained by minimizing the maximum consecutive delay and ensuring a safety block distance between trains. An acceptable speed profile for each train over the intended time horizon is provided, and a speed coordination between consecutive trains is attained ensuring blocking time separations (as described in Section 7.2). Given such an acceptable solution it may happen that some trains arrive at the platform before the scheduled arrival time, because we assumed that trains run at their maximum speed in a perturbed situation. *Early arrivals* can be avoided by letting trains run at a lower speed and consequently reducing the energy consumption without causing delays. Moreover, a speed regulator can be adopted to optimize the speed profile of each train route with respect to the signaling system constraints.

7.4.1 TSC illustrative example

A small real-world test case is introduced to illustrate how the variable-speed model can be formulated and solved by using alternative graphs. We present the blocking time plot of a disrupted situation, and show a comparison between the solutions with the iterative scheduling strategy found by the first come first served (FCFS) rule and by our branch and bound (BB) algorithm. It has to be observed that for this illustrative example the ROMA dispatching support system required a negligible computation time.

We assess a railway area (block sections from 51 to 70) of the Schiphol bottleneck area (see Appendix A). This consists of a main corridor of around 10 km length and includes the underground station of Schiphol. Figure 7.7 (top) describes the Schiphol infrastructure studied in our test case and all trains running in the network. There are two stations: Hoofddorp, dedicated to local trains, and Schiphol, dedicated to all passenger trains. We consider the

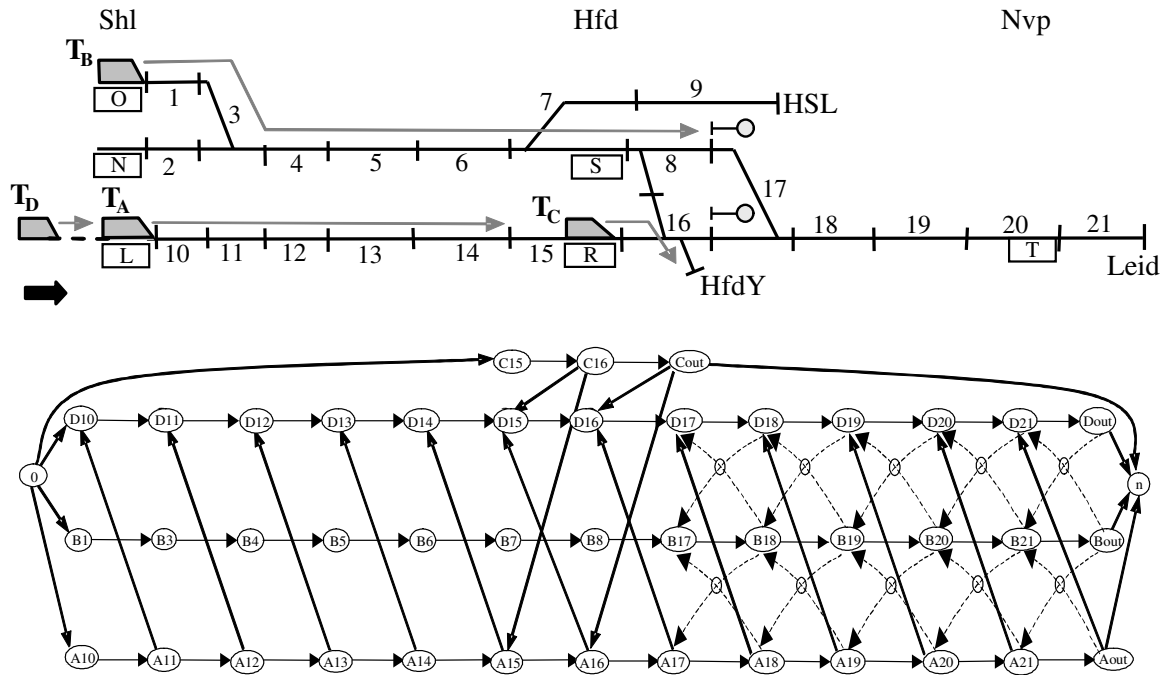


Figure 7.7: A small network and the alternative graph formulation

traffic direction from the three platforms of the Schiphol station to the high speed line, the line connecting Schiphol to Leiden and the depot of the Hoofddorp station. At time $t_0 = 0$, there are four trains in the network. T_A is a passenger train, running from Schiphol (block section 10) to Leiden (block section 21). T_B is a passenger train running from Schiphol (block section 1) to Leiden. T_C is a freight train running towards the Hoofddorp depot (block section 16). Finally, T_D is a passenger train running from Schiphol (block section 10) to Leiden. In the perturbed traffic situation, T_C is considered to be delayed while T_A has already increased its travel time due the speed restrictions at yellow signal aspects when entering block sections 14, 15 and 16 (with maximum allowed speed of 80 km/h, 60 km/h and 40 km/h, respectively). The updated speed profile of T_A causes a potential conflict between T_A and T_B in block section 17 (for which we also report the block signals). When T_D reaches block section 14 the corresponding signal aspect will be green.

Figure 7.7 (bottom) shows the alternative graph for this example. Each node of the graph denotes a pair (train, block section) or a pair (train, exit point), except for dummy nodes. Each alternative pair of arcs is associated with the usage of a common block section. In particular, T_A and T_C share block sections 15 and 16. T_A and T_D share all the block sections since they follow the same path. T_B shares block sections 17-21 with T_A and T_D . Note that due to the initial position of the trains, T_A is not allowed to precede T_C while T_D is not allowed to precede T_A . Therefore, in the graph we have alternative arcs selected a priori among those trains. The respective forbidden alternative arcs are not shown. We represent the entrance times in the network as arcs from the dummy start node 0 to nodes A_{10} , B_1 , C_{15} and D_{10} , with length equal to the timetable scheduled entrance time. Similarly, the timetable scheduled exit time at the terminal block section is modeled with arcs from nodes A_{21} , B_{21} , C_{16} and D_{21} to the dummy finish node n . For reasons of clarity, Figure 7.7

(bottom) does not indicate the fixed and alternative arc lengths.

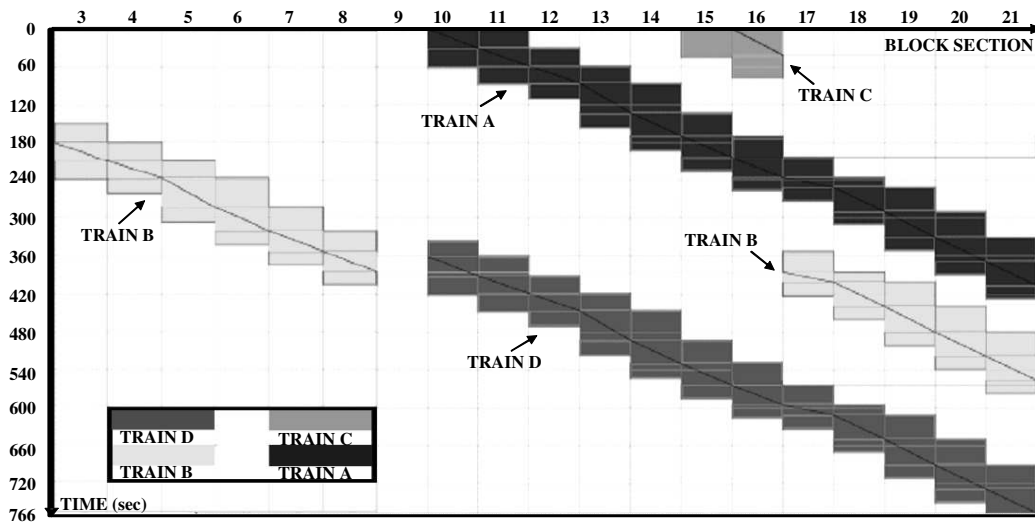


Figure 7.8: Timetable in terms of blocking time plot

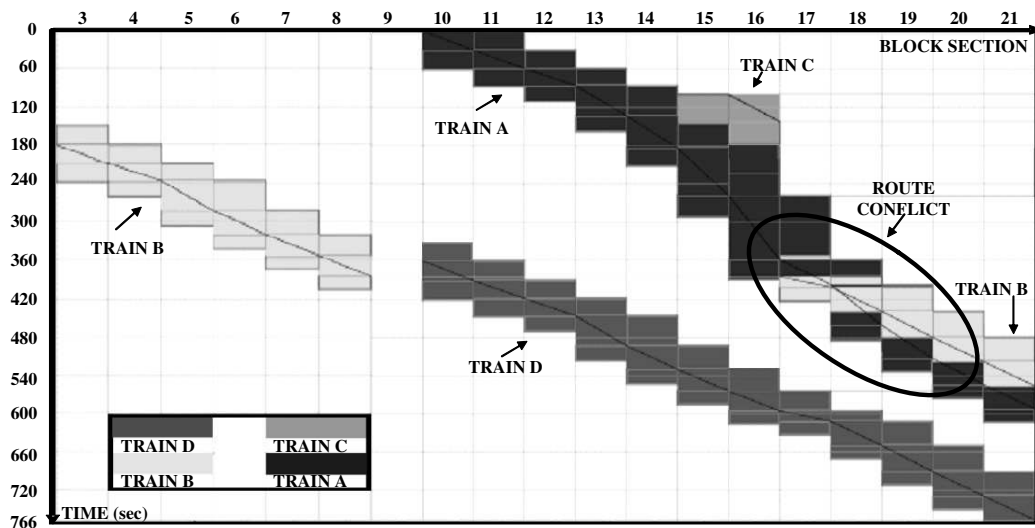


Figure 7.9: Route conflict detection between trains T_A and T_B

Figure 7.8 illustrates the timetable in terms of blocking times for all train routes involved in the example. In Figure 7.9, we then illustrate the perturbed traffic situation. Due to the large entrance delay of T_C , T_A and T_B have a conflict in block section 17 as illustrated by their time-space diagram and by the blocking time plot. In this route conflict situation, a rescheduling of the involved trains has to be considered.

The FCFS rule solves the conflict in block section 17 giving precedence to T_A , since T_A arrives before T_B (Figure 7.10 (right)). Due to the slow speed of T_A , hindering blocking time overlaps arise for T_B in block sections 17, 18, 19 and 20. T_B therefore has to be decelerated in order to solve the overlaps. Updating the speed profiles, the FCFS outcome results in a feasible solution with a maximum consecutive delay for T_B of 176 seconds. T_D

has three overlaps with T_A in block sections 18, 19 and 20, causing an average consecutive delay of 67.75 seconds per train.

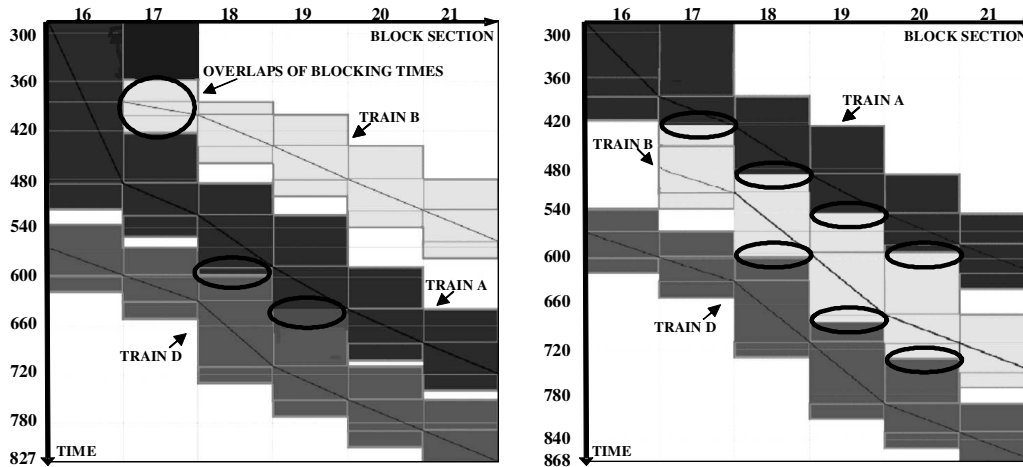


Figure 7.10: BB versus FCFS

The BB algorithm is an exhaustive search which explores all the reordering alternatives and chooses the one minimizing the maximum consecutive delay. As shown in Figure 7.10 (left), BB solves the conflict in block section 17 giving precedence to T_B , even if T_B arrives at the conflict point after T_A . The optimal solution of the fixed-speed model has 65 seconds of maximum consecutive delay. However, there are also two overlaps between T_A and T_D in block sections 18 and 19. After updating the speed profile of T_A and T_D , the resulting variable-speed model solution has a maximum consecutive delay of 127 seconds for T_A and an average consecutive delay of 38.5 seconds per train.

The proposed approach is able to globally detect train conflicts in the considered time horizons and rail network. Consequences of conflicts are considered both during the resolution of the fixed speed model and during the TSC procedure (when speed profiles are updated according to variable signal aspects). Dispatching rules, as FCFS and FLFS, detect and solve conflicts one at a time in the order they appear, whereas the AMCC and BB algorithms solve conflicts using global information. In general, the proposed procedure for the variable-speed model is a heuristic. The solution gap represents the distance between the variable-speed model solution and the best solution found for the fixed-speed model, that is a lower bound to the variable-speed model solution (as described in the computational experiments of Section 7.6). However, we recall that the fixed-speed model does not represent exactly the train dynamics in case of yellow signal aspects while the variable-speed model always computes admissible speed profiles.

We now show that the lower bound to the variable-speed model solution is not tight, i.e., the solution gap may be very large even when the solution found by the variable-speed model is optimal. Let us consider the illustrative example of this subsection. After T_C only three different orders are possible between the other trains: $T_A-T_B-T_D$, $T_A-T_D-T_B$ and $T_B-T_A-T_D$. Specifically, we have the following solutions:

- $T_A-T_B-T_D$: This train order is found by the FCFS algorithm. In this case, the solution found by the TSC procedure with iterative scheduling is 176 seconds of maximum consecutive delay and the critical train is T_B .
- $T_A-T_D-T_B$: This solution is worse than the one provided by FCFS, since the critical train T_B is even more delayed.
- $T_B-T_A-T_D$: The solution found by the TSC procedure with iterative scheduling is 127 seconds and this corresponds to the sequence provided by the BB algorithm.

7.5 Improvement of speed profiles at conflict points

The solution of the variable-speed model may include solved conflicts for which the succeeding train faces yellow and red signal aspects during operations. It follows, in practice, that the concerned train will have to reduce its speed involuntarily. This may cause the following operational disadvantages:

1. Conventional ATP systems (like the Dutch ATB-EG) will force the train driver to decrease the speed of his train, in any case, in order to be able to reach a safe state before the next signal. This is expected to be the limit of the movement authority but its distance is unknown to the onboard unit of the ATP system.
2. Train speed can only be increased again, if a signal upgrade has been recognized by the ATP system. Whereas modern ATP systems (like ATB-NG and ETCS level 2) can do so automatically in a safe manner, the older Dutch system ATB-EG does not protect against passing a red signal at a speed of less than 40 km/h. At the time the conflict has disappeared, the train will have a lower speed than originally scheduled. In order to get back to its traveling speed the driver has to re-accelerate, which costs time in the first place (and therefore causes additional delay) and energy in the second place.

In order to reduce those negative effects, we propose a constructive heuristic algorithm for the computation of the optimal train trajectories, as illustrated in the flow diagram and algorithmic steps of Figure 7.11. This algorithm is based on the idea that the train is slowed down slightly some time before such a possible conflict. This action may give way to re-accelerate the train without being hindered and to reach the conflict area with optimal distance behind the other train causing the conflict. The driver should thus never have to pass a signal aspect that forces a modification of its optimal trajectory.

Of course, the algorithm only has to be started in case that a virtual blocking time overlap (speed decreases due to variable signal aspects) may occur; assuming the train starts from its actual state and enters the corridor on its planned (energy-optimal) path. If that is the case, Step 1 of the algorithm (Figure 7.11) consists of the computation of the fastest possible

trajectory from the start state of the train (position, speed) to its target state (scheduled passing time) at the end of the corridor without regard of the signaling system. For this trajectory, the most critical conflict can be determined (biggest negative difference between passage time of the time-optimal trajectory and signal upgrade to undisturbed operation - green signal aspect). Step 2 consists of determining the so-called target trajectory by shifting the time-optimal trajectory in time in such a way, that the train passes the critical conflict at the time of the signal upgrade. It is assumed that the arrival time of this trajectory is later than the planned arrival time. So, this target trajectory is the only solution with minimal delay which does not have to pass a yellow signal. However, there may still be other solutions with smaller delay at the end of the corridor, which necessitate passing a yellow signal once (Albrecht, 2005b).

The next steps of the algorithm consist of finding the best transition between the expected trajectory of the train (1) and the target trajectory (2). These are computed according to the following criteria:

- The train will be slowed down the least possible, because every slowing down needs to be compensated by re-acceleration which costs energy.
- The smaller the distance and the longer the time to the critical conflict, the more the train has to be slowed down.
- The earlier a train diverges from its original trajectory, the later the train will be at any given position before the conflict. Those delays may cause consecutive delays of subsequent trains.

Those three factors have to be taken into account in the construction phase of the transition trajectory. In order not to hinder subsequent trains that partially use the same infrastructure, Step 3 sets a start state on the original trajectory. The time the train may diverge from its trajectory can be computed by estimating the blocking times recursively (from the time the next train will need to use the infrastructure back to the required sectional release time of the same infrastructure for the examined train).

In Step 4, the target state is fixed at the last point on the target trajectory, which theoretically would allow having the maximal distance between the start and target state. Then, the transition trajectory must be computed (Step 5), containing minimally three regimes and two switching points (Albrecht, 2005b). Here three phases are proposed: A regime of braking or acceleration, cruising at a given cruising speed (i.e., running at a constant speed) and re-acceleration to the target speed. In order to find the switching points between the regimes, the maximal cruising speed between the start and target state is successively reduced. The first and the last phase are prolonged respectively. The arrival time at the target state increases and Step 5 is repeated until the target state is reached on time.

It must then be checked, whether the train can follow this transition trajectory with respect to the signaling system (Step 7) or whether the train has to brake involuntarily when following it. In the latter case, the target state is moved towards the start state (Step 6). This

generally leads to lower cruising speeds of the transition trajectory, where the risk of hurting the signaling constraints decreases. If the target state is moved too much towards the start state, it may be possible that no feasible solution for Step 5 exists and the desired target trajectory can not be reached under the given constraints. Either the constraints have to be redefined (move start state towards section entrance, Step 3) or the target trajectory moved forward in time and the optimization has to be restarted. In case the computed target trajectory arrives too early at the exit of the regarded corridor, the available running-time reserve must be distributed either before or after the conflict. That may be done using Dynamic Programming (Albrecht & Oettich, 2002).

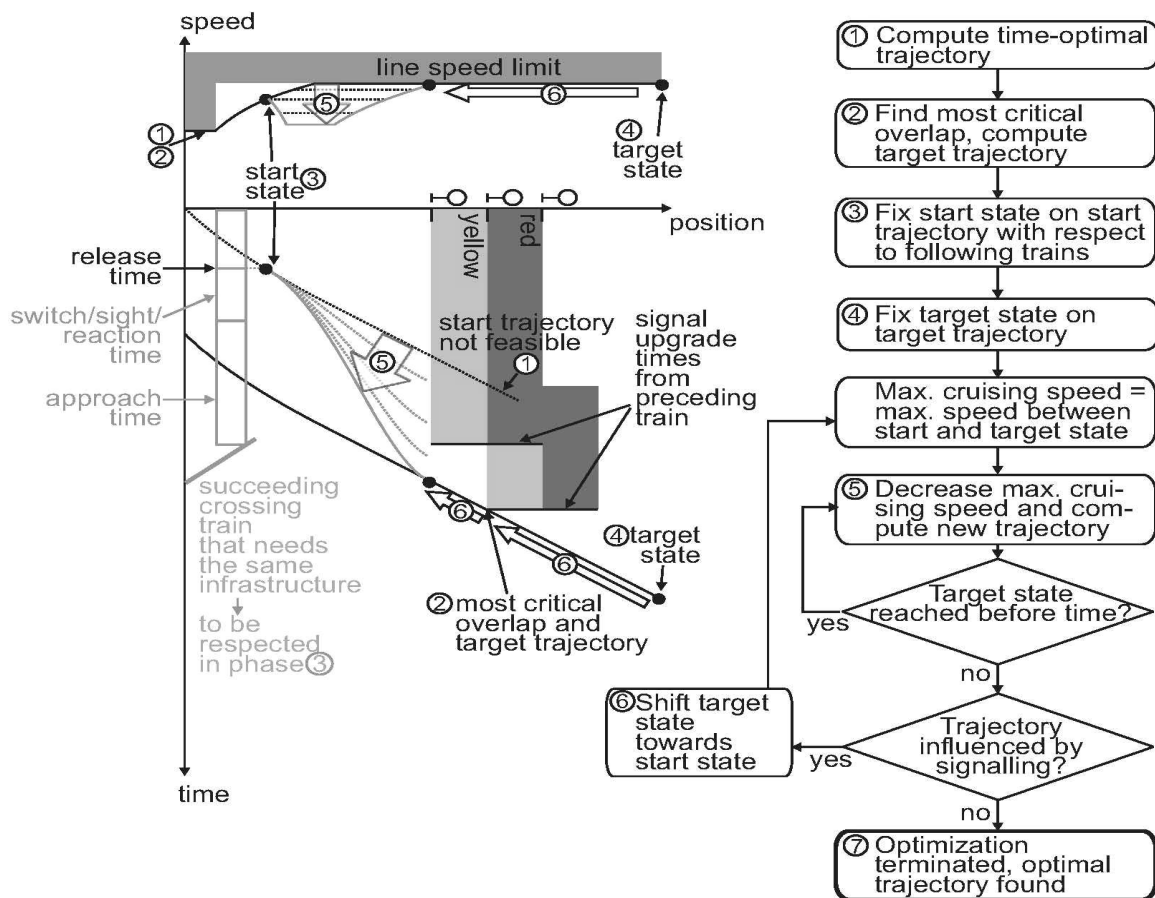


Figure 7.11: Flow diagram and algorithmic steps illustrated in a theoretical example

7.6 Computational experiments

This section proposes computational experiments to test ROMA as a decision support system for train dispatchers including the variable-speed model solutions. We consider only the TSC procedure with the iterative scheduling strategy while the single scheduling strategy will be studied and compared with the former one in Chapter 8. Moreover, dispatching algorithms are executed on a laptop equipped with an Intel Pentium M processor (1.6 GHz),

512 MB Ram and Linux operating system. Computation times and delays are always expressed in seconds.

In this section, the switching time to release or set the interlocking routes and signals on open tracks is taken to be one second, on the basis of automatic signal blocks with electronic technology. Sight and reaction times at sight distance of the approach signal are assumed equal to 12 seconds. The signaling system consists of two speed levels in case of a conflict ahead (80 km/h, 40 km/h), which are signaled depending on the section lengths. The Dutch ATP system ATB-EG requires braking before approaching a block section, where the signaled speed is lower than track maximal speed, which is also considered in the simulation.

7.6.1 Applications of the train speed coordination procedure

This subsection compares different configurations of our train dispatching support system in two Dutch railway dispatching areas and various delay configurations. The actual timetable may therefore be adjusted and a new feasible solution has to be produced within a short computation time. We compare the solutions found by the TSC procedure and adopting simple dispatching rules and advanced scheduling algorithms.

Schiphol dispatching area

We study the whole Schiphol dispatching area between Leiden and Amsterdam, that is around 20 km long and up to 54 trains are scheduled each hour (see Appendix A), to evaluate the ROMA performance even for large and dense scheduling instances. We consider 48 cases of timetable perturbation in order to test our dispatching support system under strong disorder (Table 7.1). We investigate 4 time horizons ranging from 15 minutes to 1 hour. For each time horizon, we generate 12 instances with random (Uniform or Gaussian) entrance delays. In Table 7.1, the first three columns refer to the size of the time horizon expressed in minutes, the number of trains running in the network and the number of alternative pairs ($|A|$) necessary to model the instance. The last three columns report on the perturbations characteristics as the maximum and average entrance delays and the number of delayed trains entering the network in the first half of the time horizon under examination.

Time Horizon	Running Trains	Alternative Pairs	Max Delay	Avg Delay	Delayed Trains
15	14	448	300	67.5	5
30	27	1892	300	67.8	10
45	41	4513	300	66.2	15
60	54	8060	300	68.8	20

Table 7.1: Timetable perturbations

Table 7.2 describes the average results for the four timetables with different time horizons, described in Table 7.1. We refer the reader to D’Ariano and Pranzo (2005) for more information about single instances. The dispatching support system solutions are obtained by

four CDRFR algorithms (i.e., the BB procedure, the AMCC greedy heuristic and the FLFS and FCFS dispatching rules described in Chapter 5) for each test. In Table 7.2, we show the maximum and average total delays for the stations of Schiphol and Hoofddorp and for the dispatching area borders. Since the effects of speed updating are evaluated in terms of the fixed and variable-speed models, the results are thus grouped under the columns “Fixed Speed Model” and “Variable Speed Model” (train speed coordination according to the required safe distance headways). We also report the number of TSC iterations (“Num Iter.”), and the percentage of trains and block sections modified by the variable-speed model, which are indicators of the perturbation severity level. The column “Num Arcs Changed” reports the number of alternative pairs that are selected differently compared to the solution obtained for the fixed-speed model.

Algo Used	Fixed Speed Model		Variable Speed Model						
	Max Delay	Avg Delay	Max Delay	Avg Delay	Comp. Time	Num Iter.	Num Arcs Changed	% Trains Modified	% Sections Modified
15 minutes									
BB	221	27.92	237.25	39.02	0.20	30.42	2.5	10.71	2.81
AMCC	221	27.92	237.25	39.02	0.15	30.42	2.5	10.71	2.81
FLFS	238.42	33.3	252.42	41.55	0.17	31.83	4.83	14.875	5.43
FCFS	246.08	34.48	259.58	41.27	0.16	31.42	7.33	16.66	6.69
30 minutes									
BB	307.5	42.16	388	66.42	4.27	60.58	18.5	25.28	16.48
AMCC	332.5	42.47	365.75	69.44	3.01	60	19.83	26.52	16.96
FLFS	413.25	54.27	493.92	75.74	3.05	61.25	31.25	34.84	23.84
FCFS	419.58	55.11	450.33	75.18	2.88	59.17	21.92	25.9	19.29
45 minutes									
BB	332.42	49.66	486.5	73.92	46.24	100.25	43.33	38.63	28.59
AMCC	344	50.42	501.92	75.74	37.72	101.5	40.17	34.77	25.88
FLFS	514.92	59.41	583.75	79.98	35	97.58	57.42	37.41	35.76
FCFS	564.5	62.62	625.5	83.91	35.06	97.67	40.75	27.65	28.69
60 minutes									
BB	454.41	48.5	779.83	90.35	194.16	145	31.92	24.36	22.38
AMCC	458.08	48.78	668.08	104.08	102.01	144.25	32.83	25.13	23.16
FLFS	694.75	63.56	988.5	100.15	64.42	140.58	91.08	45.79	45.14
FCFS	736.5	67.96	1018.33	105.83	64	139.25	88.16	44.25	38.43

Table 7.2: Dispatching solutions for the Schiphol dispatching area

Next, we discuss the solutions obtained by the four CDRFR algorithms. Among these, the BB procedure provides better quality solutions but requires more computation time. We recall that BB is able to improve train punctuality by adopting the objective function of minimizing the maximum consecutive delay at each step of the TSC procedure with the iterative scheduling strategy. This approach results also in a significant reduction of the average total delays compared to the other algorithms, as shown in Table 7.2. So, BB is the best algorithm and is followed by the AMCC greedy heuristic. However, quite surprisingly, for some solutions of the variable-speed model the AMCC is able to find better solutions in terms of maximum total delays compared to BB. This is probably due to the iterative adjustment of speed profiles at conflict points. When comparing the two proposed dispatching rules, it turns out that the FLFS rule outperforms the FCFS rule.

As shown in column “Comp. Time”, the BB algorithm is able to solve the variable-speed model for the hourly Schiphol timetable within about three minutes, whereas the smaller

instances are solved in a few seconds. Note that there is a rapid increase of computation time between the time horizons of 30 and 45 minutes due to the larger number of alternative pairs in the graph (see Table 7.1). The AMCC heuristic requires around 100 seconds to solve, on average, the variable-speed model for the time horizon of 60 minutes, while the dispatching rules need around one minute of CPU time. For all the test cases, the fixed-speed model solutions are computed, on average, within one second.

Furthermore, Table 7.2 shows the difference between the fixed and variable-speed model solutions (by comparing the respective maximum and average delays), which can be viewed as an indicator of the inaccuracy of the fixed-speed model for the train scheduling problem. In fact, in the fixed-speed model solutions the train dynamics neglect the time loss due to acceleration and braking at conflict points. On the other hand, only the variable-speed solutions, which generate larger delays, represent feasible schedules and acceptable train dynamics. In other words, the fixed-speed model underestimates the consequences of unexpected braking and re-acceleration, while the variable-speed model presents much more realistic solutions. A comparison between the maximum total delays obtained by the BB algorithm, with the fixed and variable-speed models, shows, clearly, that the adjustment of the train speed profiles (due to consecutive delays caused by conflicting train routes) increases when the time horizon for the test cases is enlarged.

When comparing the overall average results of the fixed-speed and variable-speed models, the train orders change from 10% to more than 45% (see column “% Trains Modified”). These modifications are not limited to small portions of the railway network (see column “% Sections Modified”) but involve a large part of the dispatching area (up to 45%).

