Real-time Railway Traffic Management: dispatching in complex, large and busy railway networks

Francesco Corman

Real-time Railway Traffic Management: dispatching in complex, large and busy railway networks

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof.ir. K.C.A.M. Luyben,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op vrijdag 10 december 2010 om 12:30 uur
door Francesco CORMAN
ingegnere gestionale e dell'automazione
Roma Tre University
geboren te Rieti (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. M. Carey	University of Leeds, United Kingdom
Prof. dr. L. Kroon	Erasmus Universiteit Rotterdam
Prof. dr. D. Pacciarelli	Roma Tre University, Italy
Prof. S. Ricci	Sapienza University, Italy
Prof. Dr.-Ing. E. Wendler	RWTH Aachen University, Germany
Prof. dr. C. Witteveen	Technische Universiteit Delft
Prof. dr. ir. S.P. Hoogendoorn	Technische Universiteit Delft, reserve

TRAIL Thesis Series T2010/14, the Netherlands TRAIL Research School

Published and distributed by: Francesco Corman
E-mail: f.corman@tudelft.nl

ISBN 978-90-5584-133-2

Keywords: railway traffic, real-time rescheduling, distributed optimization

Copyright © 2010 by Francesco Corman

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 of the author.

Printed in the Netherlands

Preface

When I was graduating, I was asking myself what to do next. I have been spending 5 years of engineering study; 6 months for the thesis project, of which three months abroad, as a guest student at Delft University of Technology. I had a nice time there, I enjoyed my freedom, even though after so much studying, one thing was clear to me, that I was not interested in academic research. Being an engineer, I've been always sure of my ideas, and believing them in full; in this case, I instead made an exception, waiting a few months to come back and almost start a PhD in Rome, then working there as a researcher, and finally ending in Delft again as a PhD student.

I generally feel lucky for what I got, and those last 4 years, through their highs and lows, result in the same feeling. First of all, I really enjoyed a long period abroad to enlarge my horizons and understanding towards being a real citizen of Europe. Moreover, I had the opportunity to start an academic career in a top-level university. I was able to exploit the knowledge of the field of real-time railway rescheduling acquired for my thesis, combine my interests in science, mathematics, system theory, optimization, design and programming of software systems towards a concrete and useful goal, concerning things that I might eventually touch with my hands. A lot of satisfaction came from an amount of awards and acknowledgements received for my academic career, for my master thesis, and the academic results of my PhD. Despite I still believe that I am not working to honour and satisfy my pride, they made me believe that I was going on the right direction.

I owe many thanks to Andrea D'Ariano, which during the span of the years shifted of role from supervisor to colleague and very productive co-worker, always there to guide and motivate me, able to find the interesting part in works that I would not consider much, keeping a clear view of the objectives that was often missing to me, and for the nice and peaceful exchange of opinions about how to write ideas on paper.

Moreover, I must acknowledge the help of Professor Ingo A. Hansen, who let me free of performing my research while offering his critical approach to remember me to keep an eye on real operations and constraints that my simplistic and naive computer-science background would have happily avoided. Valuable help from a complementary point of view came from fruitful discussions on theoretical and methodological aspects with Professor Dario Pacciarelli. Many times I felt being an unstable and shaky bridge between the practical requirements of railway operations, and the ideas suggested by a pure mathematical and optimization approach.

During my research, I had contacts with various managers from the Dutch railway infrastructure manager ProRail, to direct my efforts to solve interesting problems, and to profit from their experience and comments; they also helped me with the large amount of data needed for my investigations; among them I am indebted the most to Dick Middelkoop, Leo Lodder, Lesley Valies. While discovering what the strange life of a PhD was about, I enjoyed sharing ideas, views and working experience with many other railway researchers, among which I include Rob Goverde from my same department, and a handful of researchers spread around Europe, including among many others Gabrio Caimi, Marco Luethi, Giorgio Medeossi, Marco Pranzo, Stefan Wegele.