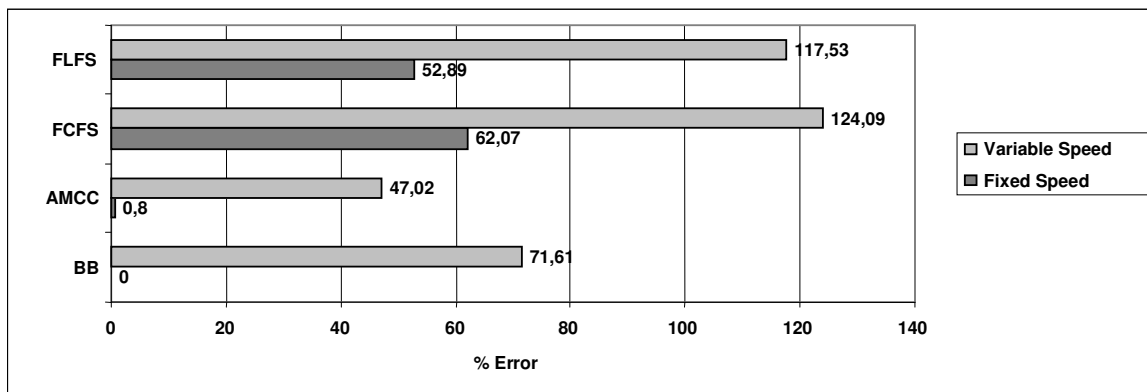


Figure 7.12: Average relative errors from the lower bound on the value of the fixed-speed model solution

Since the TSC procedure may only cause an increase in the running times, the optimal solution of the fixed-speed model (computed by the BB algorithm for the CDRFR problem) is a lower bound (LB) to the value of the solution obtained by the TSC procedure with the iterative scheduling strategy. A relative error can be calculated as $(x - LB)/LB$, where x is the value of the objective function. In Figure 7.12, we compare the relative error of the fixed-speed and variable-speed model solutions computed for the 60 minutes time horizon. We note that the relative errors are fairly large and may be considerably due to the lower

bound, i.e., to the train dynamics adopted by the fixed-speed model. For instance, in the real-world example of Section 7.4.1, the variable-speed model finds the optimal solution but the relative error of the optimal solution from the lower bound is more than 90%.

Figure 7.13 describes the average effects over the 12 test cases with a time horizon of 60 minutes. We consider the solutions obtained by each CDRFR algorithm in terms of delayed trains with a total delay of more than two minutes. For the fixed-speed and variable-speed models, we show the percentage of delayed trains. It can be observed that for all the algorithms there is a substantial difference between the fixed-speed and variable-speed model solutions. This means that updating the train speed profiles causes a large number of delayed trains which would be neglected by applying the fixed-speed model only. Besides, for the variable-speed model the FCFS solutions present, on average, almost two (3.47%) more delayed trains compared to the BB solutions.

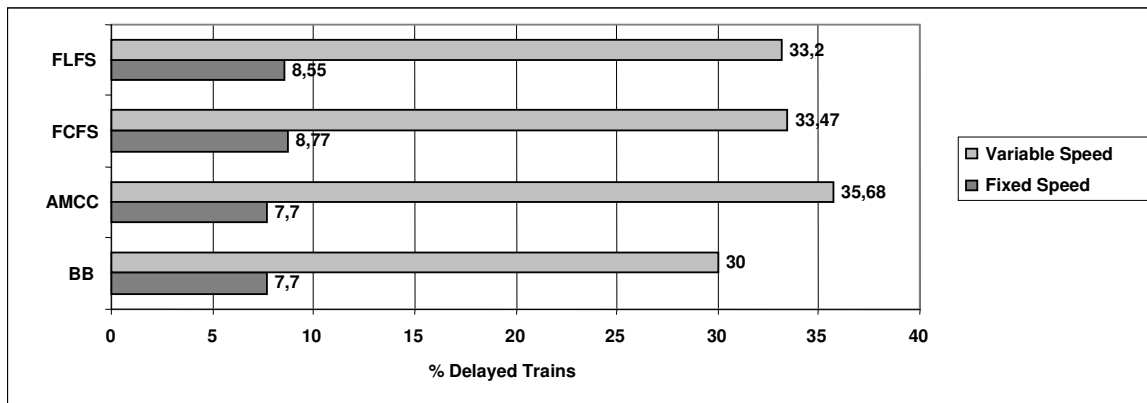


Figure 7.13: Comparison of performance of the four algorithms in terms of delayed trains with a total delay of more than two minutes

The variable-speed model solutions obtained by the BB algorithm for the hourly timetable can also be of interest to detect which are the critical scheduled stops of the dispatching area. In particular, Schiphol station has 35% of the trains with a total output delay of more than two minutes. This percentage is higher than at the other stations (Hoofddorp has 20% and the stations at the borders of the dispatching area have 32%). This information on the critical scheduled stops can be used when considering the application of further dispatching measures, such as train rerouting or dwell time modifications.

These computational experiments on the Schiphol dispatching area show that the proposed variable-speed approach can handle large instances with a high level of detail and a short computation time. The influence of the train speed updates and the resulting delay propagation is evident when comparing the fixed and variable-speed models in terms of maximum and average delays. Furthermore, there is also a large difference of the variable-speed model solutions compared to the lower bound on the value of the fixed-speed model solution. The objective function of minimizing the maximum consecutive delay represents a good trade-off between the minimization of both average and maximum total delays (as shown by the BB solutions). Moreover, the use of advanced scheduling algorithms allows a significant

reduction of train delays and number of delayed trains as compared to practical dispatching actions usually adopted by traffic controllers for the Schiphol railway bottleneck (see Hemelrijk et al. (2003) and Schaafsma and Bartholomeus (2007)).

Utrecht - Den Bosch dispatching area

We present a computational study on the Dutch railway link between Utrecht and Den Bosch, which is around 40 km long and includes the intermediate stations of Houten, Houten Castellum, Culemborg, Geldermalsen and Zaltbommel. We consider 12 cases of disturbed traffic conditions in order to test the ROMA dispatching support system. Each test case is based on the hourly timetable in which 26 trains run in the area around Geldermalsen (i.e., block sections 1 - 140, as indicated in Figure A.3). The resulting alternative graph contains 2752 alternative pairs. The perturbations are delays at the entrance points of the railway network and are generated randomly from a uniform distribution. The maximum entrance delay varies between 200 and 1800 seconds while the average entrance delay of the overall perturbations is around 270 seconds.

Table 7.3 describes the average results on the 12 test cases calculated by four CDRFR algorithms (i.e., the BB procedure and the FCFS, PRULE1 and PRULE2 dispatching rules described in Chapter 5) and by means of “no reordering actions” (PRULE3). The idea is to compare dispatching rules of practical interest for this railway area with the BB procedure developed in this thesis. Column 2 reports the computation time of the TSC procedure, column 3 gives the number of iterations needed to adjust the train speeds, which is an indicator of the perturbation severity. Column 4 presents the number of alternative arcs changed between the fixed-speed and variable-speed model solutions. The last columns report the maximum and average consecutive and total delays, computed at the intermediate stations and at the exit points. We refer the reader to D’Ariano, Hansen, and Hemelrijk (2006) for comments on single instances.

Algo Used	Comp. Time	Num Iter.	Num Arcs Changed	Consecutive Delay		Total Delay	
				Max	Avg	Max	Avg
BB	1.73	24.08	1.5	142.92	11.34	794.92	235.16
FCFS	2.02	29.75	1.67	325.92	17.24	872.67	241.05
PRULE1	2.09	29.92	1.75	340.25	18.31	869.25	242.12
PRULE2	2.24	30.75	10.17	331.75	22.59	871.25	246.40
PRULE3	4.37	36.5	0	644.25	106.92	953.17	330.73

Table 7.3: Variable-speed model solutions for the Utrecht - Den Bosch dispatching area

Among the adopted CDRFR algorithms, BB is the best algorithm in terms of the number of iterations, maximum and average delays, and requires very short computation times (as shown in Table 7.3). The maximum (average) consecutive delay computed by the BB algorithm is around 50% (30%) less than the other CDRFR algorithms. In fact, the BB algorithm is able to reduce delays according to the objective function of minimizing the

maximum consecutive delay at each step of the TSC procedure with the iterative scheduling strategy. This approach is also useful to reduce average consecutive delays considerably.

When comparing the three proposed dispatching rules (FCFS, PRULE1 and PRULE2) in terms of consecutive delays, it turns out that the FCFS dispatching rule performs better than the two heuristics based on priority tables. Moreover, PRULE 1 obtains better results than PRULE 2 in terms of average delays. We also note that the total delays show lower differences between the algorithms due to the occurrence of initial delays.

The CDRFR algorithms should also be compared with the PRULE3 solutions. This rule corresponds to solving each conflict between running trains by maintaining the off-line scheduled train orders. However, if the original sequence between two trains at a conflict point results infeasible, e.g. due to the presence of entrance delays, PRULE3 automatically selects the alternative solution. Table 7.3 shows clearly that the BB algorithm and the three proposed dispatching rules improve PRULE3 strongly. In detail, the BB solutions provide around 8 minutes less maximum consecutive delay than the fixed train order solutions during the one hour period of traffic prediction.

As shown in the columns 2 and 3 of Table 7.3, the ROMA dispatching support system with the BB algorithm is able to compute, on average, a dispatching solution in less than two seconds. The limited number of iterations shows clearly that solving the Utrecht - Den Bosch railway link, under the proposed entrance perturbations, is not a too time consuming task. On the other hand, the experiments on the Schiphol dispatching area report considerably longer computation times (a few minutes) and higher number of TSC iterations (more than one hundred) under similar entrance disturbances.

The dispatching solutions obtained by means of the BB algorithm can also be used to detect which are the critical stops in the railway network. Specifically, we found that at Houten and Culemborg stations around 12% of the trains have a total output delay of more than two minutes. This percentage is higher than at the other stations (4% at Geldermalsen, 5% at Zaltbommel and 8% at Houten Castellum).

Furthermore, the number of selected alternative arcs changed between the first run and the last run of the TSC procedure (column 4 of Table 7.3) gives the effort to reach more accurate and reliable solutions. For the BB solutions, we have that only 0.05% of the selected alternative arcs changes between the first and the last run of the TSC procedure. This means that a large part of the solutions has been unaltered in terms of reordering decisions. For the three practical rules, the changes of reordering decisions are also very few. From this set of experiments, the time reserves result well distributed in the timetable.

These computational experiments analyze small and distributed perturbations while we did not consider the effects of severe disturbances. The CDRFR algorithms perform significantly better than simple hypothetical dispatching rules in terms of maximum and average consecutive delays. We obtained small improvements with respect to the total decrease of delays but the increase of punctuality may be considerable if the dispatching support system is successively applied in consecutive dispatching areas.

7.6.2 Effects of optimal train speed profiles

This subsection illustrates the influence of optimal train speed profiles and the trade-off between train punctuality and energy consumption objectives by means of another real-world example. The Dutch railway network between Utrecht and Geldermalsen is studied (see Figure 7.14). This consists of a main corridor of around 14 km and includes Culemborg station. There are four trains in the network. T_A is a freight train, running from Culemborg (block section 15) to Utrecht (block section 22). T_B and T_C are intercity passenger trains running from Geldermalsen (block section 1) to Utrecht. Finally, T_D is a local passenger train running from Geldermalsen to Dordrecht (block section 8). The track speed limits are 80 km/h for the block sections 1-8, 130 km/h otherwise.

We assume that T_A is delayed and the running of the following trains on the corridor is influenced by the presence of block signaling constraints at the entrance of Culemborg station. Potential conflicts can also be found at the exit of Culemborg station. T_A , T_B and T_C follow the same block sections 15-22. T_D claims block sections 5 and 6 with T_B and T_C . The junction at block section 5 (Figure 7.14 shows the corresponding block signals) is the relevant conflict point when computing the train sequence in the network. Note that due to the initial position of trains, T_C is not allowed to precede T_B while T_A can not be surpassed, and therefore we have a priori order among those trains.

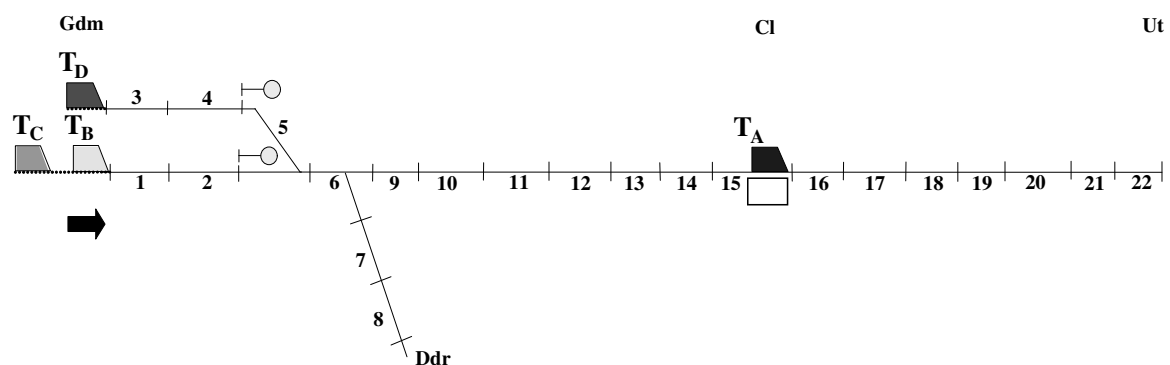


Figure 7.14: The railway link between Utrecht and Geldermalsen

TSC procedure with iterative scheduling

The solution obtained by the TSC procedure, with the iterative scheduling strategy and the BB algorithm, is given in Figure 7.15. All train routes involved in the example are represented in terms of blocking times. Due to the large entrance delay of T_A , retiming decisions have been taken to obtain a feasible schedule with respect to headway and signaling constraints, while the system solution presents the train ordering of the original timetable. In detail, T_B and T_C interfere on block sections 15-17 as illustrated by the virtually overlapping blocking times. In this situation, the planned speed profiles of those trains can no longer be used in the scheduling algorithm and are adapted according to the TSC procedure.

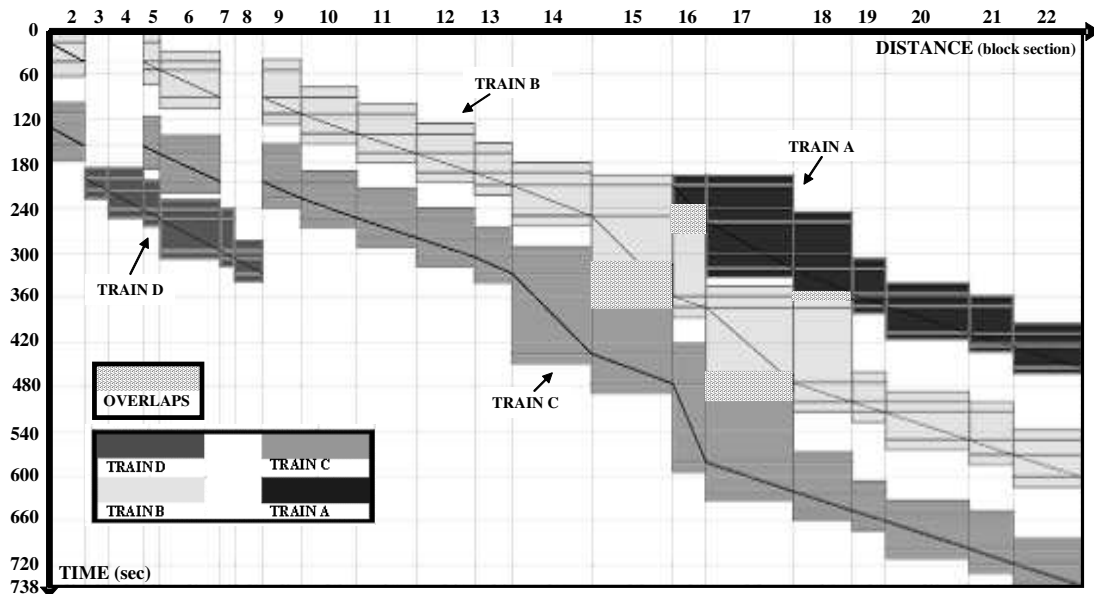


Figure 7.15: Blocking time plot of the dispatching solution

In the proposed example, delayed trains run at their maximum allowed speeds and in case of a yellow (or red) signal aspect an ordinary driver behavior is applied: At sight distance of the approach signal drivers must prepare to start decreasing train speed (see Section 7.2). T_D is not delayed and thus maintains its original speed profile.

Optimal running times at conflict points

For the case of Figure 7.14, three dispatching solutions are discussed in terms of train speed trajectories: The solution given by the TSC procedure with the iterative scheduling strategy (no speed optimization) and two variants of speed optimization (top-down and constrained), as proposed in Figure 7.16. In the three cases, T_A runs at maximum speed while T_B , T_C and T_D adopt different speed profiles.

The TSC procedure with the iterative scheduling strategy presents acceptable speed profiles for each train (see Figure 7.15). At the exit of the network T_B (T_C) has a delay of 128 seconds (167 seconds) while T_D is not delayed, assuming the drivers are supposed to react to the current state of the signaling system only.

The next solution represents a speed optimization that only considers the previous train and no subsequent trains. Precisely, Step 3 of the algorithm described in Section 7.5 is not considered here. The speed trajectories of T_B , T_C and T_D are optimized while the train order at the critical section is fixed. To reduce energy consumption of the individual trains, T_B and T_C are slowed down as soon as they enter the corridor. As a consequence, T_D is delayed. In detail, the energy consumption of all trains is reduced by 45% compared to the previous solution, whereas the sum of delays is about the same (reduced delay for T_B and

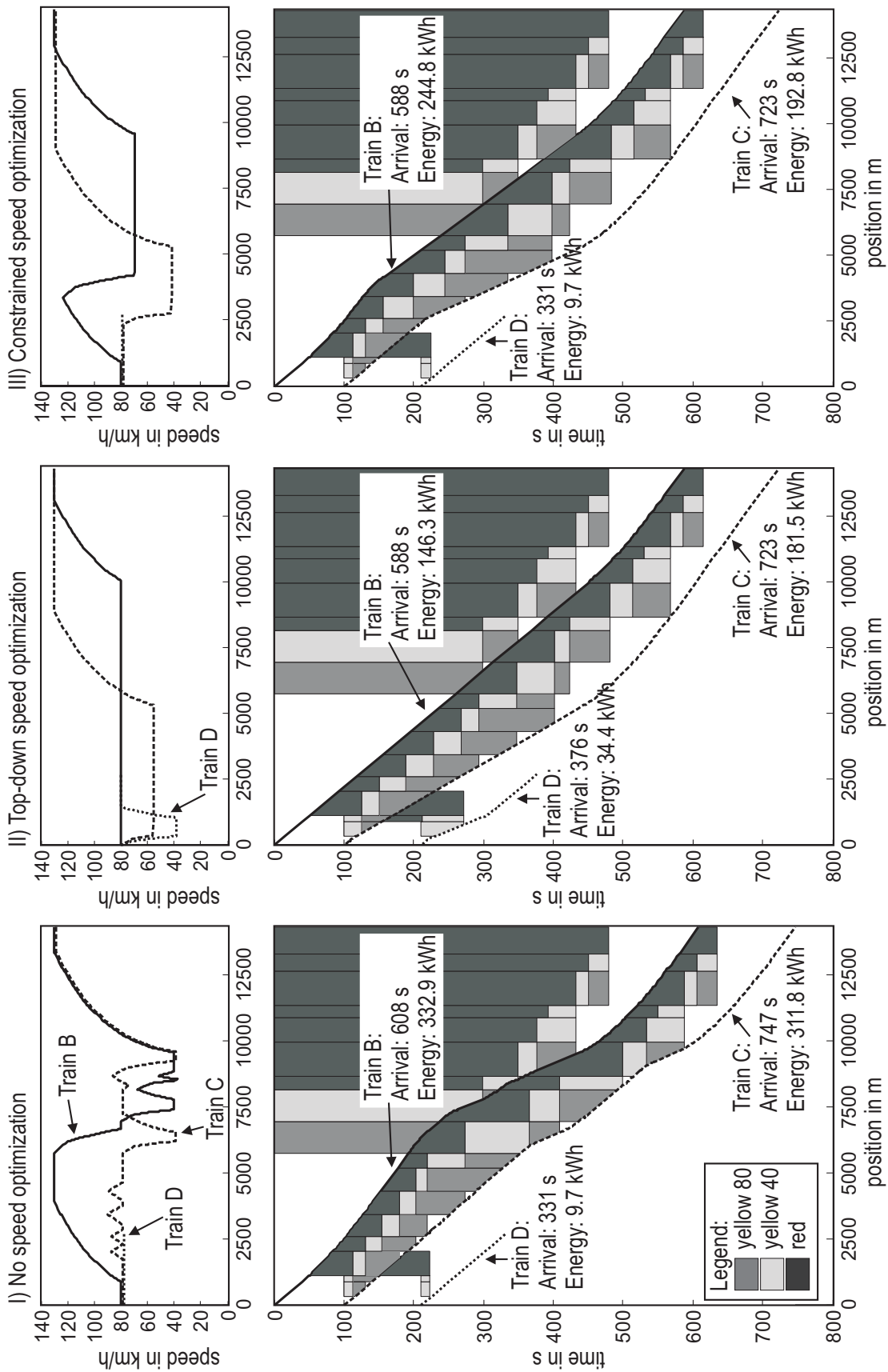


Figure 7.16: Comparison of three variants of train speed trajectories

T_C while increased delay for T_D). This represents a deterioration of the solution quality of T_D compared to the case without speed optimization.

A second speed optimization run is done, this time respecting the constraints given by T_D when fixing the starting states of T_C and T_B in Step 3. The possibility of a speed decrease of T_C is limited to the sections where T_D is not influenced. T_B must be re-optimized in order not to disturb T_C . Moreover, the regarded T_B , T_C and T_D require 30% less energy than without speed optimization. T_D is not disturbed and, therefore, the overall delay can be reduced by 45 seconds. The two proposed speed optimization solutions are Pareto-optimal: Both min. energy consumption and min. delay. The dispatcher would need to decide which of the alternatives fits the authority objectives best.

7.7 Conclusions

This chapter presents a variable-speed model that makes use of different CDRFR algorithms to reduce consecutive delays by pro-actively identifying potential conflicts and rescheduling the train traffic at a network scale and updating the train speed profiles on the basis of the Dutch signaling system NS54. A specific TSC procedure with single or iterative scheduling strategy is proposed. At each step, a train scheduling problem is solved and a feasibility check is performed by applying blocking time theory. If unacceptable speed profiles arise then the train facing yellow and/or red signal aspects is decelerated, the length of its corresponding fixed and alternative arcs in the graph are updated and a train scheduling problem has to be solved. The overall procedure terminates in a finite number of iterations, returns a deadlock-free and conflict-free schedule while maintaining minimum safety distance headways and provides acceptable speed profiles for each train, considering physical constraints during acceleration and braking.

Section 7.6.1 reports our computational results on the Schiphol railway bottleneck and the dispatching area between Utrecht and Den Bosch. The results demonstrate a considerable difference in the schedules obtained by the fixed-speed and variable-speed models. An extensive comparison between advanced scheduling algorithms and simple dispatching rules proves the efficiency of the dispatching support system developed. These findings highlight the need of considering tailor-made models for a dynamic computation of train speed profiles, and implementing advanced algorithms to improve the effectiveness of traffic control.

Section 7.6.2 describes how the speed optimization can be adopted in order to pro-actively react when signal aspects change. A case study on the Utrecht - Geldermalsen railway link is used to quantify the effects of a constructive algorithm that is able to optimize the speed trajectory of each train involved in a conflict situation, even without changing the train orders and routes computed by the dispatching system. This example underlines the need of implementing additional on-board driver information and support systems in order to minimize train delays and energy consumption.

Chapter 8

Short-term traffic prediction

In busy railway networks, conflicts and resulting train delays propagate considerably in time and space during operations. In order to accurately minimize the propagation of train delays, train traffic flow predictions could be extended up to several hours (Short-Term Traffic Prediction (STTP)). On the other hand, as the magnitude of the time horizon increases too much the problem becomes computationally intractable and hard to tackle.

This chapter presents further extensions of the ROMA dispatching support system in order to cope with the STTP problem and strong timetable disturbances. The objective is to proactively evaluate the effects of train rescheduling actions for a time period of some hours within the same day. If the effects of standard delay management are deemed inadequate by the traffic controller, other drastic measures will be adopted such as the use of emergency timetables or the cancelation of train routes.

In this research context, Rodriguez (2000) has shown that a temporal decomposition and constraint programming approach can be used to compute suitable solutions in considerably less computation time compared to the global resolution of the problem. We also decompose the temporal horizon of traffic prediction into tractable time intervals to be solved in cascade. The coordination of each time interval is achieved by inserting the position and the speed of each train at the end of the time interval as an input constraint for the subsequent time interval and by also considering the constraints of re-using the same rolling stock for different train trips. The independent resolution of each hour of dispatching permits handling large time horizons within a linear increase of computation time. We compare this approach with a global formulation of the STTP problem in order to evaluate the error due to the temporal decomposition.

Extensive computational tests are carried out on the Utrecht - Den Bosch railway network in order to simulate the control area of a human dispatcher. We study the short-term consequences of train delays and temporary unavailability of some tracks. The effects of increasing timetable disturbances are analyzed carefully while considering several timetable hours. Furthermore, randomly generated blocked tracks are included in order to obstruct a main traffic direction and to cause a serious propagation of train delays. The disturbance propagation is measured in terms of punctuality for each successive timetable hour.

The chapter is organized as follows. While the global approach has to solve a very big alternative graph, Section 8.1 introduces another resolution procedure that consists of decomposing the time horizon of traffic prediction in intervals and adopting an alternative graph for solving each interval. Section 8.1.1 then gives an explanatory example of the global and temporal decomposition procedure, including a graphical representation of the alternative approaches. Section 8.2 presents the results of our computational experiments based on the Utrecht - Den Bosch dispatching area.

8.1 Temporal decomposition approach

The STTP problem is a large-scale decision problem with several decision variables and constraints. A common operations research practice to solve such a large decision problem is to decompose it into a series of smaller problems that progressively arrive at the final solution providing a good solution to the entire problem (see, e.g. Ovacik and Uzsoy (1997), Ahuja et al. (1993) and Pardalos and Resende (2002)). Specifically, decomposition approaches have been proposed successfully to solve complex job shop scheduling problems, such as machine decomposition (see, e.g. Adams et al. (1988)) or temporal decomposition (see, e.g. Chambers et al. (1991) and Zeng et al. (1998)).

This section describes our temporal decomposition procedure and the traffic optimization model extensions necessary in order to tackle the STTP problem. We first generate an alternative graph $\mathcal{G} = (N, F, A)$ representing all the trains running during the entire time horizon. We then divide the time horizon in η time intervals. Clearly, when dealing with a single time interval ($\eta = 1$), no decomposition is performed and the approach corresponds to a global conflict resolution. On the other hand, for $\eta > 1$ we adopt a temporal decomposition approach. In what follows, we explain the two approaches with reference to the algorithmic description of Figure 8.1.

The global conflict resolution approach directly solves $\mathcal{G} = (N, F, A)$ by adopting the TSC procedure with single or iterative scheduling strategies (see Chapter 7). In this case ($\eta = 1$), ν is the length of the entire time horizon and a solution in the time interval $[0, \nu]$ is denoted as $\mathcal{G}(S)$, where S is a complete consistent selection for $\mathcal{G} = (N, F, A)$.

When the traffic prediction is enlarged up to several hours, the alternative graph $\mathcal{G} = (N, F, A)$ becomes difficult to solve due to the huge problem size. The large instance is thus divided into η tractable alternative graphs to be solved successively. In this case, we suppose that each graph is extended over a time interval of length ν . The temporal length of the entire time horizon of traffic prediction is therefore $\eta\nu$.

The temporal decomposition approach may require the splitting of some train routes over two time intervals. The resolution of successive time intervals thus requires the satisfaction of coordination constraints for the split train routes (i.e., the computation of feasible positions and speeds for those train routes). For example, if a train runs at maximum speed in a given time interval and has to brake at the entrance of the successive time interval there

Procedure TemporalDecomposition**Input:**

A temporal length of each time interval ν ,

A $\eta\nu$ -time horizon instance $\mathcal{G}(N, F, A)$,

begin

Generate the first time interval instance $\mathcal{G}^1(S^\emptyset)$,

Compute a solution $\mathcal{G}^1(S)$ using the TSC procedure with single or iterative scheduling,

if $\eta > 1$ then**begin**

For all i from $i = 2$ up to $i = \eta$ **do**

begin

Generate the i -th time interval instance $\mathcal{G}^i(S^\emptyset)$,

SetDecompositionConstraints($\mathcal{G}^{i-1}(S), \mathcal{G}^i(S^\emptyset)$),

Compute a solution $\mathcal{G}^i(S)$ using the TSC procedure with single or iterative scheduling.

end**end****end****Procedure SetDecompositionConstraints****Input:**

The $(i - 1)$ -th time interval (solved) instance $\mathcal{G}^{i-1}(S)$,

The i -th time interval (unsolved) instance $\mathcal{G}^i(S^\emptyset)$,

begin

Let $Z_i = \emptyset$ be the list of trains running in both $[(i - 2)\nu, (i - 1)\nu]$ and $[(i - 1)\nu, i\nu]$,

Initialize the set of alternative arcs $X_i = \emptyset$,

For all running trains in $\mathcal{G}^{i-1}(S)$ **do**

begin

if the current train T_y is also in $\mathcal{G}^i(S)$ **then**

begin

Insert T_y in Z_i ,

Find the starting time t_{T_y} of the first operation of T_y in the i -th time interval,

Starting from t_{T_y} , calculate the minimum time $t_{T_y}^*$ at which T_y can stop,

Insert in X_i all the alternative pairs of T_y in $\mathcal{G}^i(S^\emptyset)$ until $t_{T_y}^*$,

end**end**

forall alternative pairs of X_i **do**

begin

if one arc (h, k) of the current pair is in the selection of $\mathcal{G}^{i-1}(S)$ **then**

begin

Select the arc (h, k) also in $\mathcal{G}^i(S')$,

end

else select the current pair in $\mathcal{G}^i(S')$ giving precedence to the trains in Z_i .

end**end**

Figure 8.1: Sketch of the temporal decomposition procedure

may be problems due to minimum required space headway and time for braking. It follows that the resolution of successive time intervals must be coordinated in order to avoid infeasibilities.

We solve the problem of coordinating η time intervals, enforcing some precedence relation constraints between consecutive time intervals, i.e., the coordination constraints. Given an i -th time interval, we use the solutions obtained for the previous time intervals to generate the graph $\mathcal{G}^i(S^\theta)$. Moreover, we set the release time of each train in $\mathcal{G}^i(S^\theta)$ according to the rolling stock re-use constraints. The “SetDecompositionConstraints” procedure of Figure 8.1 then computes the required timing constraints between $\mathcal{G}^{i-1}(S)$ and $\mathcal{G}^i(S^\theta)$ as follows. For each train T_y spanning the two time intervals $[(i-2)\nu, (i-1)\nu]$ and $[(i-1)\nu, i\nu]$, we have to find the starting time t_{T_y} of its first operation in the i -th time interval. Starting from t_{T_y} , we calculate the minimum time $t_{T_y}^*$ at which T_y can stop its run. Specifically, $t_{T_y}^*$ is computed starting from the scheduled speed and position of T_y at t_{T_y} while respecting the signaling system, infrastructure and rolling stock characteristics (see Chapter 7 for a description of the adopted driver behaviors).

Before the resolution of the i -th time horizon, we set the precedence relations between T_y and the other running trains until $t_{T_y}^*$. A subset of alternative pairs X_i is thus created to insert the timing constraints in $\mathcal{G}^i(S^\theta)$. Specifically, all alternative pairs of X_i between two trains running in $[(i-2)\nu, (i-1)\nu]$ are selected in $\mathcal{G}^i(S^\theta)$ as in the selection of $\mathcal{G}^{i-1}(S)$. The remaining unselected alternative pairs of X_i are selected in $\mathcal{G}^i(S^\theta)$ giving precedence to the trains running in both time intervals. Finally, the resulting partially selected graph $\mathcal{G}^i(S')$ is solved. Successively, the alternative graph of the next time interval is generated and a new run of the “SetDecompositionConstraints” procedure is performed. This iterative procedure terminates when the η -th time interval is solved. In general, we define the set of all alternative pairs constrained between consecutive time intervals as $X = \sum X_i$ with $i = 2, \dots, \eta$. This decomposition procedure enables our dispatching support system to compute locally feasible schedules corresponding to a globally feasible solution.

8.1.1 STTP illustrative example

Consider the small railway network reported in the upper part of Figure 8.2, in which we have four running trains (T_A , T_B , T_C and T_D), and a time horizon of length 2ν . For these four trains we assume that T_A completes its route within the time interval $[0, \nu]$, while T_B spans the time interval $[0, 2\nu]$. The other two trains, T_C and T_D , travel in $[\nu, 2\nu]$. Furthermore, T_A and T_C are scheduled to be coupled at a platform track station outside the railway area considered. In other words, T_A runs completely over its network route and, after a *turning buffer time* (i.e., the minimum turning-around time plus some time reserves), its rolling stock is re-used by T_C . For simplicity, in this example we do not specify the running and setup times of the involved trains, and we only show the location of the most relevant block signals.

The alternative graph shown in the lower part of Figure 8.2 represents the global formulation approach, i.e., all the running trains are modeled in the graph. We denote a node with the

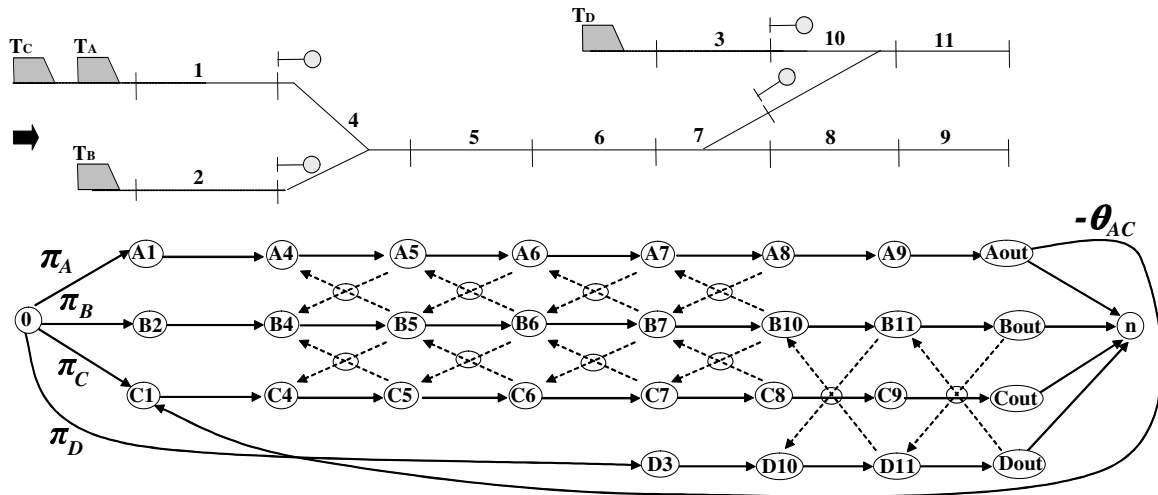


Figure 8.2: Alternative graph formulation of a small network with four trains

pair (train, block section) of the associated operation or with the pair (train, exit point), except for dummy nodes. The four fixed arcs departing from node 0 model the release time of each train (π_A , π_B , π_C and π_D), whereas the arcs entering node n model the objective function. In general, a train T_y arriving in the dummy finish node n at a time greater than the length of the corresponding arc entering node n will be late. The fixed arc connecting T_A with T_C is used to represent the rolling stock constraint between these two trains and the corresponding turning buffer time has length θ_{AC} . There are ten alternative pairs, which are depicted by dashed arcs.

We now show how the small example of Figure 8.2 can be temporally decomposed into two instances. Let ν be the length of the time interval of each instance. In the alternative graph of Figure 8.3 (a), we show the alternative graph formulation of $[0, \nu]$. Two trains, T_A and T_B , and four alternative pairs are depicted in this graph.

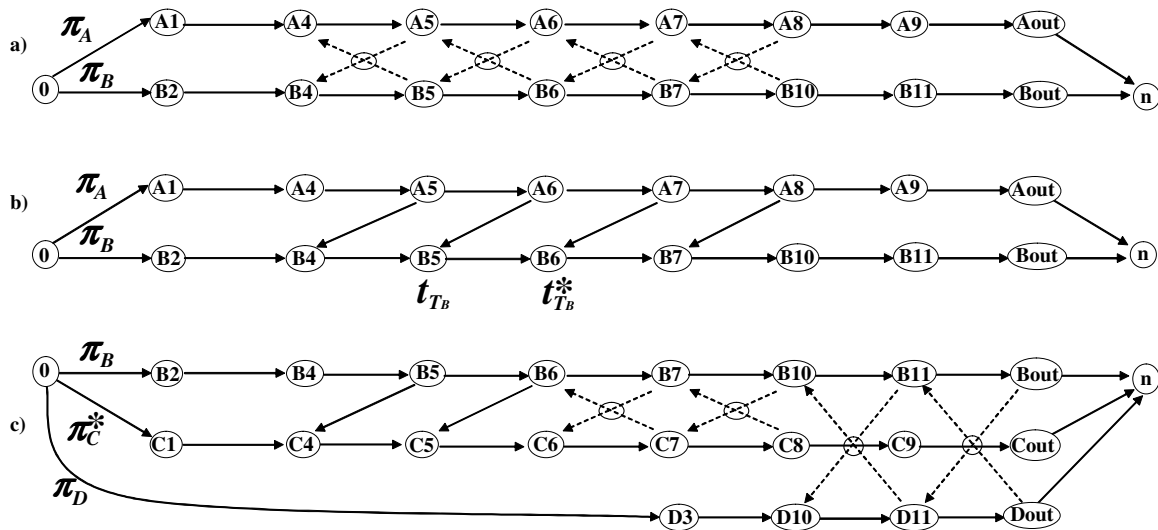


Figure 8.3: Decomposition phases: a) formulation of time interval $[0, \nu]$; b) solution to time interval $[0, \nu]$; c) formulation of time interval $[\nu, 2\nu]$

A dispatching solution to $[0, \nu]$ leads to the graph shown in Figure 8.3 (b), in which T_A precedes T_B on the block sections 4, 5, 6 and 7. Specifically, only the selected alternative arcs are depicted in Figure 8.3 (b). Let t_{T_B} be the time at which the first operation of T_B starts in $[\nu, 2\nu]$, corresponding to the entrance of block section 5, and let $t_{T_B}^*$ be the minimum time required to stop T_B in $[\nu, 2\nu]$. In order to solve $[\nu, 2\nu]$ after the resolution of $[0, \nu]$, T_B must not change its speed profile at least until $t_{T_B}^*$.

In Figure 8.3 (c), we depict the alternative graph of $[\nu, 2\nu]$. Since T_B has coordination constraints with T_C until $t_{T_B}^*$, this must be the first train passing on block sections 4 and 5, corresponding to the selection of two alternative arcs in the graph of $[\nu, 2\nu]$. The set X therefore contains the two corresponding alternative pairs. To compute a complete selection for this graph, four alternative pairs must be selected. In this way, we ensure that the solution to $[\nu, 2\nu]$ is compatible with the solution to $[0, \nu]$. In order to include the rolling stock constraint between T_A and T_C , we also have to model a new release time π_C^* , which must include the original release time π_C plus the delay of the rolling stock used by T_A , if positive. The solution obtained by combining the local solution to the three graphs of Figure 8.3 may be sub-optimal, even if every graph is solved to optimality. This is clearly due to the presence of the coordination constraints between consecutive time intervals, which are needed to avoid schedule infeasibilities (i.e., wrong train trajectories). But even if these constraints are added to the global formulation approach, the solution obtained can be better than the decomposed one.

8.2 Computational experiments

This section presents our experiments on a large sample of practical size instances. We consider the Dutch railway link between Utrecht and Den Bosch and the complex area around Den Bosch station with eight additional platform stops. The whole dispatching area is around 50 km long (see Appendix A) and up to 40 trains are scheduled each hour. We study this network by simulating disturbed traffic conditions, i.e., entrance delays, dwell time perturbations and blocked tracks. We also take into account additional railway constraints such as a minimum transfer time between connected train services (as described in Appendix A).

For STTP purposes, we have to add other railway constraints due to the re-use of rolling stock when trains leave and enter the dispatching area. Intercity trains generally have long turning buffer times, and small delays are very likely to be completely absorbed. For local services, the situation is slightly different. Outside the studied dispatching area, some trains run along the corridor with long dwell times at major railway stations (up to 10 minutes), so we assume that delays will fade out. For other trains having larger delays or shorter dwell times, delays may survive the remaining part of their round-trip. In this case, we model rolling stock re-use constraints with the following assumptions: Trains with up to two hours round-trip have a maximum turning buffer time of twelve minutes, while trains with larger round-trip times have one of thirty minutes.

We next test the ROMA dispatching support system under severe disturbances. Effects of timetable disruptions and entrance perturbations are studied while considering a time horizon of several timetable hours. Randomly generated disruptions are included in each hour of traffic prediction while a set of trains running in the first timetable hour is delayed on the basis of entrance perturbations chosen in a time window of typical train delays. Output delays of each hour cause extra entrance delays since the rolling stock is re-used on a cyclic basis and delays cannot be completely absorbed. Moreover, infrastructure disruptions obstruct main traffic directions and cause a serious propagation of train delays. In fact, when a double track corridor is blocked in one of the two directions trains traveling in opposite directions must share the only track available.

Table 8.1 describes the disturbances introduced to evaluate the new ROMA features. Column 1 indicates the disturbance category while column 2 presents the values of maximum entrance delay (in seconds) and column 3 shows the number of trains with an entrance delay larger than zero. For those trains the entrance delay is chosen randomly according to a uniform distribution and up to a given maximum value. Three instances are generated for each value of columns 2 and 3, yielding a total of 27 entrance delay configurations. The timetable disruptions are then introduced in the remaining columns. Column 4 gives the number of trains rerouted by the disruption recovery procedure (as described in Chapter 3). Column 5 shows the number of blocked tracks in the network. Specifically, the case of one blocked track corresponds to the unavailability of block section 83, while the other disruption (the large disturbance) corresponds to the unavailability of block sections: 168, 164, 67 and 175. The last column shows the dwell time perturbations (in seconds) which are applied to all the rerouted trains at their perturbed station stops. For each couple (blocked tracks, dwell time perturbation), we generate a disrupted timetable and for each of these timetables we test the 27 perturbations. In total, there are 81 disturbances, which are divided in 45 small, 27 medium and 9 large.

Disturbance Category	Entrance Perturbations		Timetable Disruptions		
	Max Delay	Delayed Trains	Rerouted Trains	Blocked Tracks	Perturbed Dwell Time
Small	300	5	0	0	0
Medium	900	10	5	1	30
Large	1500	20	10	4	60

Table 8.1: Timetable disturbances description

In the next two subsections, we present and discuss our computational results for the disturbances described in Table 8.1. We first evaluate the best ROMA configuration over a subset of large disturbances, and then provide a detailed description of the solutions obtained using the best configuration. Computation times and train delays are always expressed in seconds. Dispatching algorithms are executed on a laptop equipped with an Intel Pentium M processor (1.6 GHz), 512 MB Ram and Linux operating system. Each run of a CDRFR algorithm is truncated after 30 seconds of computation time.

8.2.1 Configurations analysis

The purpose of real-time train dispatching imposes strict time limits to produce a new feasible timetable, limiting the execution of the real-time optimization procedure. A reduction of the computation time of the scheduling algorithm can be attained by decomposing the time horizon into reasonable time intervals, at the cost of less accurate schedules. In this subsection, we study these aspects for three large disturbances of Section 8.2.

Table 8.2 presents a detailed numerical comparison between four ROMA configurations, varying the scheduling algorithm (i.e., BB or FCFS) and the scheduling strategy (i.e., single or iterative). The results are shown in terms of consecutive delays, since initial delays are considered unavoidable. Each row of the table presents average results computed for an increasing length of the time horizon. Specifically, column 1 indicates the time horizon in terms of timetable hours. Column 2 reports on the solutions computed using one of the two scheduling strategies. Columns 3–6 and 7–10 show the results with BB and FCFS, respectively. Columns 3 and 7 give the maximum consecutive delays. Columns 4 and 8 show the average consecutive delays. Columns 5 and 9 indicate the average computation time of the four ROMA configurations over the 3 disturbances. Finally, columns 6 and 10 give the average number of speed profile adjustments due to trains facing a yellow or red signal aspect (i.e., the runs of the speed updating procedure). Being a simple counter, this indicator may be very sensitive to the number of running trains.

Time Horiz	Scheduling Strategy	Algorithm BB				Algorithm FCFS			
		Max Delay	Avg Delay	Comp Time	Num Iter	Max Delay	Avg Delay	Comp Time	Num Iter
1	Single	336	25.76	6	49	440	24.25	6	51
2	Single	384	24.81	151	114	480	37.21	122	113
3	Single	384	21.09	2057	149	480	35.63	2514	154
1	Iterative	305	25.97	14	50	336	25.59	14	53
2	Iterative	339	23.27	690	107	423	23.50	693	114
3	Iterative	350	21.05	16740	142	423	30.01	19870	156

Table 8.2: Computational results with the global formulation approach and four system configurations

From this set of experiments, the combination of BB and iterative scheduling strategy is the best configuration in terms of maximum consecutive delay minimization and provides good results in terms of average consecutive delays. However, the main difference between the four configurations is that the computation time increases considerably when the time horizon of traffic prediction is enlarged. Feasible solutions within computation times of up to three minutes can be achieved for the evaluation of the single scheduling strategies over a time horizon of two hours, while the iterative scheduling strategies may need about twelve minutes of computation time. This effect does not depend on the number of speed profile adjustments but is directly related to the scheduling process and instance size. When dealing with real-time application purposes, the global approach formulation is only suitable for time horizons of up to two hours for this railway network.

Table 8.3 presents a comparison of the two resolution approaches of Section 8.1 (i.e., the global and temporal decomposition) when varying the time horizon of the traffic prediction. For both approaches, we use the best configuration of Table 8.2 (i.e., BB and iterative scheduling strategy). For Table 8.3, column 1 refers to the time horizon length, expressed in timetable hours. Column 2 reports on the solutions computed using the global approach (i.e., the first three rows) and temporal decomposition approach (i.e., the successive rows). Columns 3 and 4 show the maximum and average consecutive delays, respectively. Column 5 indicates the average computation time and column 6 gives the average number of speed profile adjustments. Column 7 presents the average number of trains running in the corresponding time horizon. We note that each hour of timetable has 40 running trains and the instances of the global approach increase by this number of trains for each hour of traffic prediction. However, the instances of the temporal decomposition approach present a larger number of trains. This is due to the fact that some delayed train routes are split into two consecutive hours of timetable, e.g. the time horizon of nine hours contains 25 split train routes. The average number of alternative pairs is indicated in column 8, while the last column shows the number of alternative pairs “fixed” when using the temporal decomposition approach.

Time Horiz	Resolution Approach	Max Delay	Avg Delay	Comp Time	Num Iter	Running Trains	$ A $	$ X $
1	Global	305	25.97	14	50	40	4133	-
2	Global	339	23.27	690	107	80	17734	-
3	Global	350	21.05	16740	142	120	40803	-
1	Decomp	305	25.97	14	50	40	4133	-
2	Decomp	443	26.29	503	175	92	12889	2858
3	Decomp	443	21.29	548	233	134	18119	3906
4	Decomp	443	19.84	597	305	177	23368	4822
5	Decomp	443	16.42	637	359	218	28506	5812
6	Decomp	443	17.59	677	422	260	33656	6677
7	Decomp	443	16.78	714	474	302	38604	7481
8	Decomp	443	16.16	750	527	343	43659	8350
9	Decomp	443	14.59	787	579	385	48606	9154

Table 8.3: Global versus decomposition

The global approach gives better results in terms of delay minimization, reducing on average the maximum consecutive delay by 21% compared to the temporal decomposition approach. The temporal decomposition approach results sub-optimal because a partial selection of alternative arcs is fixed (i.e., $|X|$) for the coordination of successive time intervals. On the other hand, the temporal decomposition is the only viable approach for computing traffic predictions over a time horizon of more than three hours. The complexity of large time horizons is directly due to the number of running trains and alternative pairs (i.e., $|A|$), while the number of speed profile adjustments is not a critical time consuming factor.

This first set of experiments allowed us to determine the best ROMA configuration. Moreover, the decomposition approach offers similar results to the global approach while making

the code more compatible with short-term predictions of railway operations.

8.2.2 Effects of timetable disturbances

The ROMA performance is here evaluated using the best configuration of the previous subsection and the temporal decomposition approach. We study the overall set of disturbances of Section 8.2 when varying the time horizon of traffic prediction. The average results are reported for enlarging the time horizon length hour by hour, up to 9 timetable hours.

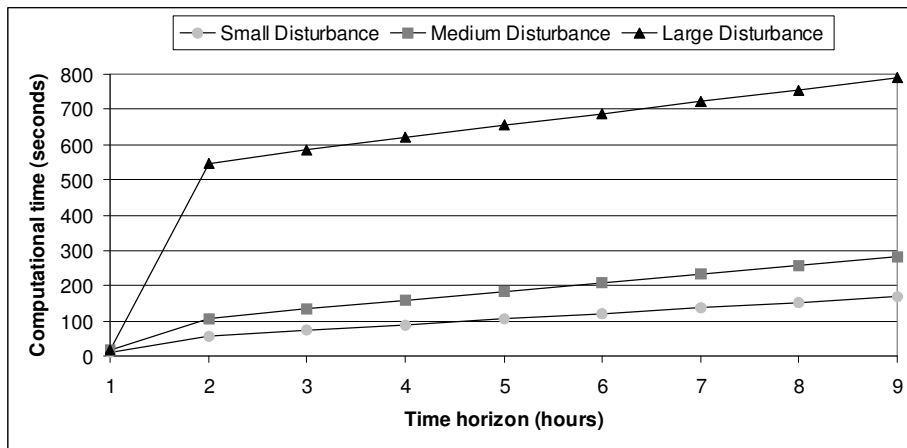


Figure 8.4: Computation times for short-term prediction of railway traffic

Figure 8.4 shows the average times to compute short-term traffic predictions. Results are divided into the three disturbances categories of Table 8.1. These computational results show that ROMA can handle one hour instances within a few seconds (independently from the disturbance category), which is a reasonable time horizon length for a real-time decision support system. When dealing with larger time horizons and up to medium disturbances, ROMA returns, on average, a solution within five minutes of computation. Starting from a time horizon of two hours, large disturbances are already too time consuming compared to real-time application purposes, i.e., considerable modifications of speed and location of trains may occur while the dispatching support system is computing a solution. In this case, the most critical factor for a real-time use of such a system is not the time horizon length but the level of disturbance in the network. Besides, the presence of several blocked tracks may cause problems when computing a new feasible timetable in real-time.

Figures 8.5 and 8.6 give the average results in terms of maximum and average total delays. The propagation of train delays reflects clearly the enforced disturbances. Since traffic perturbations are inserted in the first hour of timetable, in general, the maximum total delays increase between the first two hours of traffic prediction and decrease strongly after the second hour of traffic prediction because of available recovery times. The average total delays also decrease when enlarging the time horizon length. However, the successive hours are still perturbed because of propagation of delays and of the presence of blocked tracks that do not allow a complete return to the original traffic situation.

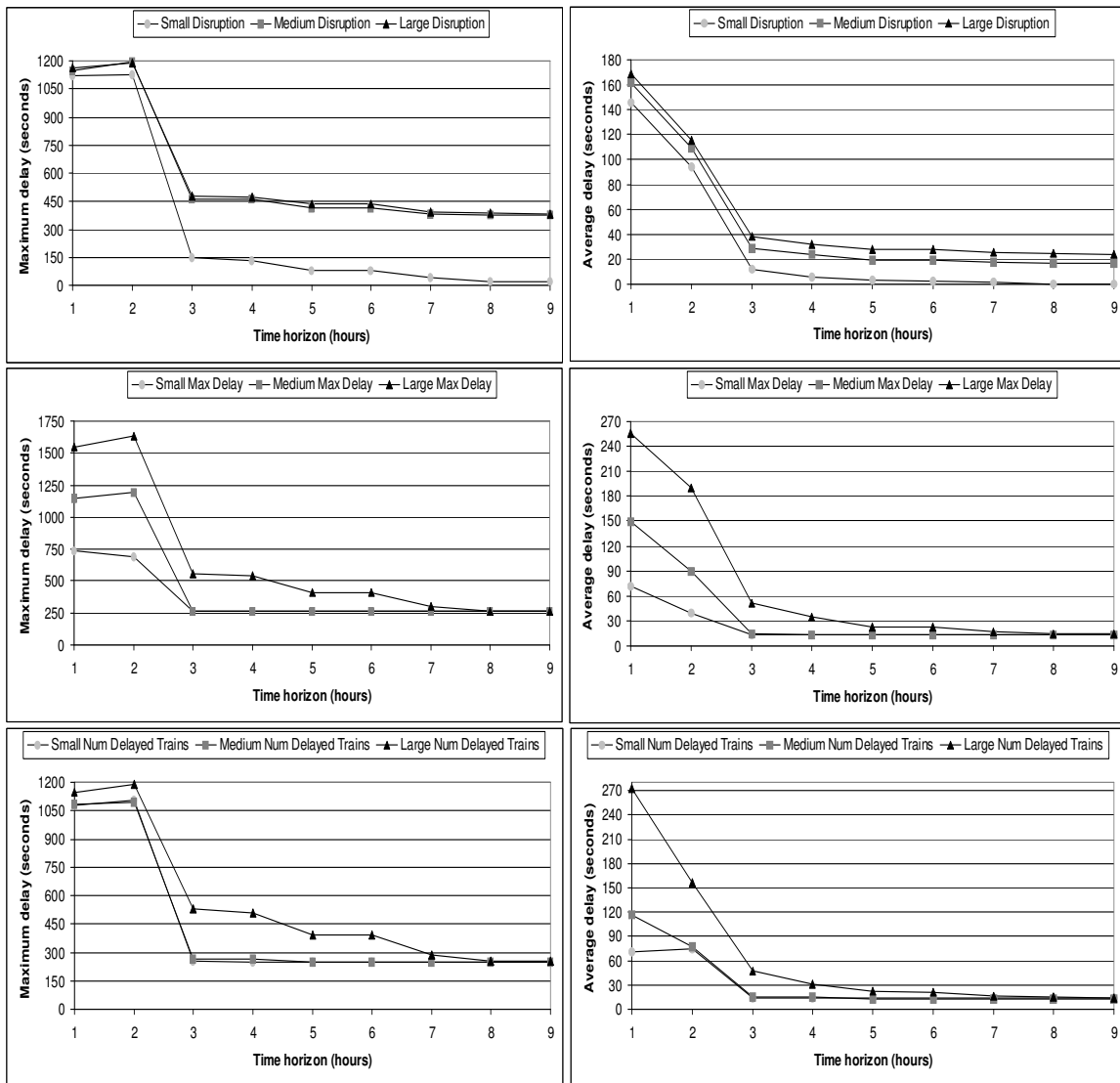


Figure 8.5: Short-term propagation of train delays varying the timetable disturbances (disruptions, maximum entrance delays and number of delayed trains)

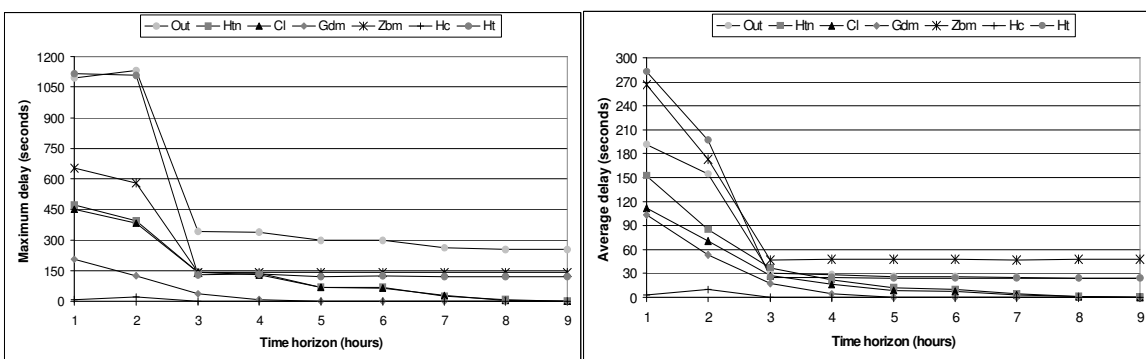


Figure 8.6: Short-term train delay propagation at stations (Htn, Cl, Gdm, Zbm, Hc and Ht) and dispatching area borders (“Out”)

Figure 8.5 reports the total delays on the basis of the different types of disturbances. The results, obtained for the overall set of instances, are divided into six plots. The three plots on the left show the maximum total delays for each disturbance category of Table 8.1 by varying the timetable disruption, the maximum entrance delay and the number of delayed trains. The average total delays are similarly shown on the right of the figure. These plots describe precisely the propagation of train delays, while dispatchers usually cannot precisely evaluate the short-term consequences of timetable disturbances in complicated railway networks. In fact, it is impossible for them to keep a complete check on all trains running in the railway network and predict and control their future movements.

When dealing only with multiple delayed trains (small disruption), the maximum total delay would drop to 20 seconds and the average total delays will fall down to zero after a time period of 9 hours. In case of blocked tracks (i.e., medium and large disruptions), the total delays are more difficult to recover as the trains change their scheduled routes. Different maximum values of entrance perturbations have a significant direct impact on the total delays. Similar results are obtained when changing the number of delayed trains.

As far as the total delay at each relevant point is concerned, from Figure 8.6 we observe that Den Bosch (Ht) is the most critical stop in the first two hours of traffic prediction. From the third hour on, the critical station becomes Zaltbommel (Zbm) and the largest delays are registered in the border areas of the dispatching area (“Out”). In general, when the STTP problem enlarges over three hours the average total delay stabilizes at Zaltbommel station, Den Bosch station and the border areas while decreasing at the other stations. After 9 hours of traffic prediction, the average total delay would become less than one minute at all relevant points whereas the maximum total delay at the border areas is reduced of 77%. Furthermore, Den Bosch (Ht) registers the largest delay in the second hour of traffic prediction, i.e., around 20 minutes. This value is greater than the maximum total delay at the other stations of this dispatching area.