Due to my peculiar situation of a computer scientist with mathematical optimization background, and knowing (still) nothing about fundamental diagrams, ramp metering, congestion shock-waves or other exotic things, I felt a lot of times being a different one in the department. And I must say, I really enjoyed this position among the Dutch (or behaving as such) community of the department; probably also because they would say that doe maar gewoon is al gek genoeg. This included for sure that strange habit of being eager to go and have lunch just few minutes after the opening of the canteen, to eat what I would not consider suitable for my needs, or to closely respect of the rule to show up in the office at 8.30, and leave at 17.30. Thanks to you all, especially to the latin room where I spent overall more than 4 years and I met the nicest people I could ask for.

If I felt at home in Delft, large thanks are due to the amount of nice friends I met in Delft, to whom I had the luck to be introduced since the time of Cornelio, and further grown without any control. I enjoyed so much the nice multicultural environment where one could spot the idiosyncrasies and habits of so many people put together, trying to understand each other and those strange Dutch. Countless things we have done together with all of you, and so many more I would like to do yet!

I'd like to remember also most of my friends at Roma Tre who opted for a PhD career and gave me the opportunity to exchange ideas about different fields with some common background, in a very straightforward and direct way. And I must thank, at least for their psychological support, the other friends of mine who got used to see me home only few weeks per year. Many thanks go of course to my family, who was not afraid of letting me making troubles in Europe, and getting closer to part of my roots. And to conclude, I am still convinced that the best things always happen by chance, and when you expect them the least. *Ad maiora!*

Contents

Preface **v**

1 Introduction **1**
1.1 General overview of railway transport 1
1.2 Motivation 4
1.3 Research questions and objectives 7
1.4 Scope and Outline 10
1.5 Main contributions 16
1.6 Practical relevance 22
1.7 Recommendations for future work 25
Bibliography 26

2 Article I - Railway dynamic traffic management **35**
2.1 Introduction 36
2.2 Problem description 39
2.3 Dynamic traffic management strategies 43
2.4 Decision support system 45
2.4.1 Load information 47
2.4.2 Disruption recovery 48
2.4.3 Real-time traffic optimization 49
2.4.4 Discussion 53
2.5 Computational experiments 54
2.5.1 Description of the test cases 54
2.5.2 Dynamic traffic management strategies 57
2.5.3 Effects of increasing perturbations 59
2.5.4 Effects of increasing disruptions 59
2.5.5 Effects of increasing time horizons 60
2.6 Conclusions 61
Bibliography 62

3 Article II - Reordering and rerouting in a complicated station area 67
3.1 Introduction . . . 68
3.2 Related literature . . . 70
3.3 Real-time train dispatching . . . 72
3.3.1 Problem description . . . 72
3.3.2 Problem formulation . . . 75
3.3.3 Decision support tool . . . 79
3.3.4 Interaction between decision support tool and dispatcher . . . 82
3.3.5 Real-time dynamic setting of the tool . . . 83
3.4 Test case . . . 85
3.4.1 Description of the instances . . . 85
3.4.2 Computational results . . . 87
3.5 Conclusions . . . 90
Bibliography . . . 90

4 Article III - Centralized versus distributed dispatching 95
4.1 Introduction . . . 96
4.2 Problem Definitions . . . 98
4.3 Overview of the Literature . . . 100
4.4 Centralized Rescheduling System . . . 104
4.4.1 Illustrative Example . . . 105
4.4.2 Train Scheduling Procedures . . . 107
4.5 Distributed Rescheduling System . . . 108
4.5.1 System Architecture . . . 108
4.5.2 Formulation with Alternative and Border Graphs . . . 109
4.5.3 Schedule Coordination Procedure . . . 114
4.6 Test Case . . . 117
4.6.1 Description of the Instances . . . 117
4.6.2 Computational Results . . . 119
4.7 Conclusions and Future Research . . . 124
4.8 Acknowledgements . . . 125
Bibliography . . . 125

5 Article IV - Dispatching and coordination 131
5.1 Introduction . . . 132
5.2 Review of the related literature . . . 135
5.3 Solution methods and procedures . . . 138
5.3.1 Disturbance handling process . . . 138
5.3.2 Traffic management architectures . . . 139
5.3.3 Dispatching model and procedures . . . 140
5.3.4 Coordination model and procedures . . . 143
5.3.5 Overview of the different techniques . . . 147