Note that in this set of experiments we enlarged the size of the Utrecht - Den Bosch railway network and the number of trains running in this network compared to the section of computational experiments presented in Chapter 7. However, the instances of one hour traffic prediction are still solved by ROMA within a few seconds of computation time. More in general, the results described here can be compared with the computational experiments presented for the Schiphol railway network that is a shorter (only 20 km long) but more complicated and densely used dispatching area (as described in Appendix A). For the Schiphol railway network, test cases consisting of a time horizon of one hour and a timetable perturbation with multiple delayed trains are still solvable within a short computation time, i.e., between two and four minutes (as shown in Table 2 of Chapter 7).

To summarize, the complexity of the automated dispatching task depends on the combination of several factors that are the complexity and density of the railway network, the length of the time horizon of traffic prediction, the level of disturbance in the network and the delivery time for computing the real-time dispatching measures. In fact, these factors play a key role in real-time railway traffic management since the decisions generated by

the dispatching support system using a limited re-planning perspective must fit well with the current evolution of train traffic in order to be implemented during operations.

8.3 Conclusions

In this chapter, we enlarge the performance of our dispatching support system in order to forecast the propagation of train delays up to several timetable hours. We decompose a long time horizon into tractable intervals to be solved in cascade, and use advanced CDRFR algorithms to pro-actively detect and globally solve train conflicts on each time interval. While the temporal decomposition approach allows the resolution of large scheduling problems, the global approach results rather time consuming but offers better quality solutions in terms of train delay minimization.

The computational experiments underline the importance to determine a compromise between the length of the time horizon of traffic prediction, that limits the number of trains running in the network, and the time available for computing a dispatching solution. Moreover, the level of disturbance and the complexity and density of the railway network are other important time consuming factors to be taken into account during the traffic prediction, since in case of severe disturbances a large number of trains are involved in conflicting situations and more decisions need to be made.

Further research should therefore be dedicated to the analysis of more sophisticated techniques of problem decomposition in order to fill the gap between solution quality and computation times for more complicated and densely used railway networks, and to cope with the railway traffic management of consecutive dispatching areas. To this end, it is central to address the decomposition of large railway networks into smaller areas to be solved by local dispatching support systems in real-time, and their coordination may ensure globally viable and effective solutions.

Chapter 9

Conclusions

9.1 Resume of the main achievements

The management of railway operations requires joint contributions from various disciplines such as railway system knowledge, traffic engineering, mathematics, physics, economics and computer science. This thesis combines operations research techniques with scientific and professional railway expertise in order to study the real-time train dispatching problem and to address the development of an advanced computerized dispatching support tool.

In the last years, this research area experienced an increasing interest due to the growth of train traffic and the limited possibilities of enhancing the infrastructure, which increase the needs for an efficient use of resources and the pressure on traffic controllers. However, the real-time train dispatching process is still dominated by human professional skills and rules of thumb, and strongly depends on the (off-line) timetabling quality. Furthermore, the state-of-the-art in optimal train dispatching algorithms can handle only low traffic densities and a short time horizon within a reasonable amount of computation time.

We design and implement a (laboratory) decision support system for dispatchers, that we call ROMA (Railway traffic Optimization by Means of Alternative graphs). This system is able to predict the future evolution of train traffic while considering accurately the actual track occupation, the Dutch signaling system and the dynamic train characteristics. The train speed profiles can be computed on the basis of static and dynamic data gathered from the signaling and safety system. Once the speed profiles are known, the actual running times of the trains and clearing times of the block sections are calculated and adopted to predict the railway traffic in a limited time horizon and dispatching area.

Extensive computational experiments are carried out on two complex dispatching areas of the Dutch railway network, i.e., the Schiphol railway bottleneck and the dispatching area between Utrecht and Den Bosch. We study practical size instances and different types of disturbances, including multiple delayed trains, dwell time perturbations and blockage of some tracks. ROMA is also able to optimize the railway traffic even when the timetable is not conflict-free and/or deadlock-free. This enables its usage for managing railway traffic

in case of severe traffic disturbances, such as when emergency timetables are required and dispatchers need extensive support to solve multiple conflicts simultaneously.

ROMA recovers from disturbances automatically, with the objective of minimizing the propagation of consecutive delays, and improves the dynamic use of railway infrastructure. When train operations are perturbed, a new timetable of feasible arrival and departure times is re-computed, which is a conflict-free and deadlock-free schedule. Specifically, a real-time traffic optimization procedure considers train reordering and local rerouting actions to solve the conflict detection and resolution problem, while a train speed coordination procedure takes into account the variability of train dynamics required to satisfy minimum distance headways between consecutive trains in case of conflicts.

The railway traffic optimization procedure is based on the alternative graph formulation of the conflict detection and resolution problem with fixed train speed profiles. This is a flexible mathematical model that enables the formulation of several relevant railway constraints and the implementation of powerful traffic optimization techniques.

As for the impact of railway dynamic traffic management principles (e.g. flexible departure times at scheduled stops, train reordering and rerouting alternatives), our computational results demonstrate that all the proposed principles may lead to interesting improvements over simple and local dispatching procedures. These benefits are the largest when the principles are used in combination with advanced traffic management algorithms.

Innovative reordering and rerouting algorithms are provided to compute optimal schedule adjustments. These are global conflict resolution algorithms that exploit the mathematical properties of the alternative graph formulation of the problem and perform speed-up procedures to obtain feasible real-time solutions within a dispatching area. The most effective scheduling algorithm is the branch and bound procedure, which is able to solve to optimality large scheduling instances within a short computation time. The most effective rerouting algorithm is the tabu search technique, which is able to improve the solutions computed by the branch and bound algorithm in terms of both maximum and consecutive delays.

The train speed coordination procedure is based on a single or iterative scheduling strategy that identifies pro-actively potential train conflicts, reschedules the train traffic at a limited network scale and adjusts the train speed profiles accordingly. Optimal train running profiles are also designed which fit better to the new train orders and cause less delays and energy consumption at the same time. Inserting traffic optimization algorithms in the variable-speed model allows an effective dynamic control of trains in a given time horizon of traffic prediction when perturbations and disruptions occur. A feasible schedule is thus generated that reduces consecutive delays and ensures minimal safety distances between any pair of trains while respecting the actual signaling constraints.

The computational results on the two dispatching areas demonstrate a considerable difference in the schedules obtained by the fixed-speed and variable-speed models. The fixed-speed model solutions present smaller delays because the consequences of braking and re-acceleration in case of conflicts are underestimated, while the variable-speed model solutions consider a dynamic adjustment of train speed profiles in those traffic situations.

Moreover, an extensive comparison between the variable-speed model solutions computed by advanced scheduling algorithms and simple dispatching rules proves the efficiency of the dispatching support system developed.

Another interesting problem studied in this thesis is the evaluation of the short-term effects of timetable disturbances of up to several hours. The problem is decomposed into tractable intervals to be solved in cascade by our scheduling algorithms. We also present computational results on different ROMA configurations and time horizons of traffic prediction, and demonstrate the ability of ROMA to reduce the delay propagation due to disturbances.

The consequences of variations in the complexity and density of the railway network, the length of the time horizon of traffic prediction, the level of disturbance in the network and the delivery time for computing the real-time dispatching measures have also been assessed. From the results on the two dispatching areas the following conclusions can be drawn. Since the area around Schiphol is higher occupied and more complicated than the dispatching area between Utrecht and Den Bosch, ROMA needs more time to compute a dispatching solution. Furthermore, short-term traffic predictions on the less dense dispatching area show that, for a real-time use of ROMA, the level of disturbance is a more time consuming factor than the time horizon length. These findings are more evident when dealing with severe disturbances such as multiple delayed trains and infrastructure disruptions.

9.2 Recommendations for future research

The ROMA dispatching support system should be implemented in an existing railway traffic control center in order to prove its performance in railway operations practice. Furthermore, this decision support tool for dispatchers should be able to manage railway traffic in a large network consisting of different traffic control areas. To achieve this goals, a number of issues remain that need further development.

First, ROMA should be connected to a comprehensive closed-loop traffic monitoring and control system that requires the transmission of all the relevant information during operations, such as the actual train delays, the continuous detection of the actual positions of the running trains, the computation of the realized blocking time diagrams, and the automatic application of dispatching measures proposed by ROMA. This would enable managing the trains dynamically when traffic disturbances occur. Moreover, the construction of a fully-updated railway traffic database (with dynamic information on the use of infrastructure, rolling stocks and actual schedules) would be achievable, thus improving the performance of the traffic management of larger networks, e.g. the overall Dutch railway network.

Second, ROMA should illustrate automatically the dynamic changes of the blocking time graphs and present the proposed ROMA dispatching measures by means of easy to manipulate effectiveness indicators, regarding e.g. the schedule evaluation or the minimization of energy consumption. This further feature should make the process of decision making less complex to the dispatchers who may interact with the system by a computer-aided analysis

of the current traffic situation. Moreover, the dispatching solutions could be demonstrated through a railway traffic control simulator that realistically represents of the Dutch railway network and train schedules, in order to evaluate more comprehensively the effectiveness of the ROMA algorithms compared to the existing standard dispatching techniques. This could prove the ROMA robustness as a computerized dispatching support system.

Third, we believe that the implementation of a decision support system for train drivers, including automatic cruise control with a display of “electronic sight” similar to the RouteLint tool (see e.g. Brookhuis and Taroni (2007)), would enable a more efficient regulation of the actual train speed profiles. ROMA could improve the computation of its dispatching solutions by connecting real-time the train speed coordination procedures to the above-cited decision support system for train drivers, providing an anticipatory control of train trajectories. Further research could address the problem of considering train dynamics as a problem-variable when choosing different train orders and routes at conflict points, i.e., in the presence of yellow and red signal aspects. Other research could be performed to limiting the changes of train speed profiles whenever possible while solving train conflicts in stations by suitable dwell times extensions at scheduled stops, i.e., a green wave strategy.

Fourth, further research should include other dispatching measures in case of severe disturbances, such as the modification of dwell times and connected train services or even the cancelation of some train service. These measures may influence the passing times, orders and routes of running trains, resulting in complex compound problems. In the Dutch railway practice, this is achieved so far by using emergency timetables or what-if scenarios for standard failures and incidents.

Fifth, a further research and development should address the problem of non-discriminatory evaluation of several alternative dispatching solutions. In general, other train operators’ objectives should also be taken into account in the problem formulation, such as dynamic train priorities including intercity, local and freight trains, passengers’ dissatisfaction due to extra running times or change of platform stops, et cetera.

Sixth, the effects of disturbances in large networks, i.e., the propagation of train delays over different dispatching areas and time horizons, are to be investigated more extensively. The modeling, test and realization of a decentralized or distributed control of train traffic needs to be studied. Each dispatching area could be managed e.g. by a local dispatch agent while the problem of coordinating the management of successive dispatching areas could be viewed as a multi-agent system. In detail, each local dispatch agent would receive information regarding the actual traffic flow in the other dispatching areas. This would allow a cooperative reasoning of each agent with the objective of minimizing the consecutive delays and energy consumptions for the trains running in the overall railway network. Coordination and cooperation issues between the dispatch agents should be therefore studied.

Seventh, up to now the principles of railway dynamic traffic management have been evaluated separately and on single case studies. We therefore identify the need to combine retiming, reordering and local rerouting actions and to assess the potential and limitations of these principles more in general on different dispatching areas and timetables.

Bibliography

- Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. *Management Science*, 34, 391–401.
- Adenso-Díaz, B., Oliva González, M., & González-Torre, P. (1999). On-line timetable re-scheduling in regional train services. *Transportation Research, Part B*, 33, 378–398.
- Ahuja, R. K., Cunha, C. B., & Şahin, G. (2005). Network Models in Railroad Planning and Scheduling. In H. J. Greenberg & J. C. Smith (Eds.), *TutORials in Operations Research* (pp. 54–101).
- Ahuja, R. K., Jha, K. C., & Liu, J. (2004). *Solving real-life railroad blocking problems* (Tech. Rep.). University of Florida, Gainesville, U.S.A.: Department of Industrial and Systems Engineering.
- Ahuja, R. K., Magnanti, T. L., & Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*. NJ, U.S.A.: Prentice Hall.
- Albrecht, T. (2005a). Energiesparende und verspätungsminimierende Zugsteuerung in Konfliktsituationen. *Signal und Draht*, 97 (12), 18–22. (in German)
- Albrecht, T. (2005b). Energy-efficient train control in suburban railways: experiences gained from onboard tests of a driver assistance system. In I. A. Hansen, F. M. Dekking, R. M. P. Goverde, B. Heidergott, & L. E. Meester (Eds.), *CD-ROM Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis*. Delft, the Netherlands.
- Albrecht, T. (2007). The influence of anticipating train driving on the dispatching process in railway conflict situations. In I. A. Hansen, A. Radtke, J. Pahl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany.
- Albrecht, T., & Oettich, S. (2002). A new integrated approach to dynamic schedule synchronisation and energy-saving train control. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciotto, & S. Sone (Eds.), *Computers in Railways VIII* (pp. 847–856). Southampton, UK: WIT Press.
- Albrecht, T., & Van Luipen, J. (2006). What role can a driver information system play in railway conflicts? In H. J. V. Zuylen & F. Middelham (Eds.), *CD-ROM Proceedings of the 11th IFAC Symposium on Control in Transportation Systems*. Delft, the Netherlands.
- Armstrong, J., & McDonald, M. (2005). The Vulnerability of Railway to Disruptive Incidents, and a Proposed Framework for Addressing the Issue. In I. A. Hansen, F. M. Dekking, R. M. P. Goverde, B. Heidergott, & L. E. Meester (Eds.), *CD-ROM*

- Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis*. Delft, the Netherlands.
- Arpaci, M., & Jungherr, I. (2006). A System for Improving Network Capacity. In *INFORMS Annual Meeting*. Pittsburgh, Pennsylvania, U.S.A. (Presentation available at the web site www.rasig.net)
- Assad, A. A. (1980). Models for rail transportation. *Transportation Research, Part A*, 14, 205–220.
- Asuka, M., & Komaya, K. (2003). Automatic train operation using braking pattern transitional time. In *Proceedings of the World Congress on Railway Research* (pp. 628–636). Edinburgh, UK.
- Bailey, C. (1995). *European Railway Signalling*. London, England: Institution of Railway Signal Engineers, A & C Black.
- Balas, E. (1969). Machine sequencing via disjunctive graphs: an implicit enumeration approach. *Operations Research*, 17, 941–957.
- Ball, M., Barnhart, C., Nemhauser, G., & Odoni, A. (2007). Air Transportation: Irregular Operations and Control. In G. Laporte & C. Barnhart (Eds.), *Handbooks in Operations Research and Management Science* (Vol. 14, pp. 1–67).
- Barnhart, C., Jin, H., & Vance, P. H. (2000). Railroad blocking: A network design application. *Operations Research*, 48, 603–614.
- Barter, W. (2004). Forecasting robustness of timetables. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, & S. Sone (Eds.), *Computers in Railways IX* (pp. 563–572). Southampton, UK: WIT Press.
- Berends, N., & Ouburg, N. (2005). *Beschrijving ARI-functionaliteit* (Tech. Rep. No. 20). Utrecht, the Netherlands: ProRail Internal Specification (in Dutch).
- Boudali, I., Ben Jaafar, I., & Ghedira, K. (2007). Distributed decision evaluation model in public transportation systems. *Engineering Applications of Artificial Intelligence*. (in press)
- Brännlund, U., Lindberg, P. O., Nöu, A., & Nilsson, J. E. (1998). Railway Timetabling Using Lagrangian Relaxation. *Transportation Science*, 32, 358–369.
- Brookhuis, K. A., & Taroni, I. (2007). Effects of a new railway information system on train driver efficiency and subjective performance. In D. D. Waard, G. R. J. Hockey, P. Nickel, & K. A. Brookhuis (Eds.), *Human Factors Issues in Complex System Performance* (pp. 241–250). Shaker Publishing.
- Brucker, P. (1998). *Scheduling algorithms*. Heidelberg: Springer.
- Brucker, P., Jurisch, B., & Krämer, A. (1994). The job-shop problem and immediate selection. *Annals of Operations Research*, 50(1), 73–114.
- Bussieck, M. (1998). *Optimal Lines in Public Rail Transport*. PhD thesis, Technische Universität Braunschweig.
- Cai, X., & Goh, C. J. (1994). A Fast Heuristic for the Train Scheduling Problem. *Computers and Operations Research*, 21, 499–510.
- Cai, X., Goh, C. J., & Mees, A. I. (1998). Greedy heuristics for rapid scheduling of trains on a single track. *IIE Transaction*, 30, 481–493.
- Caimi, G., Burkolter, D., & Herrmann, T. (2005). Finding delay-tolerant train routings

- through stations. In H. Fleuren (Ed.), *Selected Papers of the Annual International Conference of the German Operations Research Society Jointly Organized with the Netherlands Society* (pp. 136–143).
- Caimi, G., Burkolter, D., Herrmann, T., Chudak, F., & Laumanns, M. (2007). Design of a new railway scheduling model for dense services. In I. A. Hansen, A. Radtke, J. Pachl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany.
- Caprara, A., Fischetti, M., & Toth, P. (2002). Modeling and Solving the Train Timetabling Problem. *Operations Research*, 50, 851–861.
- Caprara, A., Kroon, L. G., Monaci, M., Peeters, M., & Toth, P. (2006). Passenger Railway Optimization. In C. Barnhart & G. Laporte (Eds.), *Handbooks in Operations Research and Management Science* (Vol. 14, pp. 129–187).
- Caprara, A., Monaci, M., Toth, P., & Guida, P. L. (2006). A Lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete Applied Mathematics*, 154, 738–753.
- Carey, M. (1994a). Extending a Train Pathing Model from One-Way to Two-Way Track. *Transportation Research, Part B*, 28, 395–400.
- Carey, M. (1994b). A Model and Strategy for Train Pathing with Choice of Lines, Platforms and Routes. *Transportation Research, Part B*, 28, 333–353.
- Carey, M. (1999). Ex ante heuristic measures of schedule reliability. *Transportation Research, Part B*, 33 (7), 473–494.
- Carey, M., & Carville, S. (2003). Scheduling and platforming trains at busy complex stations. *Transportation Research, Part A*, 37 (3), 194–224.
- Carey, M., & Crawford, I. (2007). Scheduling trains on a network of busy complex stations. *Transportation Research, Part B*, 41 (2), 159–178.
- Carey, M., & Kwiecinski, A. (1994). Stochastic approximation to the effects of headways on knock-on delays of trains. *Transportation Research, Part B*, 28 (4), 251–267.
- Carey, M., & Kwiecinski, A. (1995). Properties of expected costs and performance measures in stochastic models of scheduled transport. *European Journal of Operational Research*, 83, 182–199.
- Carey, M., & Lockwood, D. (1995). A Model, Algorithms and Strategy for Train Pathing. *Journal of the Operational Research Society*, 46, 988–1005.
- Carlier, J. (1982). The one-machine sequencing problem. *European Journal of Operational Research*, 11, 42–47.
- Carlier, J., & Pinson, E. (1989). An algorithm for solving the job-shop problem. *Management Science*, 35 (2), 164–176.
- Carlier, J., & Pinson, E. (1994). Adjustment of heads and tails for the job-shop problem. *European Journal of Operational Research*, 78, 146–161.
- Chambers, R., Carraway, R., Towe, T., & Morin, T. (1991). Dominance and decomposition heuristics for single machine scheduling. *Operations Research*, 39 (4), 639–647.
- Chiang, T. W., & Hau, H. Y. (1995). Railway Scheduling System Using Repair-based Approach. In *Proceedings of the 7th International Conference on Tools with Artificial Intelligence* (pp. 71–79).

- Chou, Y. H., Weston, P. F., & Roberts, C. (2007). Dynamic Distributed Control for Real-time Rescheduling of Railway Networks. In I. A. Hansen, A. Radtke, J. Pachl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany.
- Conte, C., & Schöbel, A. (2007). Identifying dependencies among delays. In I. A. Hansen, A. Radtke, J. Pachl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany.
- Cordeau, J. F., Toth, P., & Vigo, D. (1998). A Survey of Optimization Models for Train Routing and Scheduling. *Transportation Science*, 32 (4), 380–404.
- Corman, F., D’Ariano, A., Pacciarelli, D., & Pranzo, M. (2007). A tabu search algorithm for rerouting trains during rail operations. In *INFORMS Annual Meeting*. Seattle, Washington, U.S.A. (First prize winner at RAS Student Paper Competition. Presentation available at the web site www.rasig.net)
- Corman, F., D’Ariano, A., Pacciarelli, D., & Pranzo, M. (2008). Evaluation of green wave policy in real-time railway traffic management. In *Proceedings of the 10th International Conference on Applications of Advanced Technologies in Transportation*. Athens, Greece. (Accepted for presentation, 27-31 May 2008)
- Crainic, T. G. (2003). Long-Haul Freight Transportation. In R. W. Hall (Ed.), *Handbooks in Transportation Science* (Vol. 56, pp. 451–516).
- Daamen, W., Goverde, R. M. P., & Hansen, I. A. (2007). Non-Discriminatory Automatic and Distinct Registration of Primary and Secondary Train Delays. In I. A. Hansen, A. Radtke, J. Pachl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany.
- D’Ariano, A., & Albrecht, T. (2006). Running time re-optimization during real-time timetable perturbations. In J. Allan, C. A. Brebbia, A. F. Rumsey, G. Sciotto, S. Sone, & C. J. Goodman (Eds.), *Computers in Railways X* (pp. 531–540). Southampton, UK: WIT Press.
- D’Ariano, A., Corman, F., & Hansen, I. A. (2008). Railway traffic optimization by advanced scheduling and rerouting algorithms. In *Proceedings of the World Congress on Railway Research*. Seoul, Korea. (Accepted for presentation, 18-22 May 2008)
- D’Ariano, A., Corman, F., Pacciarelli, D., & Pranzo, M. (2006a). Modelling reordering and local rerouting strategies to solve train conflicts during rail operations. In *INFORMS Annual Meeting*. Pittsburgh, Pennsylvania, U.S.A. (Second prize winner at RASIG Student Paper Competition. Presentation available at the web site www.rasig.net)
- D’Ariano, A., Corman, F., Pacciarelli, D., & Pranzo, M. (2006b). Real-time train conflict detection and resolution: Global sequencing and local rerouting. In H. J. V. Zuylen (Ed.), *Proceedings of the 9th TRAIL Congress: Selected Papers, TRAIL Conference Proceedings P2006* (pp. 77–92). Delft University Press. (Nominated as one of the best papers at the conference)
- D’Ariano, A., Corman, F., Pacciarelli, D., & Pranzo, M. (2007). A tabu search algorithm for a short-term train rerouting problem. In *1st Annual ARRIVAL Workshop (Robust planning and Rescheduling in Railways)*. Utrecht, the Netherlands. (Presentation available at the web site www.ecopt.nl)

- D'Ariano, A., Corman, F., Pacciarelli, D., & Pranzo, M. (2008). Reordering and local rerouting strategies to manage train traffic in real-time. *Transportation Science*. (Submitted for publication in the Focused Issue of Railroad Applications)
- D'Ariano, A., Hansen, I. A., & Hemelrijk, R. (2006). Railway dynamic traffic management: Application of a train rescheduling system reducing delays. In H. J. V. Zuylen & F. Middelham (Eds.), *CD-ROM Proceedings of the 11th IFAC Symposium on Control in Transportation Systems*. Delft, the Netherlands.
- D'Ariano, A., & Hemelrijk, R. (2006). Designing a multi-agent system for cooperative train dispatching. In A. Dolgui, G. Morel, & C. Pereira (Eds.), *Information Control Problems in Manufacturing* (pp. 367–372).
- D'Ariano, A., Pacciarelli, D., & Pranzo, M. (2005). Flexible timetables for real-time train rescheduling. In W. H. K. Lam & J. Yan (Eds.), *Proceedings of the 10th International Conference of Hong Kong Society for Transportation Studies* (pp. 540–549). Hong Kong Polytechnic University.
- D'Ariano, A., Pacciarelli, D., & Pranzo, M. (2006). Conflict resolution strategies in real-time railway traffic management. In *Proceedings of the 11th Jubilee Meeting of the EURO Working Group on Transportation* (pp. 677–681). Bari, Italy.
- D'Ariano, A., Pacciarelli, D., & Pranzo, M. (2007a). A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183 (2), 643–657.
- D'Ariano, A., Pacciarelli, D., & Pranzo, M. (2007b). Effects of flexible timetables in real-time scheduling of rail operations. In *Preprints of the 6th Triennial Symposium on Transportation Analysis*. Phuket Island, Thailand.
- D'Ariano, A., Pacciarelli, D., & Pranzo, M. (2008). Assessment of flexible timetables in real-time traffic management of a railway bottleneck. *Transportation Research, Part C*, 16 (2), 232–245.
- D'Ariano, A., Pacciarelli, D., Pranzo, M., & Hemelrijk, R. (2007). Evaluating the performance of railway dynamic traffic management. In *DVD-ROM Proceedings of the 11th World Conference on Transport Research*. Berkeley, U.S.A.
- D'Ariano, A., & Pranzo, M. (2004). A real-time train dispatching system based on the blocking time theory. In P. H. L. Bovy (Ed.), *Proceedings of the 8th TRAIL Congress: Selected Papers, TRAIL Conference Proceedings P2004/1* (pp. 129–152). Delft University Press. (Nominated as one of the best papers at the conference)
- D'Ariano, A., & Pranzo, M. (2005). Conflict resolution and train speed co-ordination for solving timetable perturbations. In I. A. Hansen, F. M. Dekking, R. M. P. Goverde, B. Heidergott, & L. E. Meester (Eds.), *CD-ROM Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis*. Delft, the Netherlands. (Nominated as one of the best papers at the conference. Presentation available at the web site www.jror.nl)
- D'Ariano, A., & Pranzo, M. (2007). An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. In I. A. Hansen, A. Radtke, J. Pachl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and*

- Analysis*. Hannover, Germany. (First prize winner of the Young Researcher Award. To appear in *Networks and Spatial Economics*. Presentation available at the web site www.iaror.org)
- 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.
- Davidsson, P., Henesey, L., Ramstedt, L., Törnquist, J., & Wernstedt, F. (2005). An analysis of agent-based approaches to transport logistics. *Transportation Research, Part C*, 13, 255–271.
- Delorme, X., Rodriguez, J., & Gandibleux, X. (2001). Heuristics for railway infrastructure saturation. In *Electronic Notes in Theoretical Computer Science* (Vol. 50 (1), pp. 39–53).
- Demitz, J., Höbschen, C., & Albrecht, T. (2004). Timetable Stability – Using Simulation to Ensure Quality in a Regular Interval Timetable. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciotto, & S. Sone (Eds.), *Computers in Railways IX* (p. 549-562). Southampton, UK: WIT Press.
- De Schutter, B., & Van Den Boom, T. J. J. (2002). Connection and speed control in railway systems — A model predictive control approach. In M. Silva, A. Giua, & J. M. Colom (Eds.), *Proceedings of the 6th International Workshop on Discrete Event Systems* (pp. 49–54). Zaragoza, Spain.
- De Schutter, B., Van Den Boom, T. J. J., & Hegyi, A. (2002). A model Predictive Control Approach for Recovery from Delays in Railway Systems. *Transportation Research Record*, 1793, 15–20.
- Dessouky, M. M., Lu, Q., Zhao, J., & Leachman, R. C. (2006). An exact solution procedure to determine the optimal dispatching times for complex rail networks. *IIE Transaction*, 38 (2), 141–152.
- Dorfman, M. J., & Medanic, J. (2004). Scheduling trains on a railway network using a discrete event model of railway traffic. *Transportation Research, Part B*, 38, 81–98.
- Fay, A. (2000). A fuzzy knowledge-based system for railway traffic control. *Engineering Application of Artificial Intelligence*, 13, 719–729.
- Flamini, M. (2005). *Job-Shop scheduling algorithms with applications to railway traffic control*. PhD thesis, Università Roma Tre.
- Flamini, M., & Pacciarelli, D. (2007). Real time management of a metro rail terminus. *European Journal of Operational Research*. (in press)
- Frank, O. (1966). Two-way traffic on a single line of railway. *Operations Research*, 14 (5), 801–811.
- Franke, R., Terwiesch, P., & Meyer, M. (2000). An algorithm for the optimal control of the driving of trains. In *Proceedings of the 39th IEEE International Conference on Decision and Control* (pp. 2123–2127).
- Gandibleux, X., Jorge, J., Angibaud, S., Delorme, X., & Rodriguez, J. (2005). An ant colony optimization inspired algorithm for the set packing problem with application to railway infrastructure. In *Proceedings of the 6th Metaheuristics International Conference* (pp. 390–396).

- Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to Theory of NP-Completeness*. New York: Freeman.
- Geitz, W. D. (2007). Railway Operations Research in Emerging Economies – A Practical Point of View. In I. A. Hansen, A. Radtke, J. Pahl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany.
- Gély, L., Dessagne, G., & Lérin, C. (2006). Modelling Train Re-scheduling with Optimization and Operational Research Techniques: Results and Applications at SNCF. In *CD-ROM Proceedings of the World Congress on Railway Research*. Montreal, Canada.
- Ghoseiri, K., Szidarovszky, F., & Asgharpour, M. J. (2004). A multi-objective train scheduling model and solution. *Transportation Research, Part B*, 38, 927–952.
- Giannettoni, M., & Savio, S. (2004). The European Project COMBINE 2 to Improve Knowledge on Future Rail Traffic Management Systems. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, & S. Sone (Eds.), *Computers in Railways IX* (pp. 695–704). Southampton, UK: WIT Press.
- Glover, F. (1986). Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers and Operations Research*, 13, 533–549.
- Glover, F., & Laguna, M. (1997). *Tabu Search*. Boston, U.S.A.: Kluwer Academic Publishers.
- Goddard, E. (2006). Overview of signalling and train control systems. In *The 9th Institution of Engineering and Technology Professional Development Course on Electric Traction Systems* (p. 336-350). Manchester, UK.
- Goh, C. J., & Mees, A. I. (1991). Optimal control on a graph with application to train scheduling problems. *Mathematical and Computer Modelling*, 15 (2), 49–58.
- Goossens, J. H. M., Van Hoesel, C. P. M., & Kroon, L. G. (2004). A Branch-and-Cut Approach for Solving Railway Line-Planning Problems. *Transportation Science*, 38, 379–393.
- Goossens, J. H. M., Van Hoesel, C. P. M., & Kroon, L. G. (2005). On Solving Multi-type Railway Line Planning Problems. *European Journal of Operational Research*, 168, 403–424.
- Gorman, M. F. (1998). An application of Genetic and Tabu Searches to the Freight Railroad Operating Plan Problem. *Annals of Operations Research*, 78, 51–69.
- Goverde, R. M. P. (1998). Synchronization Control of Scheduled Train Services to Minimize Passenger Waiting Times. In P. H. L. Bovy (Ed.), *CD-ROM Proceedings of the 4th TRAIL Congress*. Delft University Press.
- Goverde, R. M. P. (1999). Improving punctuality and transfer reliability by railway timetable optimization. In P. H. L. Bovy (Ed.), *CD-ROM Proceedings of the 5th TRAIL Congress*. Delft University Press.
- Goverde, R. M. P. (2005). *Punctuality of Railway Operations and Timetable Stability Analysis*. PhD thesis, Delft University of Technology.
- Goverde, R. M. P., Bovy, P. H. L., & Olsder, G. J. (1998). The Max-plus algebra approach to transportation problems. In H. Meersman & E. V. D. Voorde (Eds.), *Proceedings*

- of the 8th World Conference on Transport Research* (Vol. 3, pp. 377–390). Antwerp, Belgium.
- Goverde, R. M. P., & D’Ariano, A. (2005). Railway traffic monitoring and real-time rescheduling. In *CD-ROM Proceedings of the Bijdragen Verkeerskundige Werkdagen crow*. Hilversum, the Netherlands.
- Goverde, R. M. P., & Hansen, I. A. (2000). TNV-Prepare: Analysis of Dutch Railway Operations Based on Train Detection Data. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciotto, & S. Sone (Eds.), *Computers in Railways VII* (pp. 779–788). Southampton, UK: WIT Press.
- Goverde, R. M. P., & Odijk, M. A. (2002). Performance evaluation of network timetables using PETER. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciotto, & S. Sone (Eds.), *Computers in Railways VIII* (p. 731-740). Southampton, UK: WIT Press.
- Grabowski, J., & Pempera, J. (2000). Sequencing of jobs in some production system. *European Journal of Operational Research*, 125 (3), 535–550.
- Graham, R. L., Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. H. G. (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, 4, 287–326.
- Hailes, S. (2006). Modern telecommunications systems for train control. In *The 11th Institution of Engineering and Technology Professional Development Course on Railway Signalling and Control Systems* (pp. 185–192).
- Hall, N. J., & Sriskandarajah, C. (1996). A survey on machine scheduling problems with blocking and no-wait in process. *Operations Research*, 44, 510–525.
- Hansen, I. A. (2001). Improving railway punctuality by automatic piloting. In *Proceedings of the IEEE International Conference on Intelligent Transportation Systems* (pp. 792–797).
- Hansen, I. A. (2004). Increase of capacity through optimized timetabling. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciotto, & S. Sone (Eds.), *Computers in Railways IX* (pp. 529–538). Southampton, UK: WIT Press.
- Hansen, I. A. (2006). State-of-the-art of railway operations research. In J. Allan, C. A. Brebbia, A. F. Rumsey, G. Sciotto, S. Sone, & C. J. Goodman (Eds.), *Computers in Railways X* (pp. 565–577). Southampton, UK: WIT Press.
- Hansen, I. A., Dekking, F. M., Goverde, R. M. P., Heidergott, B., & Meester, L. E. (2005). (eds) CD-ROM Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis. Delft, the Netherlands.
- Hansen, I. A., Radtke, A., Pahl, J., & Wendler, E. (2007). (eds) CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis. Hannover, Germany.
- Harker, P. T. (1995). Services and Technology: Reengineering the Railroads. *Interfaces*, 25, 72–80.
- Hemelrijk, R., Kruijer, J., & De Vries, D. K. (2003). *Schiphol tunnel 2007. Description of the situation* (Tech. Rep. No. 22). Utrecht: Holland Railconsult.
- Herrmann, T. M. (2005). *Stability of Timetables and Train Routings through Station Regions*. PhD thesis, Swiss Federal Institute of Technology, Zurich.

- Hertel, G. (1992). Die maximale Verkehrsleistung und die minimale Fahrplanempfindlichkeit auf Eisenbahnstrecken. *Eisenbahntechn. Rundschau ETR*, 41 (10), 665–672.
- Higgins, A. (1997). *Optimisation of Train Schedules to Minimise Transit Time and Maximise Reliability*. PhD thesis, Queensland University of Technology.
- Higgins, A., & Kozan, E. (1997). Heuristic techniques for single line train scheduling. *Journal of Heuristics*, 3, 43–62.
- Higgins, A., Kozan, E., & Ferreira, L. (1996). Optimal scheduling of trains on a single line track. *Transportation Research, Part B*, 30, 147–161.
- Hirai, C., Tomii, N., Tashiro, Y., Kondou, S., & Fujimori, A. (2006). An algorithm for train rescheduling using rescheduling pattern description language R. In J. Allan, C. A. Brebbia, A. F. Rumsey, G. Sciutto, S. Sone, & C. J. Goodman (Eds.), *Computers in Railways X* (pp. 551–561). Southampton, UK: WIT Press.
- Ho, T. K., Norton, J. P., & Goodman, C. J. (1997). Optimal Traffic Control at Railway Junctions. *IEE Proceedings*, 144, 140–148.
- Ho, T. K., & Yeung, T. H. (2001). Railway junction traffic control by heuristic methods. *IEE Proceedings Electric Power Application*, 148 (1), 77–84.
- Hooghiemstra, J. S., Kroon, L. G., Odijk, M. A., Salomon, M., & Zwaneveld, P. J. (1999). Decision Support Systems Support the Search for Win-Win Solutions in Railway Network Design. *Interfaces*, 29 (2), 15–32.
- Hooghiemstra, J. S., Middelkoop, A. D., & Teunisse, M. J. G. (1998). Improving the Dutch railway services by network-wide timetable simulation. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, & B. Mellitt (Eds.), *Computers in Railways VI* (p. 299–308). Southampton, UK: WIT Press.
- Howlett, P. G., Milroy, I. P., & Pudney, P. J. (1994). Energy-efficient train control. *Control Engineering Practice*, 2 (2), 193–200.
- Huerlimann, D. (2001). *Objektorientierte Modellierung von Infrastrukturelementen und Betriebsvorgängen im Eisenbahnwesen*. PhD thesis, ETH Zurich.
- Huisman, T., & Boucherie, R. J. (2001). Running times on railway sections with heterogeneous train traffic. *Transportation Research, Part B*, 35, 271–292.
- 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, 30–51.
- Isaai, M. T., & Cassaigne, N. P. (2001). Predictive and reactive approaches to the train scheduling problem: A knowledge management perspective. *IEEE Transactions on Systems, Man and Cybernetics – Part C: Applications and Reviews*, 31 (4), 476–484.
- Iyer, R. V., & Gosh, S. (1995). DARYN – A distributed decision-making algorithm for railway networks: Modeling and simulation. *IEEE Transactions on Vehicular Technology*, 44 (1), 180–191.
- Jackson, J. R. (1955). *Scheduling a production line to minimize maximum tardiness* (Tech. Rep. No. 43). Los Angeles: University of California. (Management Science Research Project)
- Jacobs, J. (2004). Reducing delays by means of computer-aided ‘on-the-spot’ rescheduling. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, & S. Sone (Eds.), *Computers in*

- Railways IX* (pp. 603–612). Southampton, UK: WIT Press.
- Jain, A. S., & Meeran, S. (1999). Deterministic job-shop scheduling: Past, present and future. *European Journal of Operational Research*, 113(2), 390–434.
- Jespersen-Groth, J., Clausen, J., & Larsen, J. (2007). Optimal reinsertion of cancelled train line. In I. A. Hansen, A. Radtke, J. Pachl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany.
- Jovanovic, D. (1989). *Improving railroad in-time performance: models, algorithms and applications*. PhD thesis, Decision Science Department, Wharton School, University of Pennsylvania.
- Jovanovic, D., & Harker, P. T. (1990). A Decision Support System for Train Dispatching: An Optimization-Based Methodology. *Journal of the Transportation Research Forum*, 31, 25–37.
- Jovanovic, D., & Harker, P. T. (1991). Tactical Scheduling of Train Operations: The SCAN I System. *Transportation Science*, 25, 46–64.
- Kaminsky, R. (2001). *Pufferzeiten in Netzen des spurgeführten Verkehrs in Abhängigkeit von Zugfolge und Infrastruktur*. PhD thesis, Arbeiten Institut für Verkehrswesen, Eisenbahnbau und -betrieb, Universität Hannover. (In German)
- Katwijk, R. (2008). *Multi-Agent Look-Ahead Traffic-Adaptive Control*. PhD thesis, Delft University of Technology.
- Kauppi, A., Wikström, J., Sandblad, B., & Andersson, A. W. (2006). Future train traffic control: control by re-planning. *Cognition, Technology & Work*, 8 (1), 50–56.
- Kawakami, T. (1997). Future Framework for Maglev Train Traffic Control System Utilizing Autonomous Decentralized Architecture. In *Proceedings of the 3rd International Symposium on Autonomous Decentralized Systems* (pp. 327–334).
- Khmelnitsky, E. (2000). On an optimal control problem of train operation. *IEEE Transactions on Automatic Control*, 45 (7), 1257–1266.
- Komaya, K., & Fukuda, T. (1991). A knowledge-based approach for railway scheduling. In *Proceedings of the 7th IEEE International Conference on Artificial Intelligence for Applications* (pp. 404–411). IEEE Comput. Soc. Press.
- Konig, S., & Schnieder, E. (2001). Modeling and Simulation of an Operation Concept for Future Rail Traffic. In *Proceedings of the 4th IEEE International Conference on Intelligent Transportation Systems* (pp. 808–812).
- Kraay, D. R., & Harker, P. T. (1995). Real-time Scheduling of Freight Railroads. *Transportation Research, Part B*, 29, 213–229.
- Kraay, D. R., Harker, P. T., & Chen, B. (1991). Optimal pacing of trains in freight railroads: model formulation and solution. *Operations Research*, 39, 82–99.
- Kraft, E. (1987). A Branch and Bound Procedure for Optimal Train Dispatching. *Journal of the Transportation Research Forum*, 28 (3), 263–276.
- Kroon, L. G., Romeijn, H., & Zwaneweld, P. J. (1997). Routing trains through railway networks: complexity issues. *European Journal of Operational Research*, 98, 485–498.
- Lamma, E., Mello, P., & Milano, M. (1997). A distributed constraint-based scheduler.

- Artificial Intelligence in Engineering*, 11, 91–105.
- Landex, A., Kaas, A. H., & Hansen, S. (2006). *Railway Operation* (Tech. Rep. No. 4). Technical University of Denmark: Centre for Traffic and Transport.
- Lee, T. S., & Gosh, S. (2001). Stability of RYNSORD – A decentralized algorithm for railway networks under perturbations. *IEEE Transactions on Vehicular Technology*, 50 (1), 287–301.
- Lenstra, J. K., & Rinooy, A. H. G. (1979). Computational complexity of discrete optimization problems. *Annals of Discrete Mathematics*, 4, 121–140.
- Liebchen, C. (2004). Symmetry for Periodic Railway Timetables. In *Electronic Notes in Theoretical Computer Science* (Vol. 92, pp. 34–51).
- Liebchen, C., & Möhring, R. H. (2002). A Case Study in Periodic Timetabling. In *Electronic Notes in Theoretical Computer Science* (Vol. 66 (6), pp. 1–14).
- Lindner, T. (2000). *Train Schedule Optimization in Public Rail Transport*. PhD thesis, Technische Universität Braunschweig.
- Lusby, R., Larsen, J., Ryan, D., & Ehrgott, M. (2006). *Routing Trains Through Railway Junctions: A New Set Packing Approach* (Tech. Rep. No. 21). Technical University of Denmark.
- Lüthi, M., Medeossi, G., & Nash, A. (2007). Evaluation of an Integrated Real-Time Rescheduling and Train Control System for Heavily Used Areas. In I. A. Hansen, A. Radtke, J. Pahl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany.
- Makkinga, F., & Zigterman, L. (2002). Network control for improved performance — A new method for on-line scheduling and dispatching. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciotto, & S. Sone (Eds.), *Computers in Railways VIII* (pp. 943–952). Southampton, UK: WIT Press.
- Manne, A. S. (1960). On the job shop scheduling problem. *Operations Research*, 8(2), 219–223.
- Marín, A., & Salmerón, J. (1996a). Tactical design of rail freight networks. Part I: Exact and heuristic models. *European Journal of Operational Research*, 90, 26–44.
- Marín, A., & Salmerón, J. (1996b). Tactical design of rail freight networks. Part II: Local search methods with statistical analysis. *European Journal of Operational Research*, 94, 43–53.
- Mascis, A., & Pacciarelli, D. (2002). Job shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, 143, 498–517.
- Mascis, A., Pacciarelli, D., & Pranzo, M. (2001, June). Train Scheduling in a Regional Railway Network. In *Preprints of the 4th Triennial Symposium on Transportation Analysis* (pp. 487–492). Sao Miguel, Portugal.
- Mascis, A., Pacciarelli, D., & Pranzo, M. (2002). *Models and algorithms for traffic management of rail networks* (Tech. Rep. No. 74). Dipartimento di Informatica e Automazione, Università Roma Tre.
- Mascis, A., Pacciarelli, D., & Pranzo, M. (2004). Scheduling Models for Short-term Railway Traffic Optimisation. In *CD-ROM Proceedings of the 9th International Conference on Computer-Aided Scheduling of Public Transport*. San Diego, CA, U.S.A.

- Mastrolilli, M., & Gambardella, L. M. (2000). Effective Neighborhood Functions for the Flexible Job Shop Problem. *Journal of Scheduling*, 3 (1), 3–20.
- Matsumoto, M. (2005). The revolution of train control system in Japan. In *Proceedings of the 7th International Symposium on Autonomous Decentralized Systems* (pp. 599–606).
- Mazzarello, M., & Ottaviani, E. (2007). A traffic management system for real-time traffic optimisation in railways. *Transportation Research, Part B*, 41 (2), 246–274.
- Meloni, C., Pacciarelli, D., & Pranzo, M. (2004). A rollout metaheuristic for job-shop scheduling problems. *Annals of Operations Research*, 131, 215–235.
- Mes, M., Van Den Heijden, M., & Van Harten, A. (2007). Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems. *European Journal of Operational Research*, 181, 59–75.
- Middelkoop, A. D., & Bouwman, B. (2001). SIMONE: Large scale train network simulations. In B. A. Peters, J. S. Smith, D. J. Medeiros, & M. W. Rohrer (Eds.), *Proceedings of the 2001 Winter Simulation Conference* (pp. 1–6).
- Middelkoop, A. D., & Hemelrijk, R. (2004). Exploring the effects of Dynamic Traffic Management. In *CD-ROM Proceedings of the Dagstuhl Seminar on Algorithmic Methods for Railway Optimization*. Schloss Dagstuhl Wadern Germany.
- Middelkoop, A. D., & Loeve, L. (2006). Simulation of traffic management with FRISO. In J. Allan, C. A. Brebbia, A. F. Rumsey, G. Sciutto, S. Sone, & C. J. Goodman (Eds.), *Computers in Railways X* (pp. 501–509). Southampton, UK: WIT Press.
- Missikoff, M. (1997). An object-oriented approach to an information and decision support system for railway traffic. In *Proceedings of the 1st International Conference on Knowledge-Based Intelligent Electronic Systems* (pp. 633–641).
- Montigel, M., Kleiner, C., & Achermann, E. (2007). Rescheduling in the ETCS Level 2 environment. In I. A. Hansen, A. Radtke, J. Pachel, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany.
- Moreno Pérez, J. A., Moreno-Vega, J. M., & Rodríguez Martín, I. (2003). Variable neighborhood tabu search and its application to the median cycle problem. *European Journal of Operational Research*, 151, 365–378.
- Mühlhans, E. (1990). Berechnung der Verspätungsentwicklung bei Zugfahrten. *Eisenbahntechn. Rundschau ETR*, 39 (7/8), 465–468.
- 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, UK: WIT Press.
- Negenborn, R. R. (2007). *Multi-Agent Model Predictive Control with Applications to Power Networks*. PhD thesis, Delft University of Technology.
- Negenborn, R. R., De Schutter, B., & Hellendoorn, J. (2006). Multi-agent model predictive control for transportation networks: Serial versus parallel schemes. In A. Dolgui, G. Morel, & C. Pereira (Eds.), *Proceedings of the 12th IFAC Symposium on Information Control Problems in Manufacturing* (pp. 339–344). Saint-Etienne, France.
- Neil, G. (2006). On board train control and monitoring systems. In *The 9th Institution of*

- Engineering and Technology Professional Development Course on Electric Traction Systems* (pp. 211–241). Manchester, UK.
- Newton, H. N., Barnhart, C., & Vance, P. H. (1998). Constructing railroad blocking plans to minimize handling costs. *Transportation Science*, 32, 330–345.
- Nie, L., & Hansen, I. A. (2005). System analysis of train operations and track occupancy at railway stations. *European Journal of Transport and Infrastructure Research*, 5 (1), 31–54.
- Odijk, M. A. (1996). A Constraint Generation Algorithm for the Construction of Periodic Railway Timetables. *Transportation Research, Part B*, 30 (6), 455–464.
- Oh, S. M., Hong, S. H., & Choi, I. C. (2004). Railway conflict detection and resolution in the Korean railway system. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, & S. Sone (Eds.), *Computers in Railways IX* (pp. 675–684). Southampton, UK: WIT Press.
- Oliveira, E. (2001). *Solving Single-Track Railway Scheduling Problem Using Constraint Programming*. PhD thesis, The University of Leeds.
- Oliveira, E., & Smith, B. M. (2000). *A Job-Shop Scheduling Model for the Single-Track Railway Scheduling Problem* (Tech. Rep. No. 21). England: School of Computing, University of Leeds.
- Olsson, N. O. E., & Hauglund, H. (2004). Influencing factors on train punctuality – results from some Norwegian studies. *Transport Policy*, 11, 387–397.
- Ovacik, I. M., & Uzsoy, R. (1997). *Decomposition Methods for Complex Factory Scheduling Problems*. Boston, U.S.A.: Kluwer Academic Publishers.
- Pacciarelli, D., & Pranzo, M. (2006). Speed regulation in rail networks. In H. J. V. Zuylen & F. Middelham (Eds.), *CD-ROM Proceedings of the 11th IFAC Symposium on Control in Transportation Systems*. Delft, the Netherlands.
- Pachl, J. (2002). *Railway Operation and Control*. Mountlake Terrace: VTD Rail Publishing.
- Pardalos, P. M., & Resende, G. C. (2002). *Handbook of Applied Optimization*. N.Y., U.S.A.: Oxford University Press.
- Peeters, L. (2003). *Cyclic Railway Timetable Optimization*. PhD thesis, Erasmus Universiteit Rotterdam.
- Pinedo, M. (1995). *Scheduling: Theory, Algorithms and System*. Prentice Hall.
- Ping, L., Axin, N., Limin, J., & Fuzhang, W. (2001). Study on intelligent train dispatching. In *Proceedings of the 4th IEEE International Conference on Intelligent Transportation Systems* (pp. 949–953).
- Pinson, E. (1990). A practical use of Jackson's preemptive schedule for solving the job shop problem. *Annals of Operations Research*, 26, 269–287.
- Pranzo, M., D'Ariano, A., & Pacciarelli, D. (2005). The effectiveness of static implications in real-time railway traffic. In A. Jaskiewicz, M. Kaczmarek, J. Zak, & M. Kubiak (Eds.), *Proceedings of the 10th Jubilee Meeting of the EURO Working Group on Transportation* (pp. 298–303). Publishing House of Poznan University of Technology, Poznan, Poland.
- Pudney, P., & Wardrop, A. (2004). Generating train plans with Problem Space Search.

- In *CD-ROM Proceedings of the 9th International Conference on Computer-Aided Scheduling of Public Transport*. San Diego, CA, U.S.A.
- Radtke, A., & Hauptmann, D. (2004). Automated Planning of Timetables in Large Railway Networks Using a Microscopic Data Basis and Railway Simulation Techniques. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, & S. Sone (Eds.), *Computers in Railways IX* (p. 615-625). Southampton, UK: WIT Press.
- Rodriguez, J. (2000). Empirical study of a railway traffic management constraint programming model. In J. Allan, R. J. Hill, C. A. Brebbia, G. Sciutto, & S. Sone (Eds.), *Computers in Railways VII* (pp. 1017–1025). Southampton, UK: WIT Press.
- Rodriguez, J. (2007a). A constraint programming model for real-time train scheduling at junctions. *Transportation Research, Part B*, 41 (2), 231–245.
- Rodriguez, J. (2007b). A study of the use of state resources in a constraint-based model for routing and scheduling trains. In I. A. Hansen, A. Radtke, J. Pachl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany.
- Roy, B., & Sussman, R. (1964). *Les problèmes d'ordonnancement avec contraintes disjointives* (Tech. Rep. No. 9). SEMA, Paris.
- Şahin, G., Ahuja, R. K., & Cunha, C. B. (2004). New approaches for the train dispatching problem. In *INFORMS Annual Meeting*. Denver, Colorado, U.S.A.
- Şahin, İ. (1999). Railway traffic control and train scheduling based on inter-train conflict management. *Transportation Research, Part B*, 33, 511–534.
- Sakowitz, C., & Wendler, E. (2006). Optimising train priorities to support the regulation of train services with the assistance of active and deductive databases. In J. Allan, C. A. Brebbia, A. F. Rumsey, G. Sciutto, S. Sone, & C. J. Goodman (Eds.), *Computers in Railways X* (pp. 489–499). Southampton, UK: WIT Press.
- Salido, M. A., Abril, M., Barber, F., Ingolotti, L., Tormos, P., & Lova, A. (2007). Domain-dependent distributed models for railway scheduling. *Knowledge-Based Systems*, 20, 186–194.
- Santos, A. J. D., Soares, A. R., De Almeida Redondo, F. M., & Carvalho, N. B. (2005). Tracking trains via radio frequency systems. *IEEE Transactions on Intelligent Transportation Systems*, 6 (2), 244–258.
- Sauder, R., & Westerman, W. (1983). Computer aided train dispatching: Decision support through optimization. *Interfaces*, 13, 24–37.
- Schaafsma, A. A. M. (2001). *Dynamisch Railverkeersmanagement; besturingsconcept voor railverkeer op basis van het Lagenmodel Verkeer en Vervoer*. PhD thesis, Delft University of Technology. (In Dutch, English summary)
- Schaafsma, A. A. M. (2005). Dynamic traffic management - innovative solution for the Schiphol bottleneck 2007. In I. A. Hansen, F. M. Dekking, R. M. P. Goverde, B. Heidergott, & L. E. Meester (Eds.), *CD-ROM Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis*. Delft, the Netherlands.
- Schaafsma, A. A. M., & Bartholomeus, M. M. G. P. (2007). Dynamic Traffic Management in the Schiphol bottleneck. In I. A. Hansen, A. Radtke, J. Pachl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Opera-*

- tions Modelling and Analysis*. Hannover, Germany.
- Schaefer, H. (1995). Computer-aided train dispatching with expert systems. In *Proceedings of the International Conference on Electric Railways in a United Europe* (pp. 28–32).
- Schöbel, A. (2001). A model for the delay management problem based on mixed-integer programming. In *CD-ROM Proceedings of the 1st Workshop on Algorithmic Methods and Models for Optimization of Railways and Electronic Notes in Theoretical Computer Science* (Vol. 50 (1), pp. 1–10). Crete, Greece.
- Scholl, S. (2005). *Customer-Oriented Line Planning*. PhD thesis, University of Kaiserslautern.
- Schrijver, A., & Steenbeek, A. (1994). *Dienstregelingontwikkeling voor Railned: Rapport CADANS 1.0* (Tech. Rep.). Amsterdam, the Netherlands: Centrum voor Wiskunde en Informatica. (In Dutch)
- Schultze, K. (1985). *Modell für die asynchrone Simulation des Betriebes in Teilen des Eisenbahnnetzes*. PhD thesis, Veröffentl. Verkehrswissenschaftl. Institut RWTH Aachen. (In German)
- Schutte, J. (2001). Recent trends in automatic train controls. In *Proceedings of the 4th IEEE International Conference on Intelligent Transportation Systems* (pp. 813–819).
- Schwanhäüßer, W. (1974). *Die Besessung der Pufferzeiten im Fahrplangefüge der Eisenbahn*. PhD thesis, Veröffentlichungen verkehrswissenschaftl, Institut RWTH Aachen. (In German)
- Schwanhäüßer, W. (1994). The status of German railway operations management in research and practice. *Transportation Research, Part A*, 28, 495–500.
- Semet, Y., & Schoenauer, M. (2005). An Efficient Memetic, Permutation-Based Evolutionary Algorithm for Real-World Train Timetabling. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation* (pp. 1761–1768).
- Serafini, P., & Ukovich, W. (1989). A Mathematical Model for Periodic Event Scheduling Problems. *SIAM Journal on Discrete Mathematics*, 2 (4), 550–581.
- Sewcyk, B., Radtke, A., & Wilfinger, G. (2007). Combining microscopic and macroscopic infrastructure planning models. In I. A. Hansen, A. Radtke, J. Pahl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany.
- Shakhlevich, N. V., Sotskov, Y. N., & Werner, F. (2000). Complexity of mixed shop scheduling problems: A survey. *European Journal of Operational Research*, 120(2), 343–351.
- Shoji, S., & Igarashi, A. (1997). New Trends of Train Control and Management Systems with Real-Time and Non-Real-Time Properties. In *Proceedings of the 3rd International Symposium on Autonomous Decentralized Systems* (pp. 319–326).
- Stadlmann, B. (2006). Automation of Operational Train Control on Regional Branch Lines by a Basic Train Control System. In *Proceedings of the 9th IEEE International Conference on Intelligent Transportation Systems* (pp. 50–54).
- Szpigel, B. (1973). Optimal Train Scheduling on a Single Track Railway. In M. Ross (Ed.), *Operational Research '72* (pp. 343–352). Amsterdam, the Netherlands.
- Takagi, R., Weston, P. F., Goodman, C. J., Bouch, C., Armstrong, J., J. Preston, & Sone,

- S. (2006). Optimal train control at a junction in the main line rail network using a new object-oriented signalling system model. In J. Allan, C. A. Brebbia, A. F. Rumsey, G. Sciutto, S. Sone, & C. J. Goodman (Eds.), *Computers in Railways X* (pp. 479–488). Southampton, UK: WIT Press.
- Takeuchi, Y., & Tomii, N. (2005). Robustness Indices for Train Rescheduling. In I. A. Hansen, F. M. Dekking, R. M. P. Goverde, B. Heidergott, & L. E. Meester (Eds.), *CD-ROM Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis*. Delft, the Netherlands.
- Tazoniero, A., Goncalves, R., & Gomide, F. (2005). Fuzzy algorithm for real-time train dispatch and control. In *Proceedings of the 24th International Conference of the North American Fuzzy Information Processing Society* (pp. 332–336).
- Törnquist, J. (2005). Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms. In *CD-ROM Proceedings of the 5th Workshop on Algorithmic Methods and Models for Optimization of Railways*. (downloaded from <http://drops.dagstuhl.de/opus/volltexte/2006/659>)
- Törnquist, J. (2006). *Railway traffic disturbance management*. PhD thesis, Blekinge Institute of Technology.
- Törnquist, J. (2007). Railway traffic disturbance management — An experimental analysis of disturbance complexity, management objectives and limitations in planning horizon. *Transportation Research, Part A*, 41, 249–266.
- Törnquist, J., & Persson, J. A. (2005). Train Traffic Deviation Handling Using Tabu Search and Simulated Annealing. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences* (pp. 1–10).
- Törnquist, J., & Persson, J. A. (2007). N-tracked railway traffic re-scheduling during disturbances. *Transportation Research, Part B*, 41 (3), 342–362.
- Ullman, J. (1975). NP-complete scheduling problems. *Journal of Computer and System Science*, 10 (3), 384–393.
- Vakhtel, S. (2002). *Rechnerunterstützte analytische Ermittlung der Kapazität von Eisenbahnnetzen*. PhD thesis, Veröffentl. Verkehrswissenschaftl. Institut RWTH Aachen. (In German)
- Van Den Boom, T. J. J., & De Schutter, B. (2007). On a model predictive control algorithm for dynamic railway network management. In I. A. Hansen, A. Radtke, J. Pahl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany.
- Van Den Top, J. (2005). Dynamic traffic management: planning with uncertainty to offer certainty. In I. A. Hansen, F. M. Dekking, R. M. P. Goverde, B. Heidergott, & L. E. Meester (Eds.), *CD-ROM Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis*. Delft, the Netherlands.
- Van Den Top, J. (2006). Effectiveness of performance indicators and process control in Dutch rail transport. In H. J. V. Zuylen (Ed.), *CD-ROM Proceedings of the 9th TRAIL Congress*. Delft University Press.
- Vernazza, G., & Zunino, R. (1990). A distributed intelligence methodology for railway traffic control. *IEEE Transactions on Vehicular Technology*, 39 (3), 263–270.