5.4 Computational experiments 148
5.4.1 Test case description 148
5.4.2 Timetable perturbations 150
5.4.3 Disrupted traffic situation 153
5.4.4 Discussion 157
5.5 Conclusions and future research 158
5.6 Acknowledgements 159
Bibliography 159

6 Article V - Evaluation of green wave policy 165
6.1 Introduction 166
6.2 Models and Algorithms 168
6.2.1 Illustrative example 170
6.2.2 Scheduling algorithms 173
6.2.3 Speed adjustment 174
6.3 Computational Experiments 174
6.3.1 Description of instances 174
6.3.2 Performance of algorithms and policies 176
6.3.3 Discussion 179
6.4 Conclusions and Future Research 180
6.5 Acknowledgements 181
Bibliography 181

Summary 185

Samenvatting 191

About the Author 197

TRAIL Thesis Series 199

Chapter 1

Introduction

1.1 General overview of railway transport

Railway is a very important mode of transportation for passengers and freight, due to its peculiar characteristics. Few other transportation modes combine dedicated infrastructure connecting point to point cities and places of interests, high operational speed, high reliability, cost effective operations, high energy efficiency and very high safety rate.

In Europe, the need for moving passengers and commodities increases steadily at a pace similar to that of the Gross Domestic Product, that is around 2% per year for the last years. However, the most of this increase is sustained by road and maritime traffic. In perspective, the share of railway traffic (i.e., the modal split) is actually decreasing, from 11% in 2000 to an expected 8% in 2020 concerning freight transport, and from 6% to an expected 5% concerning passenger transport. In comparison, the share of road traffic represents over 40% of freight and over 75% of passengers transport [11].

The slower growth of railways and the unfavourable modal split towards railways have been acknowledged by governments and transport authorities which are looking to counterbalance this trend in the long term, to decongest highways and urban roads, foster sustainable development, and decrease pollution levels in cities and globally.

Despite providing a good solution for commuting and transfer between cities, railway transport has a limited attractiveness to potential users, mainly due to perceived consequences of delays and unreliability towards the travel time ratio between train and other modes, like cars. In the European Communities, around the year 2000, around 57% of rail users said to be satisfied with the service provided; in Germany only 46% [1]. A key factor in order to gain higher share in passenger transportation demand is therefore a substantial improvement in the (perceived) attractiveness of railway services.

Key areas of improvement include providing a better service, decreasing or keeping low fares, increasing reliability and connectivity for both railway traffic and inter-modal connections [10]. In the Netherlands, the Government and the Ministry of Transport have the long-term goal of making train services quicker and more frequent, more comfortable for passengers and more reliable [35]. In fact, the quality of service of the Dutch railway system was relatively low around the year 2000, with less than 80% of trains arriving at major stations within 3 minutes of delay [37]. The current target set by the contract of service with the Government requires a punctuality level of at least 87%; four major transportation stakeholders operating in the Netherlands pointed out in 2003 a target level of service of a punctuality level of 95% to be reached within year 2015, and this even despite the expectation of a significant increase of train traffic volumes [51].

With the goal of exploiting existing railway potentials to provide reliable, frequent and integrated transport services, attractive to passengers and freight transportation, legal frameworks have been recently put into existence by the European Communities and governmental bodies. In particular, deregulation of the sector (cf. European Directive 2001/14/EC) separates the concerns of the railway infrastructure manager, that has the duty to make accessible, available and exploitable the railway infrastructure, and the train operating companies, which offer transport services to passengers and goods in a market environment. Similarly, compensation schemes are introduced, with the aim of explaining in a fair and non-discriminatory way the sources causing delay to traffic, that could be traced back either to negligence of the infrastructure manager, a train operating company, or some external cause. In this way, the interaction of the multiple competing actors must be unbundled and become transparent, and reasons of train delays and unreliable operations need to be spotted and dealt with [28].

In this deregulated framework, railway companies operate in a market environment to deliver the best service in terms of transport possibility, reliability and frequency of service, either to win market share or to fulfil contractual agreements with national and regional public authorities. On the other side, the railway infrastructure manager is a non-profit organization with the only objectives of guaranteeing maximum availability of infrastructure and boosting its utilization. Non-discriminatory access to the available rail capacity is granted in order to provide transport services to the end users, ensuring safety, regularity and punctuality of train operations by effective railway traffic control. Moreover, the infrastructure manager has to direct and manage investments in infrastructure funded by governments or public-private partnerships.