- Vieira, P., Neto, L. E., Bessa, E., & Gomide, F. (1999). Railway dispatch planning and control. In *Proceedings of the 18th International Conference of the North American Fuzzy Information Processing Society* (pp. 134–138).
- Vromans, M. J. C. M. (2005). *Reliability of Railway Systems*. PhD thesis, Erasmus Universiteit Rotterdam.
- Vromans, M. J. C. M., Dekker, R., & Kroon, L. G. (2006). Reliability and Heterogeneity of Railway Services. *European Journal of Operational Research*, 172 (2), 647–665.
- Wakob, H. (1985). *Ableitung eines generellen Wartemodells zur Ermittlung der planmäßigen Wartezeiten im Eisenbahnbetrieb unter besonderer Berücksichtigung der Aspekte Leistungsfähigkeit und Anlagenbelastung*. PhD thesis, Veröffentl, verkehrswissenschaftl, Institut RWTH Aachen. (In German)
- Watson, R. (2005). Using stochastic simulation to predict timetable performance — Status and developments in the UK. In I. A. Hansen, F. M. Dekking, R. M. P. Goverde, B. Heidergott, & L. E. Meester (Eds.), *CD-ROM Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis*. Delft, the Netherlands.
- Wegele, S., & Schnieder, E. (2005). Dispatching of train operations using genetic algorithms. In I. A. Hansen, F. M. Dekking, R. M. P. Goverde, B. Heidergott, & L. E. Meester (Eds.), *CD-ROM Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis*. Delft, the Netherlands.
- Wegele, S., Slovák, R., & Schnieder, E. (2007). Real-time decision support for optimal dispatching of train operation. In I. A. Hansen, A. Radtke, J. Pahl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany.
- Weigand, W. (1981). *Graphentheoretisches Verfahren zur Fahrplangestaltung in Transportnetzen unter Berücksichtigung von Pufferzeiten mittels interaktiver Rechentech-nik*. PhD thesis, Technische Universität Braunschweig, Braunschweig. (In German)
- Wendler, E. (1999). *Analytische Berechnung der planmäßigen Wartezeiten bei asyn-chroner Fahrplankonstruktion*. PhD thesis, Veröffentl, verkehrswissenschaftl, Institut RWTH Aachen. (In German)
- Wendler, E. (2001). Quality Management in the Operation Planning Process by Means of Harmonized Modelling. In *CD-ROM Proceedings of the World Congress on Railway Research*. Köln, Germany.
- Wendler, E. (2007). The scheduled waiting time on railway lines. *Transportation Research, Part B*, 41, 148–158.
- White, T. A. (2007). The development and use of dynamic traffic management simula-tions in North America. In I. A. Hansen, A. Radtke, J. Pahl, & E. Wendler (Eds.), *CD-ROM Proceedings of the 2nd International Seminar on Railway Operations Mod-elling and Analysis*. Hannover, Germany.
- Wong, R. C. W., & Leung, J. M. Y. (2004). Timetable Synchronization for Mass Transit Railway. In *CD-ROM Proceedings of the 9th International Conference on Computer-Aided Scheduling of Public Transport*. San Diego, CA, U.S.A.
- Wong, R. C. W., Yuen, T. W. Y., Fung, K. W., & Leung, J. M. Y. (2008). Optimizing Timetable Synchronization for Rail Mass Transit. *Transportation Science*, 42 (1),

- 57–69.
- Yasui, Y. (2006). Automatic train operation system for the high speed Shinkansen train. In J. Allan, C. A. Brebbia, A. F. Rumsey, G. Sciutto, S. Sone, & C. J. Goodman (Eds.), *Computers in Railways X* (pp. 441–446). Southampton, UK: WIT Press.
- Yuan, J. (2006). *Stochastic Modelling of Train Delays and Delay Propagation in Stations*. PhD thesis, Delft University of Technology.
- Zeng, X., Happiette, M., & Rabenasolo, B. (1998). Analysis of the temporal decomposition procedure for scheduling with release and due dates. *European Journal of Operational Research*, *109*, 599–619.
- Zhou, X., & Zhong, M. (2005). Bicriteria train scheduling for high-speed passenger railroad planning applications. *European Journal of Operational Research*, *167*, 752–771.
- Zhou, X., & Zhong, M. (2007). Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds. *Transportation Research, Part B*, *41*, 320–341.
- Zhu, P. (2001). *Betriebliche Leistung von Bahnsystemen unter Störungsbedingungen*. PhD thesis, Berichte der Institute für Automatisierungstechnik, TU Braunschweig. (In German)
- Zwaneveld, P. J., Kroon, L. G., Romeijn, E. H., Salomon, M., Dauzre-Prs, S. P. M., S. Van Hoesel, & Ambergen, H. W. (1996). Routing Trains Through Railway Stations: Model Formulation and Algorithms. *Transportation Science*, *30*, 181–194.
- Zwaneveld, P. J., Kroon, L. G., & Van Hoesel, S. P. M. (2001). Routing trains through a railway station based on a node packing model. *European Journal of Operational Research*, *128*, 14–33.

Appendix A

Test cases description

The Dutch railway network consists of around 6,500 kilometers of track and includes more than 4,500 bridges and tunnels. The tracks go across 386 stations. Approximately 5,000 passenger trains carry almost 1 million passengers a day while 230 trains transport 80,000 tons of freight. Together they make the Dutch railway network one of the busiest in Europe.

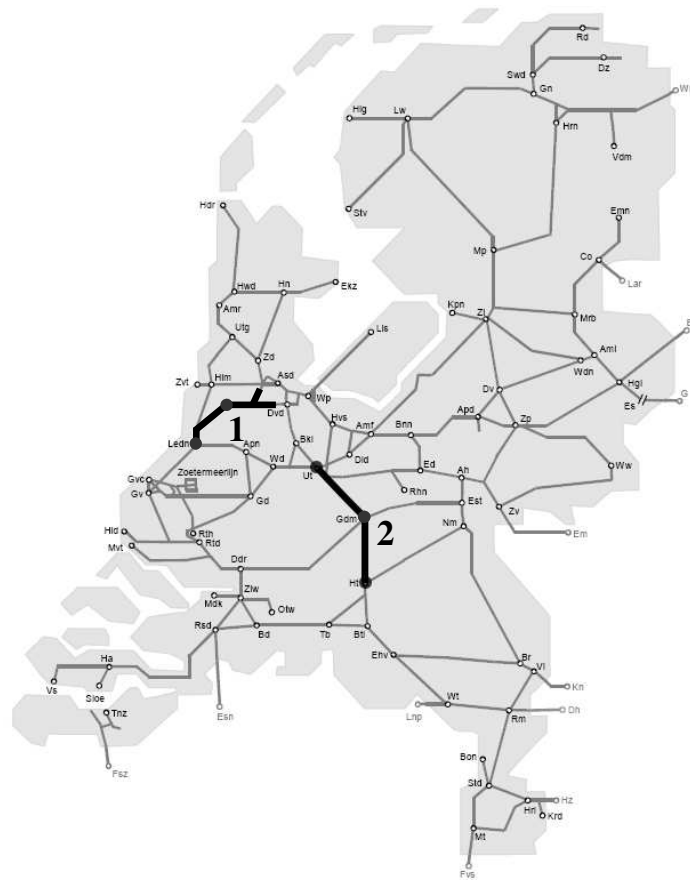


Figure A.1: The Dutch railway network (gray and black lines) and the two dispatching areas under study (black lines)

In this thesis, computational experiments are based on two bottlenecks of the Dutch railway

network, called the Schiphol dispatching area and Utrecht - Den Bosch dispatching area (indicated respectively as 1 and 2 in Figure A.1). We next give a detailed description of the two considered dispatching areas.

Schiphol dispatching area

The dispatching area around Schiphol tunnel is shown in Figure A.2. The network consists of 86 block sections, 16 platforms and two traffic directions. The railway infrastructure is around 20 km long and consists mainly of four tracks, divided into two pairs for each traffic direction. Trains enter/leave the network from/to ten access points: the High Speed Line (HSL) (block sections 20 and 59), the station of Nieuw Vennep (Nvp) (block sections 33 and 70), the yard of Hoofddorp station (HfdY) (block sections 18 and 71), and two stations in Amsterdam, namely Amsterdam Lelylaan (Asdl) (block sections 46 and 50) and Amsterdam Zuid WTC (Asdz) (block sections 77 and 45). Between Schiphol and Amsterdam, there is the Riekerpolder (Rkr) junction (block sections 26, 27, 28, 29, 85 and 75). The Hoofddorp station (Hfd) has two platforms, which are dedicated to local trains, while the Schiphol station (Shl) has six platforms, which are dedicated to all passenger trains. The two traffic directions are largely independent except around Amsterdam Lelylaan station (block sections 38, 40, 42, 44, 46, 47, 48, 49 and 50) and at the border of Hoofddorp yard (block section 4). There are several potential conflict points, merging points along each direction and two critical crossing points due to the double track use of Amsterdam Lelylaan station and Hoofddorp yard.

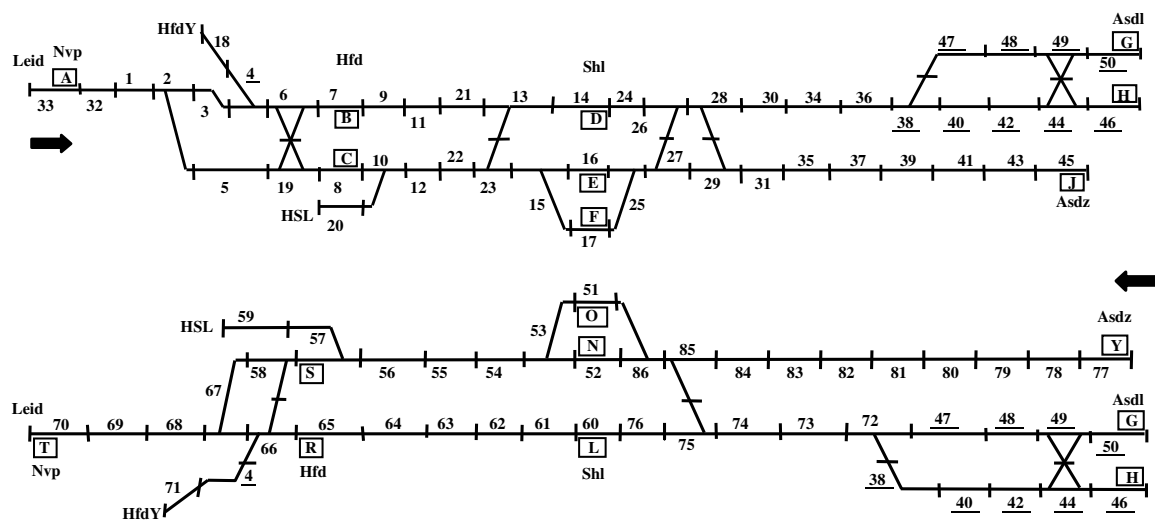


Figure A.2: Schiphol dispatching area

We consider an experimental timetable that expects an increase in traffic through this bottleneck area, thus making it an interesting test case for our study. The timetable is cyclic with a period length of one hour and contains 27 trains per direction, for a total of 54 trains running each hour. This is a (rigid) timetable with a limited amount of time reserves to

recover delays, due to the high number of trains which is not far from the network capacity saturation. It is worthwhile observing that the actual number of trains per hour scheduled at Schiphol during year 2007 is 20 trains per direction, but we chose this more challenging provisional timetable to assess the effectiveness of the ROMA dispatching support system under more severe traffic conditions. Moreover, it has to be observed that the time horizon of practical interest for railway managers is usually less than one hour for real-time purposes.

Utrecht - Den Bosch dispatching area

The Utrecht - Den Bosch dispatching area is shown in Figure A.3. The network consists of 191 block sections and 21 platforms. This includes the Den Bosch (Ht) station and the line connecting Utrecht (Ut) to Den Bosch, which is around 50 km long. There are two main tracks, divided into one long corridor for each traffic direction, a dedicated stop for freight trains nearby Zaltbommel (Ozbn) and the following passenger stations: Utrecht Lunetten (Utl), Houten (Htn), Houten Castellum (Hc), Culemborg (Cl), Geldermalsen (Gdm), Zaltbommel (Zbn) and Den Bosch (Ht). Each traffic direction has nine entrances: Utrecht (Ut), Dordrecht (Ddr), Nijmegen (Nm), Betuweroute, Geldermalsen Yard (GdmY), Oss (Oss), Eindhoven (Ehv), Den Bosch Yard (HtY) and Tilburg (Tl). There are several potential conflict points along each corridor due to different train speeds and four critical crossings/merging points: Geldermalsen station (block sections 104, 105, 109, 113, 114 and 117), Dordrecht corridor (block sections 101 and 102), Betuweroute corridor (block section 4) and Den Bosch station (block sections 142, 143, 146, 150, 151, 152, 154, 156, 157, 160, 161, 166, 167, 170, 171, 172, 180, 181, 182 and 183). Two extensions of the network, which are still under construction, are included (block sections 96, 98, 128, 129 and from 131 to 140). The infrastructure offers some possibility of train reordering and rerouting. Each train has a default route and a set of local rerouting options. Rerouting options can be applied along corridors or within a station, where a train may be allowed to stop at different nearby platforms. Only standard train routes are considered and some less important switches have been omitted. Considering all possible alternative rerouting options yields a set of 356 routes. Figure A.3 shows all possible rerouting zones, labeled from A to M.

We consider a provisional hourly timetable for 2007 extended to the entire railway area. During peak hours, 26 passenger and freight trains are scheduled, in both directions, for the area around Geldermalsen. A more complex situation occurs at Den Bosch station, where up to 40 trains are scheduled each hour. We also include constraints on the minimum transfer time between connected train services. Rolling stock connections are provided in Zaltbommel and Den Bosch stations. Passenger connections are located at Den Bosch station for the traffic directions from Oss to Utrecht and vice versa. The minimum time for passenger connections varies from two to five minutes, depending on the distance between the arrival platforms. Finally, route booking constraints are applied in stations, e.g. trains run without stop until their scheduled platform stop.

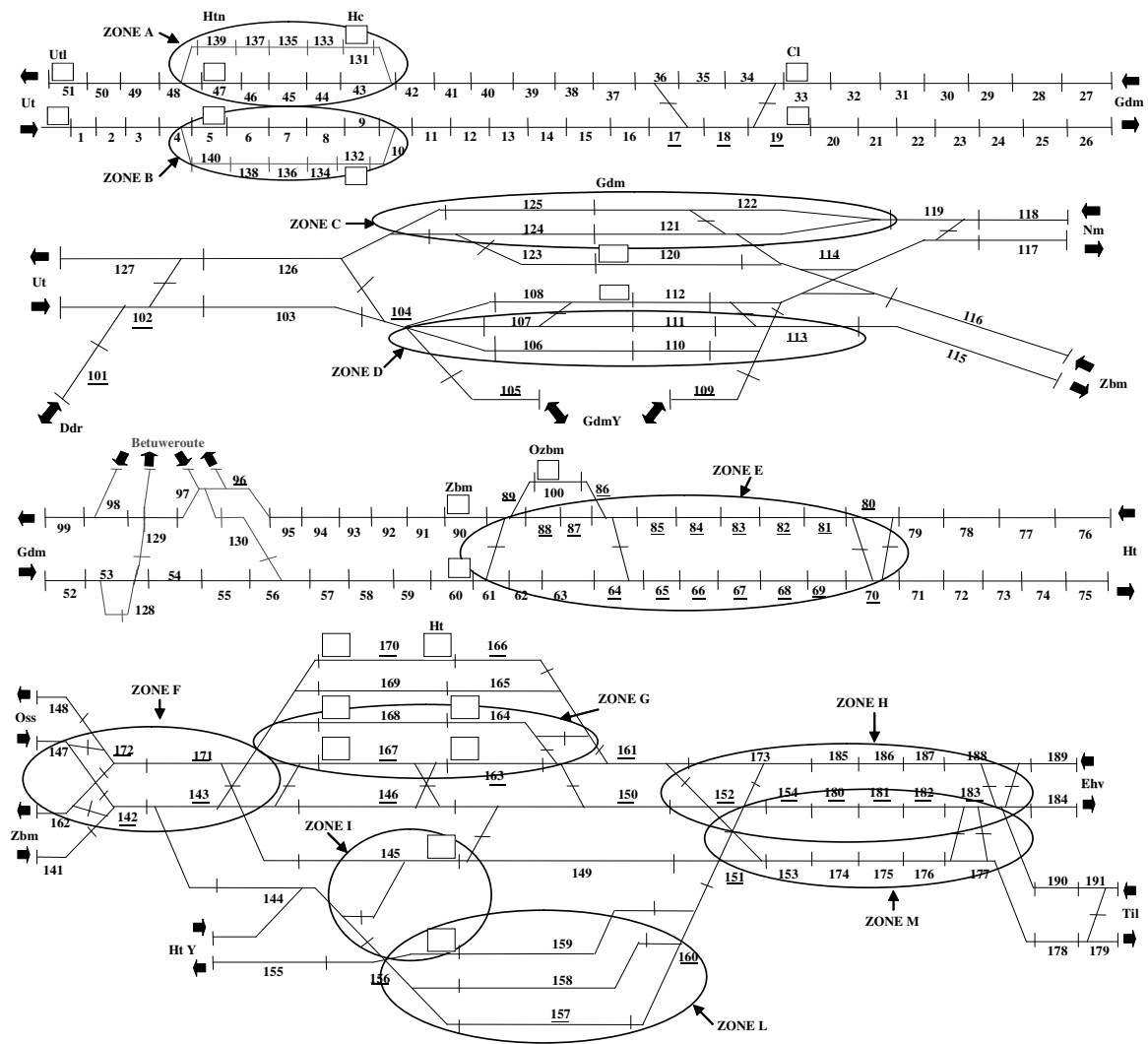


Figure A.3: Utrecht - Den Bosch dispatching area and the available rerouting zones

Appendix B

Glossary

This appendix lists the symbols used throughout this dissertation and is organized in eight tables. The first table gives the abbreviations used to describe the Dutch railway infrastructure. The three following tables contain miscellaneous acronyms for railway, mathematical and algorithmic terminologies. Finally, four tables of symbols and variables have been inserted to recall the formal description of models and algorithms for: Conflict Detection and Resolution with Fixed Routing (CDRFR), Conflict Detection and Resolution (CDR), Train Speed Coordination (TSC) and Short-Term Traffic Prediction (STTP).

Dutch railway infrastructure abbreviations

Asdl	:	Amsterdam Lelylaan	Htn	:	Houten
Asdz	:	Amsterdam Zuid	Leid	:	Leiden
Cl	:	Culemborg	Nm	:	Nijmegen
Ddr	:	Dordrecht	Nvp	:	Nieuw Vennep
Ehv	:	Eindhoven	Out	:	Exit point
Gdm	:	Geldermalsen	Ozbn	:	Stop nearby Zbn
GdmY	:	Geldermalsen Yard	Rkr	:	Riekerpolder
Hc	:	Houten Castellum	Shl	:	Schiphol
Hfd	:	Hoofddorp	Tl	:	Tilburg
HfdY	:	Hoofddorp Yard	Ut	:	Utrecht
HSL	:	High Speed Line	Utl	:	Utrecht Lunetten
Ht	:	Den Bosch	Zbn	:	Zaltbommel
HtY	:	Den Bosch Yard			

Miscellaneous railway acronyms

ARI	:	Automatische Rijweg Instelling (automatic route setting system)
ATB-NG	:	Automatische Treinbeïnvloeding - Nieuwe Generatie
ATB-EG	:	Automatische Treinbeïnvloeding - Eerste Generatie
ATP	:	Automatic Train Protection
B	:	Block section
CD	:	Conflict Detection
CDR	:	Conflict Detection and Resolution
CDRFR	:	Conflict Detection and Resolution with Fixed train Routing
COMBINE	:	enhanced COntrol centre for a Moving Block sigNalling system
CR	:	Conflict Resolution
DONS	:	Designer Of Network Schedules (Dutch timetable design system)
DSS	:	Decision Support System
ERTMS	:	European Railway Traffic Management System
ETCS	:	European Train Control System
FCFS	:	First Come First Served (i.e., first in first out)
FLFS	:	First Leave First Served (i.e., first out first in)
FRISO	:	Flexible Rail Infrastructure Simulation of Operations
NSR	:	NS Reizigers (passenger train division of Dutch railway network)
NS54	:	Dutch signaling system
P	:	Position
PETER	:	Performance Evaluation of Timed Events in Railways
PRL	:	PRocesLeidingssysteem (Dutch dispatching system, also VPT-PRL)
ProRail	:	Dutch infrastructure manager
PRULE	:	Priority Rule
Q	:	Timetable scheduled stop
RDTM	:	Railway Dynamic Traffic Management
ROMA	:	Railway traffic Optimization by Means of Alternative graphs
RWTH	:	Rheinisch Westfälische Technische Hochschule
S	:	Scenario
SCAN	:	Schedule ANalysis (train scheduling system)
SIMONE	:	Simulation MOdel of NETworks (a macroscopic traffic flow simulator)
STRESI	:	STBEche SIMulation (a tool for timetable construction and operations)
STTP	:	Short-Term Traffic Prediction
T	:	Train
TNV	:	TreinNummerVolgsysteem (Dutch train describer system)
TSC	:	Train Speed Coordination
VKL	:	VerKeersLeidingssysteem (Dutch traffic control system, VPT-VKL)
VPT	:	Vervoer Per Trein (Dutch railway planning and communication system)

Miscellaneous mathematical acronyms

AG	:	Alternative Graph
AGLibrary	:	Alternative Graph Library
ALGO	:	Algorithm
AMCC	:	Avoid Most Critical Completion Time
BB	:	Branch and Bound algorithm
BRCP	:	Backward Ramified Critical Path
C	:	Complete
CPU	:	Central Processing Unit
FBRCP	:	Forward Backward Ramified Critical Path
G	:	Gaussian
GRASP	:	Greedy Randomized Adaptative Search Procedure
JPS	:	Jackson's Preemptive Schedule
LB	:	Lower Bound
LS	:	Local Search
MILP	:	Mixed Integer Linear Program
MPC	:	Model Predictive Control
NP	:	Nondeterministic Polynomial Time
PESP	:	Periodic Event Scheduling Problem
TL	:	Tabu List
TS	:	Tabu Search
U	:	Uniform
UB	:	Upper Bound

Miscellaneous algorithmic acronyms

Avg	:	Average
BestSol	:	Best solution
Comp	:	Computation
Iter	:	Iteration
IncSol	:	Incumbent solution
L	:	List of elements
LocalBestSol	:	Local best solution
Max	:	Maximum
NextSol	:	Next solution
Num	:	Number
\mathcal{N}	:	Neighborhood structure
OptSol	:	Proven optimum solution
Tot	:	Total
x	:	Objective function value

CDRFR symbols and variables

A	:	Set of pairs of alternative (directed) arcs
a_{ij}	:	Setup time between operations o_i and o_j
α_{yu}	:	Scheduled arrival time (due date) of train T_y at B_u (of $\langle T_y, B_u \rangle$)
b_{ij}	:	Length of the (directed) arc (i, j) between nodes i and j
$\mathcal{C}(S)$:	Critical path set, i.e., the set of all critical nodes in $\mathcal{G}(S)$
d_{yu}	:	Scheduled departure time (release time) of T_y at B_u
δ_{yu}	:	Flexible departure time of T_y at B_u
e_y	:	Delay of T_y at the starting time of traffic prediction
f_i	:	Perishability constraint of operation o_i
F	:	Set of fixed (directed) arcs, i.e., set of train routes
\mathcal{G}	:	Alternative graph, i.e., $\mathcal{G} = (\mathcal{N}, \mathcal{F}, \mathcal{A})$
$\mathcal{G}(S)$:	Alternative graph solution, i.e., $(N, F \cup S)$
$\mathcal{G}(\emptyset)$:	Deselected alternative graph, i.e., $\mathcal{G}(\emptyset) = (\mathcal{N}, \mathcal{F}, \emptyset)$
i	:	i th node
(i, k)	:	Arc from node i to node k
J	:	A subset of K
K	:	A set of operations to be processed on a specific machine
$l^S(i, j)$:	Longest path from node i to node j in $\mathcal{G}(S)$
m_i	:	i th machine
$\mu(i)$:	The node that precedes node i
N	:	Set of nodes
$N(F)$:	A subset of N
o_i	:	i th operation
o_n	:	dummy “finish” operation (with zero processing time)
o_0	:	dummy “start” operation (with zero processing time)
p_i	:	Processing time of operation o_i , i.e., running time of T_y at B_u
$p_{\mu(i)}$:	Processing time of operation $o_{\mu(i)}$, i.e., approaching time of T_y at B_u
π_y	:	Scheduled entrance time (release time) of train T_y
q_i	:	Head of node i , i.e., $l^S(0, i)$
r_i	:	Tail of node i , i.e., $l^S(i, n)$
ρ_y	:	Scheduled exit time (due date) of T_y
s_i	:	Scheduled stopping time at operation o_i , i.e., dwell time of T_y at B_u
$\sigma(i)$:	The node that follows node i
S	:	Graph selection, i.e., set of train orders
S^\emptyset	:	Empty graph selection
S^*	:	Optimal graph selection
S^I	:	Initial graph selection
$Stat(i, j)$:	Alternative arcs associated to (i, j) and selected by static implications
t_i	:	Starting time of operation o_i
τ_{yu}	:	Earliest possible arrival time of T_y at B_u computed by AG

CDR symbols and variables

$A(T_y)$:	Set of pairs of alternative (directed) arcs associated to train T_y
$\mathcal{B}(F, S)$:	BRCP set, i.e., $\mathcal{B}(F, S) = \bigcup_{i \in \mathcal{C}(F, S)} [R_B(i)]$
B_u	:	u th block section
$\mathcal{C}(F, S)$:	Critical path set of $\mathcal{G}(F, S)$
F^*	:	Optimal set F
$(F, S(F))$:	Solution to the CDR problem, i.e., train routes, orders and times
$(F^*, S^*(F^*))$:	Optimal solution to the CDR problem
$\mathcal{F}(F, S)$:	FBRCP set, i.e., $\mathcal{R}(F, S) = \bigcup_{i \in \mathcal{C}(F, S)} [R_B(i) \cup R_F(i)]$
g_i	:	Hindering node of node i
\mathcal{G}'	:	Alternative graph with modified routes, i.e., $\mathcal{G}' = (\mathcal{N}, \mathcal{F}', \mathcal{A}')$
$\mathcal{G}(\mathcal{F}, S)$:	Alternative graph solution $(N, F \cup S)$
$\mathcal{G}'(\mathcal{F}', S')$:	Alternative graph solution $(N, F' \cup S')$
γ	:	Number of changed routes in a restart strategy
$l(\mathcal{C}(F, S))$:	Length of $\mathcal{C}(F, S)$
$l^{F, S(F)}(i, j)$:	Longest path from node i to node j in $\mathcal{G}(F, S)$
$l^{F', S(F')}(i, j)$:	Longest path from node i to node j in $\mathcal{G}'(F', S')$
$l^{F', S(F)}(i, j)$:	Longest path from node i to node j in $\mathcal{G}'(F', S')$ s.t. all alternative pairs in $A \cap A'$ are selected as in $S(F)$
$l(\omega)$:	Length of rerouting option ω
λ	:	Number of rerouted trains stored in a tabu list
ω	:	A chain of fixed arcs of F , i.e., a train rerouting option
P^C	:	Set of arcs of S belonging to $\mathcal{C}(F, S)$, with $P^C \subseteq S$
P^R	:	Set of arcs of F belonging to $\mathcal{R}(F, S)$
$\Pi(\omega)$:	Potential of rerouting option ω
ψ	:	Maximum number of evaluated routes
$R(i)$:	Ramification of node i
$R_B(i)$:	Backward ramification of node i
$R_F(i)$:	Forward ramification of node i
$\mathcal{R}(F, S)$:	Ramified critical path set, i.e., $\mathcal{R}(F, S) = \bigcup_{i \in \mathcal{C}(F, S)} [R(i)] \cup \mathcal{C}(F, S)$
$S(F)$:	Graph selection computed for fixed F , i.e., set of train orders for fixed train routing F
$S^*(F)$:	Optimal graph selection computed for fixed F , i.e., optimal set of train orders for fixed train routing F
T_y	:	y th train

TSC symbols and variables

t_β	:	Minimum time interval between the blocking times of two consecutive trains along a line between two stations
$t_{current}$:	Current time of the resolution procedure
t_{update}	:	Time in which the driver of the following train has to start braking according to the signaling system, i.e., time in which the train starts deviating from the scheduled speed profile due to the driver's braking action
$v_{approach}$:	Train approaching speed at the NS54 yellow signal aspect (40 km/h)
$v_{operating}$:	Train operating speed on an open track according to the scheduled speed
w	:	Difference between the running times at $v_{approach}$ and at $v_{operating}$
z	:	Difference between the clearing times at $v_{approach}$ and at $v_{operating}$

STTP symbols and variables

η	:	Number of time intervals of traffic prediction
$\eta\nu$:	Temporal length of the entire time horizon of traffic prediction
$\mathcal{G}^i(S^\emptyset)$:	Alternative graph of the i -th time interval
$\mathcal{G}^i(S)$:	Alternative graph solution of the i -th time interval
ν	:	Temporal length of a time interval
$[(i-1)\nu, i\nu]$:	i -th time interval
π_y^*	:	Scheduled entrance time (release time) of train T_y in the network plus the delay due to rolling stock re-use constraints
t_{T_y}	:	Starting time of the first operation of T_y in the next time interval
$t_{T_y}^*$:	Minimum time at which T_y can stop starting from t_{T_y}
θ_{yu}	:	Turning buffer time (rolling stock re-use constraint) between the two trains T_y and T_u
X	:	Set of alternative pairs constrained between all the consecutive time intervals ($X = \sum X_i$ with $i = 2, \dots, \eta$)
X_i	:	Set of alternative pairs constrained between $(i-1)$ -th and i -th time intervals
Z_i	:	List of trains running in both $[(i-2)\nu, (i-1)\nu]$ and $[(i-1)\nu, i\nu]$

Index

- Acceptable speed profile, 57
 - Alternative graph, 8, 68, 69
 - arc, 70
 - alternative, 69
 - companion, 70
 - fixed, 69
 - forbidden, 70
 - implied, 82
 - redundant, 82
 - selected, 70
 - cycle, 72
 - decision variable, 70, 78
 - machine, 69
 - makespan, 70, 73
 - modified due date, 73
 - node, 69
 - critical, 114
 - dummy, 69
 - finish, 69
 - head, 70
 - hindering, 114
 - isolated, 69
 - start, 69
 - tail, 70
 - waiting, 114
 - pair
 - selected, 70
 - unselected, 70
 - paired arcs, 70
 - precedence relation, 70
 - ramification, 109
 - backward, 114
 - forward, 114
 - selection, 70
 - complete, 70
 - consistent, 70
 - extension, 70
 - initial, 72
 - set
 - ascendant, 83
 - BRCP, 114
 - critical path, 70, 78, 108
 - descendant, 83
 - FBRCP, 114
 - ramified critical path, 109
 - solution, 8, 70
 - Approaching time, 135
 - Automatic route setting, 2, 4
 - Automatic train protection, 54, 146

 - Block section, 54
 - Blocked track, 61
 - Blocking time, 57
 - graph, 14
 - overlap, 57, 136
 - Branch and bound, 88
 - bounding, 88
 - branching, 88
 - branching rule, 90
 - lower bound, 88
 - near-optimum, 88
 - proven optimum, 88
 - search strategy
 - best first, 91
 - breadth first, 91
 - depth first, 91
 - hybrid, 91
 - worst first, 91
 - upper bound, 90
 - Buffer time, 14, 55, 134

 - CDR
 - problem, 67, 69
-

- solution, 78
- CDRFR
 - problem, 69
 - solution, 72
- Clearing time, 135
- Conflict, 56, 62, 136
- Conflict detection, 37, 62
- Conflict resolution, 37, 62
- Connected train service, 16, 56
- Constraint
 - blocking, *see* no-store
 - no-store, 68, 71, 112
 - no-swap, 68
 - no-wait, 77
 - perishability, 77
 - precedence, 62
- Critical track section, 14
- Data loading, 57, 59
 - off-line data, 59
 - real-time data, 59
- Deadlock, 25, 56, 62, 115
- Default routing, 61
- Delay
 - consecutive, *see* knock-on, 73
 - entrance, 56
 - initial, 2, 56, 73
 - knock-on, 2, 56
 - primary, *see* initial
 - secondary, *see* knock-on
 - total, 56, 73
- Delay propagation, 56
- Disjunctive graph, 69
- Dispatching heuristic, 63, 86
 - ARI, 86
 - FCFS, 86
 - FLFS, 86
 - PRULE, 86
- Dispatching rule, *see* Dispatching heuristic
- Disruption recovery, 57, 61
- Dwell time, 56
- Dwell time margins, 14
- Early arrival, 142
- Efficient algorithm, 63
- Emergency timetable, 61
- Entrance perturbation, 56
- Feasibility check, 64, 134
- Feasible timing, 71
- Fixed arrival time, 74
- Fixed block signaling, 54
- Flexible departure time, 74
- Free-net running, 141
- Global conflict resolution, 41, 68, 160
- Graph updating, 139
- Greedy heuristic, 87
 - AMCC, 87
- Green wave strategy, 77
- Hard instance, 99
- Immediate selection, *see* Implication rule
- Implication rule, 82, 90
 - dynamic, 82
 - static, 82, 83
- Incomplete algorithm, 110
- Individual train speed management, 44
- Jackson preemptive schedule, 85
- Job shop scheduling, 67
- Lexicographic comparison, 107
- Local search, 107, 110
- Local traffic control, 46
- Lower bound algorithm, 85
- Minimal time headway, 14, 134
- Minor perturbation, 36
- Move, 107
- Neighborhood, 110, 113
 - BRCP, 115
 - complete, 114
 - FBRCP, 115
 - opt-connected, 115
 - search strategy
 - complete, 116
 - hybrid, 116

- restart, 116
- New conflict-free plan, 68
- Non-scheduled waiting time, 14
- NP-complete, 24, 28, 40, 68
- NP-hard, 21
- Off-line timetabling, 1
- Operation, 55, 70
- Railway constraint
 - meeting, 76
 - passenger connection, 77
 - precedence, 76
 - rolling stock connection, 76
 - route booking, 77
- Railway dynamic traffic management, 49
 - principles, 50
- Railway network, 54
- Railway system, 1
 - capacity, 14
 - punctuality, 16
 - reliability, 15
 - robustness, 15, 49
 - stability, 15
 - throughput, 82
- Reaction time, 135
- Real-time traffic management, 3, 36
 - distributed traffic control, 47
 - fixed-speed model, 38, 40
 - network traffic control, 47
 - train rescheduling, 40
 - train rescheduling and local rerouting, 41
 - variable-speed model, 38, 44, 133
- Real-time traffic optimization, 58, 61
- Real-time train dispatching, 56, 57
 - drastic solution, 42
 - feasible solution, 57, 136
 - objective function, 8
- Recovery time, 14, 55
- Release time, 56
- Relevant point, 72
- Rerouting option, 60, 108
- Running time, 55, 71
- Running time reserves, *see* Recovery time
- Scheduled arrival time, 74
- Scheduled departure time, 74
- Scheduled waiting time, 14
- Scheduling strategy
 - iterative, 65, 141
 - single, 65, 141
- Setup time, 55, 71, 72
- Severe disturbance, 41
- Short-term traffic prediction, 12, 159
- Sight time, 135
- Signal, 54
 - green, 55
 - red, 54
 - yellow, 55
- Speed
 - approaching, 56, 137
 - maximum, 56
 - operating, 137
- Speed updating, 64, 137
- Stochastic approach, 31
 - delay propagation, 32
 - queueing, 31
 - simulation
 - asynchronous, 34
 - macroscopic, 35
 - microscopic, 34
 - synchronous, 34
- Switching time, 135
- Tabu search, 107, 112, 117
 - aspiration criterion, 112
 - incumbent solution neighborhood, 112
 - move, 113
 - tabu list, 112
- Temporal decomposition, 65, 160
- Time headway, 1, *see* Setup time
- Time limit, 117
- Timetable
 - cyclic, 18, 23
 - flexibility, 74
 - flexible, 81
 - periodicity, 23
 - rigid, 81
 - slack, 14

- symmetry, 23
- synchronization, 23
- Timetable disruption, 56
- Timetabling, *see* Off-line timetabling
- Traffic control center, 36
- Train
 - freight, 1, 37
 - intercity, 18
 - international, 37
 - interregional, 18
 - local, 18
 - national, 37
 - passenger, 1
- Train detection device, 16
- Train dispatching system, 4, 36, 37, 57
- Train route, 55, 70
 - passable, 55
 - timing, 55
- Train routing, 18
 - freight train routing, *see* network routing
 - line planning, 18
 - network routing, 18
 - railroad blocking problem, 19
 - routing and makeup model, 20
- Train routing and scheduling, 27
- Train run, *see* Train service
- Train scheduling, 21
 - dispatch planning, *see* operational scheduling
 - draft timetable, 25
 - master schedule, 21
 - operational scheduling, 21, 25
 - tactical scheduling, 21
- Train service, 55
- Train speed coordination, 38, 58, 64, 139
- Turn-around time, 56
- Turning buffer time, 162
- Variable neighborhood tabu search, 107, 117
- What-if scenarios, 63

Summary

This PhD thesis is principally concerned with the design, implementation and evaluation of an advanced and robust laboratory tool for supporting railway traffic controllers in the everyday task of managing timetable disturbances. To this end, we give a valuable contribution to the state-of-the-art of real-time train traffic management, including the following achievements:

- (i) An innovative model for railway traffic optimization is presented to predict accurately train traffic flows and to enable the computation of optimal network schedules, i.e., all trains are managed simultaneously in a railway network for a given time period.
- (ii) The development of fast and effective scheduling algorithms based on the proposed model for the real-time management of a complex railway network is addressed. The objectives are to predict the evolution of train traffic within short computation times and to improve the punctuality by pro-actively detecting and solving train conflicts.
- (iii) A better use of rail capacity and a further improvement of punctuality is achieved by an iterative adjustment of train orders and routes in case of disturbances. Novel problem dedicated algorithms highlight the potential use of rerouting instead of only rescheduling the trains in order to limit the delay propagation as much as possible.
- (iv) Constructive algorithms for the dynamic modification of running times are provided that satisfy the timetable constraints of train orders and routes and guarantee the real-time feasibility of the running time profile, while respecting the signaling and safety systems in use.
- (v) A temporal decomposition method is introduced for the short-term traffic planning and control over a time period of up to several hours. This approach is of interest for traffic controllers since delays propagate considerably in time and space during heavily perturbed operations.
- (vi) A large set of computational studies on real-world instances proves that the automated decision support tool provides better solutions in terms of delay minimization compared to dispatching rules adopted by traffic controllers. Test beds are the hourly timetables of the Schiphol railway bottleneck and of the dispatching area between Utrecht and Den Bosch stations. We study practical size instances and different

types of disturbances, including multiple delayed trains, dwell time perturbations and blockage of some tracks.

The railway community has been concerned increasingly with the development of optimization techniques for off-line and real-time traffic management. Chapter 2 provides an overview of the state-of-knowledge in this research field. First, recent contributions on timetable analysis are described and various models for timetable design are compared and classified with regard to train scheduling and routing problems. Second, an overview of stochastic models is presented. Third, the management of railway traffic is approached from a real-time perspective. Models for the resolution of conflicts and the coordination of train speeds are characterized on the basis of different dispatching rules, i.e., changing train orders and/or routes, canceling of connections and/or train services and variation of train speeds. The traffic prediction period and the network size are considered important factors to evaluate the applicability and validity of the reviewed models. We then introduce the new concept of railway dynamic traffic management inspired by the Dutch infrastructure manager *ProRail*. The limitations of the current state-of-the-art are finally discussed.

Chapter 3 describes the implementation of our advanced real-time train dispatching support system, called ROMA (*Railway traffic Optimization by Means of Alternative graphs*). This is a computerized system that can assist train operators in their tasks, provides optimal train routes, orders and dynamic speed targets, and may prevent the decision maker from taking less effective decisions, such as causing a deadlock situation or unsatisfactory throughput.

ROMA is able to estimate and control the future evolution of the railway traffic considering actual track occupation, varying signaling and safety constraints, as well as dynamic train characteristics. The resulting solution represents a new feasible plan of arrival and departure times, minimizes train delays with respect to a disrupted timetable and is compatible to the real-time train positions and speeds.

ROMA recovers from disturbances automatically, with the objective of minimizing the propagation of consecutive delays, and improves the dynamic use of railway infrastructure. When train operations are perturbed, a new timetable of feasible arrival and departure times is re-computed, which is a conflict-free and deadlock-free schedule. Specifically, a *traffic optimization procedure* considers train reordering and local rerouting actions to solve the real-time conflict detection and resolution problem, while a *train speed coordination procedure* takes into account the variability of train dynamics required to satisfy minimum distance headways between consecutive trains in case of conflicts.

ROMA enables the optimization of railway traffic also if the actual timetable was not conflict-free and/or deadlock-free. This provides an efficient tool for real-time traffic management in case of severe traffic disturbances, such as when emergency timetables are required and dispatchers need extensive support to solve conflicts.

In Chapter 4, we make use of a detailed discrete optimization model, called *alternative graph*, in order to optimize the railway traffic when train operations are perturbed. This graph models all possible scheduling alternatives for a given set of train routes, and is

able to efficiently treat the *no-store* aspect of the train scheduling problem due to trains traveling in the same or opposite directions. The no-store constraints of the optimization model require that a train, having reached the end of a first track segment, cannot enter the subsequent segment if the latter is still occupied by another train, thus also preventing other trains from entering the first segment. Special *alternative arcs* represent the available operational choices such as the train order at a crossing or merging section. A decision is made by selecting one of two alternative arcs which then fixes a precedence constraint between two trains at a potential conflict point. Different signaling systems can be modeled with this graph that offers a high reliability and flexibility of conflict resolution. In case of fixed block signaling, each block signal corresponds to a node in the graph and the arcs between the nodes represent the blocking times and headway times respectively. A feasible schedule assigns passage times to each node such that all precedence constraints are satisfied. The objective function is the minimization of the consecutive delays for all trains at all visited stations and relevant points of the network.

The main value of the alternative graph is the strict consistency of the structure and the level of detail requested that can be included in the model. In fact, this graph incorporates a description of the network topology at the level of railway signal aspects and operational rules. Moreover, it can easily include other constraints relevant to the railway practice, without making the formulation more complex, such as:

- (i) *Flexible arrival/departure times at scheduled stops*: A train may leave a station within a time window of [minimum, maximum] departure times in order to improve the reliability of operations, without decreasing the capacity of the lines. This can be formulated as an additional length for special arcs representing the objective function in the graph modeling.
- (ii) *Route booking for trains approaching a station or a corridor*: An interesting strategy for advanced traffic management consists in letting the circulating trains run at their scheduled speed and wait only at scheduled stops (green-wave). This can be modeled as an additional constraint (*no-wait* constraint) in the alternative graph.
- (iii) *Minimum required time for passenger and rolling stock connections*: These important operational constraints of passenger satisfaction and operations performance are formulated as additional arcs in the graph.

In Chapter 5, we use an alternative graph formulation of the conflict detection and resolution problem with fixed train speed profiles. In the scheduling optimization phase, we therefore compute train sequences over given train routes. Sophisticated scheduling algorithms are provided to improve the punctuality compared to rule-based train dispatching. Basically, two classes of scheduling algorithms are implemented: rule-based dispatching (e.g. first come first served or various types of train priorities) and innovative global conflict resolution methods based on alternative graph properties (i.e., greedy heuristics, a lower bound and an exact method).

A major contribution to solving efficiently and timely the train scheduling problem consists in a *branch and bound algorithm* with fixed train routing for finding a new feasible schedule, given a perturbed timetable. The algorithm computes an optimal schedule using dynamic implications and several speed-ups based on the infrastructure topology. Computational experiments demonstrate that a truncated version of the algorithm provides proven optimal or near-optimal solutions within a limited computation time.

The branch and bound algorithm is also used to compare near-optimal solutions when varying the departure times of trains at the scheduled stops. Extensive computational results show that flexible departure times may offer more freedom to solve train conflicts and increase the punctuality without decreasing the throughput.

Network traffic controllers do not frequently change the scheduled standard routes, although it may enable a better use of existing infrastructure capacity during incidents. This is mainly due to the complexity of dispatching train operations in railway yards after major routing modifications. In Chapter 6, the compound problem of routing and sequencing trains is approached iteratively by computing an optimal train sequencing for given train routes, and then improving this solution by locally rerouting some trains. We investigate the mathematical properties of the compound problem, make use of our advanced algorithms for sequencing train movements, and introduce a *local search algorithm* for rerouting optimization purposes. The latter approach generated further relevant questions. The first concerns the extent at which different neighborhoods or more sophisticated search schemes might improve upon the local minima found by the local search algorithm. The second question is how to achieve further algorithmic improvements, in terms of an increase of the solution quality and a reduction of the computation time.

An innovative rerouting approach is therefore proposed to determine the best routes and the corresponding train sequences in corridors and stations. The effectiveness of extensive rerouting strategies is explored by incorporating the search for new routes in a *tabu search algorithm*, aiming to escape from local minima. We investigate the effectiveness of different neighborhood structures and evaluate the benefits of local rerouting strategies to minimize the delays between consecutive trains. A comparison with advanced rescheduling solutions shows the high potential of iterative rerouting and rescheduling strategies to minimize train delays and to improve the use of the infrastructure capacity. The most effective rerouting algorithm is the tabu search, which is able to improve the solutions computed by the branch and bound algorithm in terms of both maximum and consecutive delays.

In Chapter 7, we ascertain whether a safe space headway between trains is respected and address speed coordination issues among consecutive trains. By combining the blocking time theory of railway operations with the alternative graph approach the estimated track occupation times comply fully with the constraints of the signaling and safety system and assure feasible train trajectories in absence of disturbances. However, the alternative graph model assumes deterministic blocking and waiting times and, thus, does not take the impact of deceleration and acceleration into account in case of hindrance. We present a train speed coordination procedure that updates the speed profiles of trains according to the actual signal aspects. The train speed coordination procedure is based on single or iterative schedul-

ing strategies that identify pro-actively potential train conflicts, reschedule train traffic at a network scale and adjust the train speed profiles accordingly. The iterative scheduling procedure is performed to improve the solution quality, while computing an acceptable speed profile for each running train. After a finite number of iterations, new train speed profiles are obtained that comply with the existing signaling system and rolling stock characteristics.

A constructive algorithm for the computation of energy-efficient running times is also presented. Optimal speed profiles are designed which fit better to the new train order and cause less delays and energy consumption at the same time. Inserting traffic optimization algorithms in the variable-speed model allows an effective dynamic control of trains when perturbations and disruptions occur in the given time horizon of traffic prediction. A feasible schedule is thus generated that reduces consecutive delays and ensures minimal safety distances between any pair of trains while respecting the operational constraints.

The computational results on the two dispatching areas (i.e., Schiphol and Utrecht - Den Bosch) demonstrate a considerable difference in the schedules obtained by the fixed-speed and variable-speed models, which underline the need of considering a dynamic computation of train speed profiles. In other words, the fixed-speed model underestimates the consequences of braking and re-acceleration, while the variable-speed model presents much more realistic solutions. Moreover, an extensive comparison between the variable-speed model solutions computed by advanced scheduling algorithms and simple dispatching rules proves the efficiency of the dispatching support system developed.

The short-term effects of different levels of timetable disturbances up to several hours are evaluated in Chapter 8. A *temporal decomposition procedure* is developed to enable the computation of effective dispatching solutions in a rather short time. This is of special interest in case of large timetable disturbances due e.g. to technical failures or extremely adverse weather conditions, which affect the level of service severely and are too difficult to solve manually by dispatchers, while an automated tool may be used to forecast better the propagation of the actual delays over a railway network.

We decompose a large time period into tractable intervals solved in cascade and we adopt advanced scheduling algorithms to pro-actively detect and globally solve the train conflicts in each time interval. To this aim, we present further computational experiments on different ROMA configurations and traffic management instances, and demonstrate the ability of our dispatching support tool to reduce the delay propagation within a limited computation time.

The consequences of variations in the complexity and density of the railway network, the length of the time horizon of traffic prediction, the level of disturbance in the network and the delivery time for computing the real-time dispatching measures have also been assessed. From the results on the two dispatching areas the following conclusions can be drawn. Since the area around Schiphol is higher occupied and more complicated than the dispatching area between Utrecht and Den Bosch, ROMA needs more time to compute a dispatching solution. Furthermore, short-term traffic predictions on the less dense dispatching area shows that, for a real-time use of ROMA, the level of disturbance is a more time consuming factor than the time horizon length. These findings are more evident when dealing with

severe disturbances such as multiple delayed trains and infrastructure disruptions.

In Chapter 9, we resume the findings and contributions of this research and outline briefly how the ROMA dispatching support system might be integrated in an advanced dynamic traffic management system that could improve railway operations reliability for a larger network consisting of a number of control centers. We also recommend further ROMA extensions such as the estimation of the consequences of optimal train speed control and the analysis of several suitable dispatching measures while managing train traffic in highly occupied and complex railway networks.

To summarize, this thesis provides the innovative features of our laboratory tool, ROMA, and discusses the main requirements for its practical application.

Samenvatting

Dit proefschrift beschrijft voornamelijk het ontwerp, de implementatie en de evaluatie van een geavanceerd en robuust prototype hulpmiddel voor de ondersteuning van verkeersleiders bij de dagelijkse bijsturing van verstoringen in de dienstregeling. Hiermee geven we een waardevolle bijdrage aan de state-of-the-art in real-time railverkeersmanagement, inclusief de volgende prestaties:

- (i) Een innovatief model voor de optimalisering van railverkeer wordt voorgesteld waarmee het treinverkeer nauwkeurig kan worden voorspeld en optimale netwerkdienstregelingen kunnen worden berekend, i.e., alle treinen in een railnetwerk over een gegeven tijdsperiode worden gezamenlijk bekeken.
- (ii) De ontwikkeling van snelle en effectieve algoritmen gebaseerd op het voorgestelde model voor de real-time bijsturing van complexe railnetwerken wordt behandeld. De doelen zijn het voorspellen van de treinverkeersafwikkeling binnen korte rekentijd en het verbeteren van de punctualiteit door pro-actief detecteren en oplossen van treinconflicten.
- (iii) Een betere benutting van infrastructuurcapaciteit en een verdere verbetering van punctualiteit wordt bereikt door een iteratieve aanpassing van treinvolgordes en rijwegen in geval van verstoringen. Nieuwe probleemspecifieke algoritmen tonen de potentie van herrouteren aan in plaats van alleen volgordes aanpassen om vertragingvoortplanting zo veel mogelijk te voorkomen.
- (iv) Constructieve algoritmen worden gegeven voor het dynamisch wijzigen van rijtijden die voldoen aan de randvoorwaarden van de dienstregeling omtrent volgordes en rijwegen, haalbare operationele rijtijdprofielen garanderen, en ook de toegepaste veiligheids- en seinsystemen respecteren.
- (v) Een tijdsafhankelijke decompositiemethode wordt geïntroduceerd voor de korte-termijn planning en bijsturing over een periode tot aan enkele uren. Deze aanpak is interessant voor verkeersleiders omdat vertragingen aanzienlijk kunnen verspreiden over tijd en ruimte tijdens zwaar verstoorde dienstuitvoeringen.
- (vi) Een grote verzameling computereperimenten van realistische voorbeelden laat zien dat het automatische beslissingsondersteunende systeem betere oplossingen geeft wat

betreft de minimalisatie van vertragingen dan de gebruikte beslissingsregels van verkeersleiders. Testgebieden zijn de uurdienstregeling bij het spoorknelpunt Schiphol en het spoorwegverkeer tussen de stations Utrecht en Den Bosch. We bekijken praktische probleemgroottes en verschillende soorten verstoringen, inclusief meerdere vertraagde treinen, verstoorde halteertijden en versperringen van sommige sporen.

De spoorsector is steeds meer geïnteresseerd in de ontwikkeling van optimaliseringstechnieken voor off-line en real-time verkeersmanagement. Hoofdstuk 2 geeft een overzicht van de stand van kennis in dit onderzoeksveld. Allereerst worden recente bijdragen voor dienstregelingsanalyse beschreven en diverse modellen voor het ontwerp van dienstregelingen worden vergeleken en geclassificeerd in scheduling- en routingproblemen. Als tweede wordt een overzicht van stochastische modellen gepresenteerd. Ten derde wordt de beheersing van railverkeer benaderd vanuit een real-time perspectief. Modellen voor het oplossen van conflicten en de coördinatie van treinsnelheden worden gekarakteriseerd op basis van verschillende bijsturingsregels, d.i., wijziging van treinvolgordes en/of rijwegen, verbreken van aansluitingen, uitvallen van treindiensten en variatie van treinsnelheden. De verkeersvoorspellingshorizon en de netwerkgrrootte worden als belangrijke factoren beschouwd in de evaluatie van de toepasbaarheid en validiteit van de besproken modellen. We introduceren daarna het nieuwe concept van dynamisch railverkeersmanagement geïnspireerd door de Nederlandse infrastructuurmanager *ProRail*. Tenslotte worden de beperkingen van de state-of-the-art besproken.

Hoofdstuk 3 beschrijft de implementatie van ons geavanceerde real-time beslissingsondersteunend systeem voor de bijsturing, genaamd ROMA (*Railway traffic Optimization by Means of Alternative graphs*). Dit is een geautomatiseerd systeem dat verkeersleiders in hun taak kan assisteren, optimale rijwegen, volgordes en dynamische doelsnelheden geeft, en voorkomt dat de bijstuurder minder effectieve beslissingen neemt, zoals het veroorzaken van een deadlock-situatie of een onbevredigende doorstroming.

ROMA kan de toekomstige railverkeersafwikkeling schatten en sturen wat betreft de werkelijke spoorbezetting, de variërende beveiligings- en seinrandvoorwaarden, en de dynamische treinkarakteristieken. De resulterende oplossing representeert een nieuw conflictvrij plan van aankomst- en vertrektijden, minimaliseert de treinvertragingen met betrekking tot een verstoorde dienstregeling en sluit aan bij de real-time treinposities en snelheden. ROMA herstelt verstoringen automatisch met als doel het minimaliseren van de voortplanting van volgvertragingen en verbetert het dynamisch gebruik van de spoorinfrastructuur. Als treindiensten verstoord zijn wordt een nieuwe dienstregeling van uitvoerbare aankomst- en vertrektijden herberekend, die vrij is van conflicten en deadlocks. In het bijzonder beschouwt een *verkeersoptimalisatieprocedure* de herordening van treinen en lokale rijwegaanpassingen voor het oplossen van het real-time conflictdetectie en oplossingsprobleem, terwijl een *treinsnelheidscoördinatieprocedure* de variërende treindynamica in beschouwing neemt dat nodig is om aan de minimale opvolgafstanden van opeenvolgende treinen te voldoen bij conflicten.

ROMA kan ook het railverkeer optimaliseren als de actuele dienstregeling niet vrij is van conflicten en/of deadlocks. Dit geeft een efficiënt middel voor real-time verkeersmanage-

ment bij ernstige verkeersstoringen, zoals wanneer versperringsmaatregelen nodig zijn en verkeersleiders uitgebreide ondersteuning nodig hebben om conflicten op te lossen.

In hoofdstuk 4 maken we gebruik van een gedetailleerd discreet optimaliseringsmodel, *alternative graph* genoemd, om het railverkeer te optimaliseren als treindiensten verstoord zijn. Dit graafmodel modelleert alle mogelijke planningsalternatieven voor een gegeven verzameling treinrouteringen en kan het *no-store* aspect van het dienstregelingsprobleem voor treinen die in dezelfde of tegengestelde richting rijden efficiënt behandelen. De *no-store* randvoorwaarden in het optimaliseringsmodel eisen dat een trein die het einde van een blok nadert het volgende blok niet kan ingaan als deze nog bezet is door een andere trein, en voorkomt daarmee ook dat andere treinen het blok van de betreffende trein inkomen zolang die nog in het blok aanwezig is. Speciale *alternative arcs* (alternatieve pijlen) representeren de mogelijke operationele keuzes, zoals de treinvolgorde op een spoor kruising of spoor aansluiting. Een beslissing wordt gemaakt door één van twee alternatieve pijlen te selecteren waarmee een opvolgrelatie tussen twee treinen op een potentieel conflictpunt wordt vastgelegd. Verschillende seinsystemen kunnen met deze graaf worden gemodelleerd waardoor een hoge betrouwbaarheid en flexibiliteit van conflictoplossing wordt bereikt. Bij een vast-blok seinsysteem correspondeert ieder bloksein met een knoop in de graaf en de pijlen tussen de knopen representeren de bloktijden en opvolgtijden. Een conflictvrije dienstregeling wijst passagetijden toe aan iedere knoop zodanig dat aan alle opvolgrelaties is voldaan. De doelfunctie is de minimalisatie van de volgvertragingen voor alle treinen op alle bediende stations en relevante punten in het netwerk.

De voornaamste waarde van de alternatieve graaf is de strikte consistentie van de structuur en het verlangde detailniveau dat in het model kan worden opgenomen. Deze graaf bevat een beschrijving van de netwerktopologie op het niveau van seinen en operationele regels. Bovendien kan het makkelijk andere randvoorwaarden uit de praktijk bevatten zonder de formulering complexer te maken, zoals:

- (i) *Flexibele aankomst/vertrektijden bij geplande stops*: een trein mag van een station vertrekken binnen een tijdvenster van [minimum, maximum] vertrektijden om de betrouwbaarheid van de dienstuitvoering te verbeteren zonder de spoorcapaciteit te verminderen. Dit kan worden gemodelleerd als een extra gewicht van speciale pijlen die de doelfunctie in het graafmodel representeren.
- (ii) *Rijwegreservering voor treinen die een station of baanvak naderen*: een interessante strategie voor geavanceerd verkeersmanagement bestaat er uit dat treinen rijden volgens hun dienstregelingsnelheid en alleen wachten bij geplande stops (groene golf). Dit kan worden gemodelleerd als een extra randvoorwaarde (*no-wait* of niet-wachten voorwaarde) in de alternatieve graaf.
- (iii) *Minimaal benodigde tijd voor reiziger- en materieelaansluitingen*: deze belangrijke operationele voorwaarde voor reizigerstevredenheid en operationele prestaties worden ook geformuleerd als extra pijlen in de graaf.