The railway infrastructure manager has a large share of responsibility in increasing the quality of the transport service offered, as the available capacity poses most limitations on possible improvements in performance and reliability of the railway system. Building or upgrading infrastructure (more tracks, or higher operating speeds), replacing switches and level crossings by flyovers and tunnels can increase significantly available capacity,

but at a high cost, and a long time to be built. More innovative solutions, like technological improvements, advanced and efficient planning, management and operational procedures to make a better use of existing infrastructure, are needed in order to accommodate expected growth of transport demand into an attractive and highly reliable service, while avoiding large investments in the current scenario of market incertitude and limited budget available [51].

The example of the timetable in the Netherlands shows that the potential benefits of innovation in the railway process are still great. Normally, railway operations follow a timetable, that is a carefully designed plan of traffic, defining well in advance routes, orders and passing times for all trains [7]. Railway capacity is assigned to the train operating companies to deliver their services satisfying customers' demand [57]. The use of advanced mathematical programming and operations research techniques in the timetabling process, and in the rolling stock and crew scheduling, lead to large benefits to the main railway operating company of the Netherlands, NS. The resulting yearly profit could be quantified in millions of Euro, achieved directly by cutting expenses or by increased passenger demand and punctuality bonuses for improved reliability of operations [41].

One of the main sources of unreliability of a railway system is that trains have limited possibility of overtaking along lines; they must follow each other's path sequentially, until an overtaking or passing track is reached. Failures of rolling stock, extended dwell times due to larger flow of passengers, speed restrictions due to infrastructure breakdowns or adverse weather, incidents with humans or animals, etc., result in initial delays against which little or nothing can be done. Due to the interaction of trains along open tracks and on interlocking areas at stations resulting in minimal separation time between train runs, such initial delays propagate widely as consecutive delays to other trains in the network. This domino effect reduces seriously the quality of the service offered.

While initial delays cannot be prevented themselves, running time supplements and buffer times are included into the timetable to reduce propagation of delays by absorbing minor disturbances. The underlying principle is to trade off some capacity to time reserves and waiting times [67], thereby increasing robustness [63, 64]. Anyway, no timetable can be made resistant enough against disturbances or disruptions that alter deeply circulations of train traffic, because building extra robustness in the timetable (in order to cope with situations of stronger delay) can only be done by increasing substantially time reserves, therefore decreasing at the same time capacity available for traffic. On the other hand, when a critical level of capacity exploitation is reached [38, 58], and limited buffer and recovery times are considered, train operations suffer from a very high sensitivity to perturbations, resulting in instability [37], i.e., even minimal disturbances of train traffic, like an extended dwell time, may put seriously at danger the adherence of the actual traffic to the expected plan, in the whole network.

An alternative strategy is to perform online adjustments during operations to minimize

the negative effects of the disturbances, and in more serious cases, to restore feasible operations of the railway system (i.e., avoid deadlocks, where some trains are stuck due to occupation and reservation constraints, and need to be pushed backwards to keep the rest of the traffic running). The problem solved by railway traffic control is termed train conflict detection and resolution (CDR), and consists in changing and adjusting running times, dwell times, departure times, train speeds as well as train orders and routes in real-time, with the goal to reduce the propagation of delays in the network by exploiting all the potential of improved operations provided by dynamic traffic management.

We next distinguish between various levels of control: dispatchers (or line dispatchers) are in charge of a single corridor or a limited area around a major station, i.e., a dispatching area. Their view on the traffic is limited to the area they manage, for which they decide in real-time train orders and setup train routes, authorising train movements; most of the work to physically setup a route and monitor traffic is nowadays automated by remote sensing and controlling and performed by automatic route setting systems. For this reason, the intervention of the dispatcher to update the traffic plan is normally limited to perturbed operations. On the other hand, at a higher level, network traffic controllers (or network dispatchers, or coordinators) are responsible for the traffic management over large networks (i.e., entire regional networks), supervising the complete traffic flow and coordinating at network scale the dispatching actions taken locally. While the choice of dispatching actions (such as retiming, reordering, rerouting trains) to solve the CDR problem is typically left to dispatchers, network traffic controllers are mainly interested in controlling the trains traversing multiple areas and in taking decisions at the borders between areas.