In Hoofdstuk 5 gebruiken we de alternatieve graaf formulering van het conflictdetectie en -oplossingsprobleem met vaste snelheidsprofielen. In de dienstregelingsoptimaliseringsfase berekenen we daarvoor de treinvolgordes over de gegeven routeringen. Verfijnde dienstregelingsalgoritmen worden gegeven om de punctualiteit te verbeteren ten opzichte van beslissingsregel-gebaseerde bijsturing. Twee klassen algoritmen worden geïmplementeerd: beslissingsregels (e.g. first-come-first-served of verschillende typen treinprioriteiten) en innovatieve netwerk conflictoplossingsmethoden gebaseerd op eigenschappen van de alternatieve graaf (d.i., greedy heuristieken, een ondergrens en een exacte methode).

Een belangrijke bijdrage in het efficiënt en tijdig oplossen van het dienstregelingsprobleem bestaat uit een *branch-and-bound algoritme* met vaste treinroutering voor het vinden van een nieuw conflictvrij plan bij een gegeven verstoorde dienstregeling. Het algoritme berekent een optimale dienstregeling gebruik makend van dynamische implicaties en verschillende versnellingen gebaseerd op de infrastructuurtopologie. Rekenexperimenten geven aan dat een afgekapte versie van het algoritme bewezen optimale of bijna-optimale oplossingen geeft binnen een beperkte rekentijd.

Het branch-and-bound algoritme wordt ook gebruikt voor het vergelijken van bijna-optimale oplossingen wanneer vertrektijden op geplande stops gevarieerd kunnen worden. Uitgebreide rekenresultaten laten zien dat flexibele vertrektijden meer vrijheid geven om treinconflicten op te lossen en de punctualiteit te verhogen zonder de doorstroming te verlagen.

Treindienstleiders veranderen niet vaak de standaard rijwegen alhoewel dat bij incidenten een betere benutting van de bestaande infrastructuurcapaciteit kan geven. Dit wordt voornamelijk veroorzaakt door de complexiteit van de bijsturing op emplacementen na grote rijwegwijzigingen. In Hoofdstuk 6 wordt het samengestelde probleem van de bepaling van routeringen en volgordes van treinen iteratief aangepakt door het berekenen van een optimale volgorde van treinen bij gegeven routeringen, waarna deze oplossing wordt verbeterd door lokaal herrouteren van sommige treinen. We onderzoeken de wiskundige eigenschappen van het samengestelde probleem, maken gebruik van ons geavanceerde algoritme voor het bepalen van treinvolgordes, en introduceren een *lokaal zoekalgoritme* voor de beoogde optimale herroutering. Deze laatste aanpak geeft verdere relevante vragen. Allereerst de vraag in hoeverre verschillende definities van omgevingen of verfijndere zoekmethoden de gevonden lokale minima kunnen verbeteren. De tweede vraag is hoe verdere algoritmische verbeteringen kunnen worden bereikt in termen van oplossingskwaliteit en verkorten van rekentijd.

Een innovatieve herrouteringsaanpak wordt daarom voorgesteld om de beste routeringen en corresponderende treinvolgordes te bepalen binnen corridors en emplacementen. De effectiviteit van uitgebreide herrouteringstrategieën wordt uitgebuit door het zoeken van nieuwe routeringen in te bedden in een *tabu search algoritme* om uit lokale minima te ontsnappen. We onderzoeken de effectiviteit van verschillende omgevingsstructuren en evalueren de voordelen van lokale herrouteringstrategieën voor de minimalisering van volgvertragingen. Een vergelijking met geavanceerde herplanningsoplossingen laat de grote potentie zien van het iteratief bepalen van routeringen en treinvolgordes ter minimalisatie van treinvertragingen en verbetering van de infrastructuurbenutting. Het meest effectieve herrouteringsalgo-

ritme is tabu search, dat de oplossingen berekend met het branch-and-bound algoritme kan verbeteren in termen van zowel de maximum vertraging als de volgvertraging.

In Hoofdstuk 7 gaan we na of een veilige opvolgafstand tussen treinen wordt gerespecteerd en behandelen we de kwestie van snelheidscoördinatie tussen opeenvolgende treinen. Door het combineren van de bloktijdtheorie met de alternatieve graaf aanpak voldoen de geschatte spoorbezettingstijden volledig aan de randvoorwaarden van het veiligheids- en seinsysteem en zijn haalbare treinpaden gegarandeerd als er geen verstoringen zijn. De alternatieve graaf gaat echter uit van deterministische blok- en wachttijden en verwaarloost daarmee de impact van afremmen en optrekken bij verstoringen. We geven daarom een snelheidscoördinatieprocedure die de snelheidsprofielen bijwerkt ten opzichte van de actuele seinstanden. De snelheidscoördinatieprocedure is gebaseerd op een enkele of iteratieve herplanningstrategie die pro-actief de potentiële conflicten identificeert, het railverkeer op netwerkschaal optimaliseert en de snelheidsprofielen overeenkomstig aanpast. De iteratieve herplanningsprocedure wordt toegepast om de kwaliteit van de oplossing te verbeteren bij de berekende haalbare snelheidsprofielen voor iedere trein. Na een eindig aantal iteraties worden nieuwe snelheidsprofielen verkregen die overeenkomen met het seinsysteem en de materieelkarakteristieken.

Een constructief algoritme voor de berekening van energiezuinige rijtijden wordt ook gepresenteerd. Optimale snelheidsprofielen worden ontworpen die beter passen bij de nieuwe treinvolgordes en tegelijkertijd zorgen voor minder vertragingen en lager energiegebruik. Toevoeging van verkeersoptimalisatiealgoritmen in het variabele-snelheid model maakt een effectieve dynamische sturing van treinen mogelijk als zich verstoringen voordoen in de gegeven tijdshorizon van de verkeersprognose. Zo wordt een conflictvrije dienstregeling gegenereerd die volgvertragingen reduceert en minimum volgafstanden tussen ieder paar treinen verzekert terwijl aan de operationele randvoorwaarden worden voldaan.

De rekenresultaten in de twee treindienstleidingsgebieden (d.i., Schiphol en Utrecht-'s Hertogenbosch) laten een enorm verschil zien in de dienstregelingen verkregen door het vaste-snelheid en het variabele-snelheid model, wat het belang van een dynamische berekening van snelheidsprofielen onderstreept. Met andere woorden, het vaste-snelheid model onderschat de consequenties van remmen en opnieuw optrekken, terwijl het variabele-snelheid model veel realistischere oplossingen geeft. Bovendien bewijst een uitgebreide vergelijking tussen de oplossingen van het variabele-snelheid model berekend met geavanceerde algoritmen en die van simpele beslissingsregels de efficiëntie van het ontwikkelde beslissingsondersteunende bijsturingssysteem.

De korte-termijn effecten van verschillende niveaus van dienstregelingsverstoringen tot op enkele uren worden in Hoofdstuk 8 geëvalueerd. Een *tijdsafhankelijke decompositieprocedure* wordt ontwikkeld om effectieve bijsturingsoпlossingen in korte tijd te kunnen berekenen. Dit is in het bijzonder van belang bij grote dienstregelingsverstoringen door bijvoorbeeld technische storingen of extreem ongunstige weersomstandigheden, die het kwaliteit-sniveau sterk beïnvloeden. Dit is moeilijk handmatig door verkeersleiders op te lossen, terwijl een geautomatiseerd hulpmiddel gebruikt kan worden om de voortplanting van de actuele vertragingen over het railnetwerk beter te voorspellen.

We ontlede een lange tijdsperiode in handelbare intervallen die achter elkaar worden opgelost waarbij we geavanceerde dienstregelingsalgoritmen gebruiken om pro-actief conflicten te detecteren en netwerkbreed op te lossen binnen ieder interval. Hiervoor presenteren we nadere computerexperimenten met verschillende ROMA configuraties en verkeersmanagementvoorbeelden, en laten het vermogen zien van onze bijsturingstool om vertragingvoortplanting te reduceren binnen beperkte rekentijd.

De consequenties van variaties in de complexiteit en dichtheid van het spoornetwerk, de lengte van de tijdshorizon voor de verkeersprognose, de mate van verstoring in het netwerk en de rekentijd voor het berekenen van de real-time bijsturingsmaatregelen zijn ook beoordeeld. Van de resultaten uit de twee dienstregelingsgebieden kunnen we de volgende conclusies trekken. Omdat het gebied rond Schiphol zwaarder benut en gecompliceerder is dan het gebied tussen Utrecht en 's Hertogenbosch heeft ROMA daar meer rekentijd nodig om een bijsturingsoptimalisatie te vinden. Daarnaast laten de korte-termijn verkeersvoorspellingen op het minder zwaar benutte treindienstleidersgebied zien dat voor een real-time gebruik van ROMA de mate van verstoringen een belangrijkere factor is op rekentijd dan de lengte van de tijdshorizon. Deze bevindingen zijn duidelijker bij sterke verstoringen zoals meervoudig vertraagde treinen en infrastructuurstoringen.

In Hoofdstuk 9 vatten we de bevindingen en bijdragen van dit onderzoek samen en laten kort zien hoe het ROMA beslissingsondersteunend bijsturingssysteem geïntegreerd kan worden in een geavanceerd verkeersmanagementsysteem dat de betrouwbaarheid van de treindienstuitvoering kan verbeteren voor een groter netwerk bestaande uit een aantal verkeersleidingcentra. We raden ook verdere uitbreidingen op ROMA aan, zoals het schatten van de consequenties van optimale snelheidsturing van treinen en het analyseren van enkele geschikte bijsturingsmaatregelen voor het beheersen van treinverkeer in zwaar benutte en complexe spoornetwerken.

Samenvattend levert dit proefschrift de innovatieve onderdelen van onze prototype tool ROMA en bespreken we de belangrijkste eisen voor praktische toepassing hiervan.

About the author

Andrea D'Ariano was born 1979 in Rome, Italy. He got a bachelor in Computer Science Engineering and a master in Automation and Management Engineering both at "Università degli Studi Roma Tre". His master thesis, entitled "Job Shop Scheduling for Railway Traffic Management", was conducted under the supervision of prof. D. Pacciarelli (Roma Tre) and supported by a University Scholarship, ProRail and the European project COMBINE2.

In November 2003, he joined the Department of Transport and Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology. During his Ph.D. studies, he was a part-time (visiting) researcher at the operations research group of prof. D. Pacciarelli. At the beginning of 2004, he was awarded the Professional Italian Title of Engineer.

He is a member of TRAIL Research School, joined the TRAIL PhD Council in 2004 and was selected to represent PhD students for the renewal of the formal accreditation of TRAIL as a scientific research school by the Royal Netherlands Academy of Arts and Sciences.

His research publications were acknowledged by TRAIL Congresses, International Association of Railway Operations Research Seminars and INFORMS Railway Applications Section Annual Meetings as best papers, and by Railway Age Magazine and Who's Who in Science and Engineering. In addition to this dissertation, a number of articles (by him and co-authors) are published (or forthcoming) in peer reviewed well-known international scientific journals and conference proceedings. He also served as reviewer for journals as Transportation Research Part B and European Journal of Operational Research. His general research interests include operations research and its application to transportation.

TRAIL Thesis Series

A series of The Netherlands TRAIL Research School for theses on transport, infrastructure and logistics.

Nat, C.G.J.M., van der, *A Knowledge-based Concept Exploration Model for Submarine Design*, T99/1, March 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Westrenen, F.C., van, *The Maritime Pilot at Work: Evaluation and Use of a Time-to-boundary Model of Mental Workload in Human-machine Systems*, T99/2, May 1999, TRAIL Thesis Series, Eburon, The Netherlands

Veenstra, A.W., *Quantitative Analysis of Shipping Markets*, T99/3, April 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Minderhoud, M.M., *Supported Driving: Impacts on Motorway Traffic Flow*, T99/4, July 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Hoogendoorn, S.P., *Multiclass Continuum Modelling of Multilane Traffic Flow*, T99/5, September 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Hoedemaeker, M., *Driving with Intelligent Vehicles: Driving Behaviour with Adaptive Cruise Control and the Acceptance by Individual Drivers*, T99/6, November 1999, TRAIL Thesis Series, Delft University Press, The Netherlands

Marchau, V.A.W.J., *Technology Assessment of Automated Vehicle Guidance - Prospects for Automated Driving Implementation*, T2000/1, January 2000, TRAIL Thesis Series, Delft University Press, The Netherlands

Subiono, *On Classes of Min-max-plus Systems and their Applications*, T2000/2, June 2000, TRAIL Thesis Series, Delft University Press, The Netherlands

Meer, J.R., van, *Operational Control of Internal Transport*, T2000/5, September 2000, TRAIL Thesis Series, Delft University Press, The Netherlands

Bliemer, M.C.J., *Analytical Dynamic Traffic Assignment with Interacting User-Classes: Theoretical Advances and Applications using a Variational Inequality Approach*, T2001/1, January 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

Muילerman, G.J., *Time-based logistics: An analysis of the relevance, causes and impacts*, T2001/2, April 2001, TRAIL Thesis Series, Delft University Press, The Netherlands

- Roodbergen, K.J., *Layout and Routing Methods for Warehouses*, T2001/3, May 2001, TRAIL Thesis Series, The Netherlands
- Willems, J.K.C.A.S., *Bundeling van infrastructuur, theoretische en praktische waarde van een ruimtelijk inrichtingsconcept*, T2001/4, June 2001, TRAIL Thesis Series, Delft University Press, The Netherlands
- Binsbergen, A.J., van, J.G.S.N. Visser, *Innovation Steps towards Efficient Goods Distribution Systems for Urban Areas*, T2001/5, May 2001, TRAIL Thesis Series, Delft University Press, The Netherlands
- Rosmuller, N., *Safety analysis of Transport Corridors*, T2001/6, June 2001, TRAIL Thesis Series, Delft University Press, The Netherlands
- Schaafsma, A., *Dynamisch Railverkeersmanagement, besturingsconcept voor railverkeer op basis van het Lagenmodel Verkeer en Vervoer*, T2001/7, October 2001, TRAIL Thesis Series, Delft University Press, The Netherlands
- Bockstael-Blok, W., *Chains and Networks in Multimodal Passenger Transport. Exploring a design approach*, T2001/8, December 2001, TRAIL Thesis Series, Delft University Press, The Netherlands
- Wolters, M.J.J., *The Business of Modularity and the Modularity of Business*, T2002/1, February 2002, TRAIL Thesis Series, The Netherlands
- Vis, F.A., *Planning and Control Concepts for Material Handling Systems*, T2002/2, May 2002, TRAIL Thesis Series, The Netherlands
- Koppius, O.R., *Information Architecture and Electronic Market Performance*, T2002/3, May 2002, TRAIL Thesis Series, The Netherlands
- Veeneman, W.W., *Mind the Gap; Bridging Theories and Practice for the Organisation of Metropolitan Public Transport*, T2002/4, June 2002, TRAIL Thesis Series, Delft University Press, The Netherlands
- Nes, R., van, *Design of multimodal transport networks, a hierarchical approach*, T2002/5, September 2002, TRAIL Thesis Series, Delft University Press, The Netherlands
- Pol, P.M.J., *A Renaissance of Stations, Railways and Cities, Economic Effects, Development Strategies and Organisational Issues of European High-Speed-Train Stations*, T2002/6, October 2002, TRAIL Thesis Series, Delft University Press, The Netherlands
- Runhaar, H., *Freight transport: at any price? Effects of transport costs on book and newspaper supply chains in the Netherlands*, T2002/7, December 2002, TRAIL Thesis Series, Delft University Press, The Netherlands
- Spek, S.C., van der, *Connectors. The Way beyond Transferring*, T2003/1, February 2003, TRAIL Thesis Series, Delft University Press, The Netherlands
- Lindeijer, D.G., *Controlling Automated Traffic Agents*, T2003/2, February 2003, TRAIL Thesis Series, Eburon, The Netherlands

Riet, O.A.W.T., van de, *Policy Analysis in Multi-Actor Policy Settings. Navigating Between Negotiated Nonsense and Useless Knowledge*, T2003/3, March 2003, TRAIL Thesis Series, Eburon, The Netherlands

Reeven, P.A., van, *Competition in Scheduled Transport*, T2003/4, April 2003, TRAIL Thesis Series, Eburon, The Netherlands

Peeters, L.W.P., *Cyclic Railway Timetable Optimization*, T2003/5, June 2003, TRAIL Thesis Series, The Netherlands

Soto Y Koelemeijer, G., *On the behaviour of classes of min-max-plus systems*, T2003/6, September 2003, TRAIL Thesis Series, The Netherlands

Lindveld, Ch.D.R., *Dynamic O-D matrix estimation: a behavioural approach*, T2003/7, September 2003, TRAIL Thesis Series, Eburon, The Netherlands

Weerd, M.M., de, *Plan Merging in Multi-Agent Systems*, T2003/8, December 2003, TRAIL Thesis Series, The Netherlands

Langen, P.W. de, *The Performance of Seaport Clusters*, T2004/1, January 2004, TRAIL Thesis Series, The Netherlands

Hegy, A., *Model Predictive Control for Integrating Traffic Control Measures*, T2004/2, February 2004, TRAIL Thesis Series, The Netherlands

Lint, J.W.C., van, *Reliable Travel Time Prediction for Freeways*, T2004/3, June 2004, TRAIL Thesis Series, The Netherlands

Tabibi, M., *Design and Control of Automated Truck Traffic at Motorway Ramps*, T2004/4, July 2004, TRAIL Thesis Series, The Netherlands

Verduijn, T.M., *Dynamism in Supply Networks: Actor switching in a turbulent business environment*, T2004/5, September 2004, TRAIL Thesis Series, The Netherlands

Daamen, W., *Modelling Passenger Flows in Public Transport Facilities*, T2004/6, September 2004, TRAIL Thesis Series, The Netherlands

Zoetman, A., *Railway Design and Maintenance from a Life-Cycle Cost Perspective: A Decision-Support Approach*, T2004/7, November 2004, TRAIL Thesis Series, The Netherlands

Bos, D.M., *Changing Seats: A Behavioral Analysis of P&R Use*, T2004/8, November 2004, TRAIL Thesis Series, The Netherlands

Versteegt, C., *Holonic Control For Large Scale Automated Logistic Systems*, T2004/9, December 2004, TRAIL Thesis Series, The Netherlands

Wees, K.A.P.C., van, *Intelligente voertuigen, veiligheidsregulering en aansprakelijkheid. Een onderzoek naar juridische aspecten van Advanced Driver Assistance Systems in het wegverkeer*, T2004/10, December 2004, TRAIL Thesis Series, The Netherlands

Tampère, C.M.J., *Human-Kinetic Multiclass Traffic Flow Theory and Modelling: With Application to Advanced Driver Assistance Systems in Congestion*, T2004/11, December 2004, TRAIL Thesis Series, The Netherlands

Rooij, R.M., *The Mobile City. The planning and design of the Network City from a mobility point of view*, T2005/1, February 2005, TRAIL Thesis Series, The Netherlands

Le-Anh, T., *Intelligent Control of Vehicle-Based Internal Transport Systems*, T2005/2, April 2005, TRAIL Thesis Series, The Netherlands

Zuidgeest, M.H.P., *Sustainable Urban Transport Development: a Dynamic Optimization Approach*, T2005/3, April 2005, TRAIL Thesis Series, The Netherlands

Hoogendoorn-Lanser, S., *Modelling Travel Behavior in Multimodal Networks*, T2005/4, May 2005, TRAIL Thesis Series, The Netherlands

Dekker, S., *Port Investment - Towards an integrated planning of port capacity*, T2005/5, June 2005, TRAIL Thesis Series, The Netherlands

Koolstra, K., *Transport Infrastructure Slot Allocation*, T2005/6, June 2005, TRAIL Thesis Series, The Netherlands

Vromans, M.J.C.M., *Reliability of Railway Systems*, T2005/7, July 2005, TRAIL Thesis Series, The Netherlands

Oosten, W., *Ruimte voor een democratische rechtsstaat. Geschakelde sturing bij ruimtelijke investeringen*, T2005/8, September 2005, TRAIL Thesis Series, The Netherlands

Le-Duc, T., *Design and control of efficient order picking*, T2005/9, September 2005, TRAIL Thesis Series, The Netherlands

Goverde, R., *Punctuality of Railway Operation and Timetable Stability Analysis*, T2005/10, October 2005, TRAIL Thesis Series, The Netherlands

Kager, R., *Design and implementation of a method for the synthesis of travel diary data*, T2005/11, October 2005, TRAIL Thesis Series, The Netherlands

Boer, C., *Distributed Simulation in Industry*, T2005/12, November 2005, TRAIL Thesis Series, The Netherlands

Pielage, P., *Conceptual Design of Automated Freight Transport Systems*, T2005/14, November 2005, TRAIL Thesis Series, The Netherlands

Groothedde, B., *Collaborative Logistics and Transportation Networks, a modeling approach to network design*, T2005/15, November 2005, TRAIL Thesis Series, The Netherlands

Valk, J.M., *Coordination among Autonomous Planners*, T2005/16, December 2005, TRAIL Thesis Series, The Netherlands

Krogt, R.P.J., van der, *Plan Repair in Single-Agent and Multi-Agent Systems*, T2005/17, December 2005, TRAIL Thesis Series, The Netherlands

Bontekoning, Y.M., *Hub exchange operations in intermodal hub-and-spoke networks. A performance comparison of four types of rail-rail exchange facilities*, T2006/1, February 2006, TRAIL Thesis Series, The Netherlands

Lentink, R., *Algorithmic Decision Support for Shunt Planning*, T2006/2, February 2006, TRAIL Thesis Series, The Netherlands

Ngoduy, D., *Macroscopic Discontinuity Modeling for Multiclass Multilane Traffic Flow Operations*, T2006/3, April 2006, TRAIL Thesis Series, The Netherlands

Vanderschuren, M.J.W.A., *Intelligent Transport Systems for South Africa. Impact assessment through microscopic simulation in the South African context*, T2006/4, August 2006, TRAIL Thesis Series, The Netherlands

Ongkittikul, S., *Innovation and Regulatory Reform in Public Transport*, T2006/5, September 2006, TRAIL Thesis Series, The Netherlands

Yuan, J., *Stochastic Modelling of Train Delays and Delay Propagation in Stations*, T2006/6, October 2006, TRAIL Thesis Series, The Netherlands

Viti, F., *The Dynamics and the Uncertainty of Delays at Signals*, T2006/7, November 2006, TRAIL Thesis Series, The Netherlands

Huisken, G., *Inter-Urban Short-Term Traffic Congestion Prediction*, T2006/8, December 2006, TRAIL Thesis Series, The Netherlands

Feijter, R., de, *Controlling High Speed Automated Transport Network Operations*, T2006/9, December 2006, TRAIL Thesis Series, The Netherlands

Makoriwa, C., *Performance of Traffic Networks. A mosaic of measures*, T2006/10, December 2006, TRAIL Thesis Series, The Netherlands

Miska, M., *Microscopic Online Simulation for Real time Traffic Management*, T2007/1, January 2007, TRAIL Thesis Series, The Netherlands 2006

Chorus, C., *Traveler Response to Information*, T2007/2, February 2007, TRAIL Thesis Series, The Netherlands

Weijermars, W., *Analysis of Urban Traffic Patterns Using Clustering*, T2007/3, April 2007, TRAIL Thesis Series, The Netherlands

Zondag, B., *Joined modeling of land-use, transport and economy*, T2007/4, April 2007, TRAIL Thesis Series, The Netherlands

Bok, M., de, *Infrastructure and Firm Dynamics: A micro-simulation approach*, T2007/5, May 2007, TRAIL Thesis Series, The Netherlands

Fiorenzo Catalano, S., *Choice Set Generation in Multi-Modal Transportation Networks*, T2007/6, June 2007, TRAIL Thesis Series, The Netherlands

Vlist, P., van der, *Synchronizing the retail supply chain*, T2007/7, June 2007, TRAIL Thesis Series, The Netherlands

Joksimovic, D., *Dynamic Bi-level Optimal Toll Design Approach for Dynamic Traffic Networks*, T2007/8, September 2007, TRAIL Thesis Series, The Netherlands

Warffemius, P.M.J., *Modeling the Clustering of Distribution Centers around Amsterdam Airport Schiphol. Location Endowments, Economies of Agglomeration, Locked-in Logistics and Policy Implications*, T2007/9, September 2007, TRAIL Thesis Series, The Netherlands

Driel, C., van, *Driver Support in Congestion: an assessment of user needs and impacts on driver and traffic flow*, T2007/10, November 2007, TRAIL Thesis Series, The Netherlands

Gieteling, O., *Design and Validation of Advanced Driver Assistance Systems*, T2007/11, November 2007, TRAIL Thesis Series, The Netherlands

Nutall, A., *Design Aspects of Multiple Driven Belt Conveyors*, T2007/12, November 2007, TRAIL Thesis Series, The Netherlands

Nederveen, A.A.J., *Ruimtelijke Inpassing van Lijninfrastructuur. Een onderzoek naar de geschiktheid van inspraakreacties voor het ontwerpen van lijninfrastructuur*, T2007/13, December 2007, TRAIL Thesis Series, The Netherlands

Negenborn, R.R., *Multi-Agent Model Predictive Control with Applications to Power Networks*, T2007/14, December 2007, TRAIL Thesis Series, The Netherlands

Houtenbos, M., *Expecting the Unexpected. A study of interactive driving behaviour at intersections*, T2008/1, January 2008, TRAIL Thesis Series, The Netherlands

Argiolu, R., *Office Location Choice Behaviour and Intelligent Transport Systems*, T2008/2, January 2008, TRAIL Thesis Series, The Netherlands

Katwijk, R.T., van, *Multi-Agent Look-ahead Traffic Adaptive Control*, T2008/3, January 2008, TRAIL Thesis Series, The Netherlands

Hegeman, G., *Assisted Overtaking. An assessment of overtaking on two-lane rural roads*, T2008/4, February 2008, TRAIL Thesis Series, The Netherlands

Quak, H.J., *Sustainability of Urban Freight Transport. Retail Distribution and Local Regulations in Cities*, T2008/5, March 2008, TRAIL Thesis Series, The Netherlands

D'Ariano, A., *Improving Real-Time Train Dispatching: Models, Algorithms and Applications*, T2008/6, April 2008, TRAIL Thesis Series, The Netherlands

NGInfra Theses

Theses written in the framework of the Netherlands foundation “Next Generation Infrastructures” (for ordering: info@nginfra.nl).

Martijn Kuit, 2002, *Strategic behavior and regulatory styles in the Netherlands energy industry*, Delft University of Technology, The Netherlands.

Laurens de Vries, 2004, *Securing the public interest in electricity generation markets, The myths of the invisible hand and the copper plate*, Delft University of Technology, The Netherlands.

Fernando Kuipers, 2004, *Quality of Service Routing in the Internet: Theory, Complexity and Algorithms*, Delft University of Technology, The Netherlands.

Franois Boisseleau, 2004, *The role of power exchanges for the creation of a single European electricity market: market design and market regulation*, Delft University of Technology, The Netherlands, and University of Paris IX Dauphine, France.

Ewoud Verhoef, 2004, *The ecology of metals*, Delft University of Technology, The Netherlands.

Semir Daskapan, 2005, *MEDUSA, Survivable information security in critical infrastructures*, Delft University of Technology, The Netherlands.

Kaspar Koolstra, 2005, *Transport Infrastructure Slot Allocation*, TRAIL Research School, The Netherlands.

Ruben van Wendel de Joode, 2005, *Understanding open source communities: an organizational perspective*, Delft University of Technology, The Netherlands.

Viren Ajodhia, 2006, *Regulating beyond price, integrated price-quality regulation for electricity distribution networks*, Delft University of Technology, The Netherlands.

Mark de Bruijne, 2006, *Networked Reliability, Institutional fragmentation and the reliability of service provision in critical infrastructures*, Delft University of Technology, The Netherlands.

Andrew Barendse 2006, *Regional regulation as a new form of telecom sector governance: the interactions with technological socio-economic systems and market performance*, Delft University of Technology, The Netherlands.

Wolter Lemstra, 2006, *The Internet bubble - the impact on the development path of the telecommunications sector*, Delft University of Technology, The Netherlands.

Dusica Joksimovic, 2007, *Dynamic Bi-level Optimal Toll Design Approach for Dynamic Traffic Networks*, TRAIL Research School, The Netherlands.

Rudy Negenborn, 2007, *Multi-Agent Model Predictive Control with Applications to Power Networks*, TRAIL Research School, The Netherlands.

Machtelt Meijer, 2007, *Intertwining uncertainty analysis and decision-making about drinking water infrastructure*, Delft University of Technology, The Netherlands.

Richard Cawley, 2007, *The New EU Approach to Sector Regulation in the Network Infrastructure Industries*, Delft University of Technology, The Netherlands.

Hamilcar Knops, 2008, *A functional legal design for reliable electricity supply, How technology affects law*, Delft University of Technology, The Netherlands.

Andrea D'Ariano, 2008, *Improving Real-Time Train Dispatching: Models, Algorithms and Applications*, TRAIL Research School, The Netherlands.