1.2 Motivation

Current dispatching operations show in general a reactive approach, i.e., they focus at correcting and restoring the problem once it has been experienced. Automatic route setting systems, like ARI in the Netherlands [5], manage in an automatic way train traffic running on time or with limited delay, by reserving and setting routes on the basis of the actual timetable orders and train describer data. For small deviations from the existing plan, passing orders of trains are computed by combining the timetable with simple local decision rules. No optimality or quality evaluation is attached to those dispatching actions computed. Due to the strict time limits for computing a new plan in case of disturbances, dispatchers usually follow the decision taken by the ARI system, and may additionally perform manually only a few modifications, such as adjustments of train routes, orders and speeds. To forecast the evolution of the network and manage traffic, dispatchers rely mostly on experience and local and current information, “on the spot” and “now”. The outcome may reflect simple dispatching rules, like the well known First-Come First-

Served, giving priority to the train that arrives first at a junction, or may aim at limiting the difference with the original plan; in case of more disrupted operations, what-if scenarios are considered, called emergency timetables.

In fact, the information available to dispatchers to monitor the situation are normally limited to train describers, keeping track of train positions, and no extensive flow of information is available between the traffic controller or the dispatcher and the train driver, apart from the signalling and route setting system, and a normal telephone channel between drivers and dispatchers. With no continuous flow of information available from the train, the dispatchers are asked to take decisions for perturbed traffic based on limited information at the coarse level of block sections and a rough delay indication.

This does not allow to have a sound and reliable forecast of the future consequences of the dispatching actions going to be applied, and results in a non-optimal process; moreover, the efficiency of the chosen measures is often unknown [40, 68]. The situation could be much better if decisions would be taken that are proactive (i.e., based on a reliable estimate of the future) and informed (based on detailed global information, and quantitatively evaluated).

Recently, inspired by similar dynamic planning approaches in supply chain, vehicle routing or aircraft scheduling [46], new concepts have been proposed in order to manage in a more efficient way uncertainty and stochasticity of railway operations. Dynamic railway traffic management aims at providing more reliable and flexible services, leaving less details fixed during the planning stage (e.g. to be addressed by robust timetabling providing conflict-free train paths) and more control decisions left to dynamic traffic management, to be carried out by dispatchers in real-time, depending on the actual unpredictable situation [54]. This approach requires effective traffic control procedures to fully exploit the enlarged degree of freedom during dispatching, so that it is possible to react optimally to unexpected events, by adapting the timetable to the current status of the rail network. Automated traffic control procedures to help the dispatchers have been recently investigated by many railway companies and academic research supports development of rescheduling of traffic.

Such systems all share a common architecture that is needed to represent current operations and result in improved control actions. A global view of the current traffic situation is required that monitors train positions and speeds over a large area by means of a continuous and precise feed of data from operations. The evolution of the network and train traffic in the near future is computed by simulating realistic train movements, in order to detect actual and expected train conflicts at the microscopic level of block sections. For each potential conflict, decisions are taken by considering local or global effects, resulting in a schedule of feasible train operations limiting delay propagation. Advanced systems are able to quantitatively assess at network level the impact of local choices. Factors that influence the system dynamics like driver behaviour and additional stochasticity

of operations (i.e., additional delays) can be dealt with by closing a control loop with the operations [44]. The iterative process of sensing, simulation, optimisation, choice and implementation of control measures must be faster than the dynamics of the system, leading to a maximum time in the order of tens of seconds to produce a completely new schedule for an actual situation of traffic running [48].

The challenge of the dispatching problem is to embed solution algorithms into conflict detection and resolution systems, that find quickly scheduling solutions for large problems, using proactively detailed information at microscopic level, to reduce quantitatively delay propagation at network scale. In fact, most available approaches either lack microscopic detail, a global view over large networks with complex stations and interconnected railway lines, and advanced optimisation algorithms able to deal with dense and complex railway bottlenecks.

When considering a large amount of train traffic running in a time unit in dense areas with conflicting routes, or large networks, it is difficult for dispatchers to find even approximate suboptimal solutions within a short time, and to understand in full their effects at the level of the dispatching area and beyond. On the other hand, it would be of little use to provide a solution to relatively easier problems that human dispatchers can already understand and solve in an effective way. Therefore the aim of a decision support system for dispatching is to solve the problem where and when it is more difficult and complex.

A first answer to the problem whether it is possible to set up advanced models and algorithms able to manage railway traffic in real-time was given in the PhD thesis of D'Ariano [29]. As it was an innovative combination of mathematical modelling and microscopic simulation of operations, a number of issues were not dealt with in detail, and further development was encouraged in order to increase knowledge and ease the application of (part of) the strategies and methods developed into real-life operations. Therefore, much focus has been given in pushing the limits of what advanced optimisation approaches could achieve, by investigating instances based on large, complex and busy railway networks, based on the work of D'Ariano. The PhD thesis here presented aims at filling existing gaps in this stream of research, as guided by the research objectives explained in the next Section. For a general overview of scheduling, rescheduling and real-time dispatching in the railway world, and to give an idea of the breadth, the interests of public and academic institutions, and the recent advancements achieved so far, the reader is referred to [2, 6, 29, 43] and to [38].

1.3 Research questions and objectives

This research project was setup to investigate the following research objectives, related to the introduction and development of real-time optimisation and control procedures to manage difficult instances featuring large network size, high complexity and additional degrees of freedom in railway operations. The research has been based on a laboratory rescheduling system; for some cases, modules were extended from previous work; in other cases, they needed to be newly designed to consider completely new aspects of the problem. The final setup with all the extensions and extra modules incrementally added into a comprehensive system, has been used to quantify the results by extensive experimental assessment.

In fact, the work here presented is linked almost seamlessly with the previous work of D'Ariano. His objective was to develop suitable models for investigating dynamic railway traffic management strategies. In particular, the following advanced strategies to manage traffic in real time were considered: rescheduling (i.e., solving in real-time conflicts at merging and crossing points), introducing flexibility in timetable design and operations (to exploit online recovery and buffer times and limit delays) and rerouting at bottlenecks and major stations (to assign in an optimal way available platform track and infrastructure capacity to traffic).

Once the model and algorithms were refined, the research presented in this thesis started by considering the general picture of railway rescheduling and assessing in detail the impact of dynamic railway traffic management measures. The main principles were evaluated so far separately and on a single case study [55], while an extensive simulation assessment of the benefits reachable by using jointly the various strategies was missing.

Research Objective 1: Accommodate proactive dynamic railway traffic management principles into operations and quantify their impact for the railway infrastructure manager.

Article I discusses this research question in detail, whereas the whole thesis will report on additional benefits of advanced control actions computed by mathematical models and algorithms. In fact, the introduction of such principles is going to improve operations' quality due to a larger degree of freedom exploitable online, i.e., the plan defined offline with no knowledge of the actual situation could be adapted to deal with real-time perturbed operations. Anyway, those improvements depend heavily on the performance of the algorithms used which optimise train traffic with regard to the degrees of freedom considered (scheduling, rerouting, flexibility). When no advanced algorithm is supplied able to exploit global information, much potential for improvements remains unattainable, due to the complexity of the dispatching problem [54, 61].

In fact, algorithmic advancements should complement enlarged degrees of freedom al-

lowed during dispatching. Therefore, an innovative study was carried out in order to fully understand the potential of advanced scheduling and rerouting algorithms when busy and large stations with many platforms bring about the issue of dense timetables, in connection with large amounts of inbound and outbound train traffic of various speeds and types, scheduled over many routes that cross and hinder each other in complex interlocking areas [8]. Very little work so far focused on optimisation of train traffic in complex station areas, even though extra focus on these areas is important, as they are bottlenecks of the railway system. Due to the existing physical capacity and common rules for safe operations, most conflicts are to be expected in stations; and they even remain the bottlenecks of operations after the introduction of advanced traffic management systems [38].

Moreover, microscopic detail is required to model properly railway operations, because feasible train movements (i.e., compliant with the safety system and the movement authority given to trains) can be checked only at the level of individual block sections, for instance by the blocking time theory (see e.g. [38]). The obvious drawback of a microscopic approach is that large amounts of data are required, while the computational complexity of the scheduling problem increases exponentially with the amount of elementary decisions to be taken about orders and routes. Detailed modelling of train movements in interlocking areas poses additional problems to dispatching systems [53]. An innovative extension of the alternative graph model, proposed first in [47] and used throughout the thesis of D'Ariano to model train traffic was required, as well as various algorithmic improvements, as reported in Article II, to address the following:

Research Objective 2: Manage in real-time complex railway interlocking areas around large stations with dense scheduled traffic.

In busy railway networks, due to connections between train services, and conflicts between trains on the available capacity, delays can propagate easily in areas as large as a whole national network, before fading out due to recovery times [37]. Therefore, optimising for one railway junction without having a global view of the impacts may lead to adverse suboptimal effects on a broader scale. On the other hand, keeping a microscopic level of detail to compute feasible train paths seriously conflicts with the objective to consider complex instances that schedule frequent and interconnected services over long distances and considering long time horizons. Considering instances that are larger geographically or with a longer time horizon of traffic prediction results in an extensive view of the traffic situation, and in the enlarged solution space there exists a higher potential for improved operations. D'Ariano and Pranzo [33] consider the Short-Term Train Prediction problem to take into account larger time horizon of traffic prediction, by propagating dispatching decisions from a time period to the subsequent one. A conclusion is that when dividing large instances across time, a cascade approach is in fact sufficient to deliver solutions timely. Geographical division instead leads to a large degree of interconnection between the problems, i.e., two adjacent dispatching areas influence each other mutually

[9], unlike two time periods that influence each other only from the past to the future.

In order to have rescheduling algorithms able to deliver microscopically accurate solutions for this more complex problem within the short computation time needed in practice, an innovative distributed approach has been followed. In fact, advanced algorithms have been developed that optimise large scheduling instances but with a macroscopic level of detail for the timetabling problem [6] or to manage intercity trains, cancelling connections to reduce spreading of delays [60]; microscopic approaches which also feature advanced exact solution algorithms consider junctions or small dispatching areas [52]. On the other hand, if only simple local choices are taken into account, train movements in large networks can be simulated microscopically [49], and simple dispatching solutions computed quickly [66]. The following objective was therefore investigated, resulting in a methodology and a novel framework based on decomposition of large problems into smaller ones, as reported in Article III:

Research Objective 3: Define algorithms and procedures able to manage in real-time large and complex areas, while keeping microscopic detail.

The distributed approach resulted in the possibility to handle effectively large areas, and study the impact of dispatching at a global level. On the other hand, it required to address the eminent issue of coordination between multiple dispatching problems, and how advanced heuristics can further improve the performance of dispatching systems from a global point of view. Little research has been dedicated so far to this interesting problem, even though most of the dispatching control structures in real life exploit a similar distributed and hierarchically coordinated organization between dispatchers and network traffic controllers coordinating them. The existing literature is limited to algorithmic studies addressing artificial networks, mostly neglecting peculiarities of real-life instances, coordination of dispatching solutions, and constraints of railway operations at microscopic level. Therefore, a next research objective was set:

Research Objective 4: Devise intelligent procedures to achieve network-wide reduction of delay propagation.

Article IV reports on this investigation, which resulted in innovative approaches, procedures and algorithms able to deal with the computational complexity emerging from difficult instances that are dense (more trains per hour), geographically large (more trains), and have extra complexity due to coordination issues between adjacent dispatching areas. Advanced algorithms have been assessed according to a variety of factors, and proved to be able to exploit the increased potential of improvement due to the enlarged size of the instances. Moreover, other interesting properties of the dispatching problem have been studied. In fact, railway operations go beyond the simple minimization of delay propagation, and address also other constraints and requirements connected to operating rules, policies or objectives of the many stakeholders [59]. Studying them can be useful

to bridge the gap between operations and advanced mathematical models, by providing solutions that address better the complexity of multiple objectives considered by the stakeholders or prescribed by operational and practical rules, like energy efficient operations [3] or passenger satisfaction [56]. A general objective considered was therefore:

Research Objective 5: Study the impact of simple, yet potentially effective, policies in order to take into account additional interests and multiple objectives that are closer to real-life operational needs.

Such a generic and abstract research goal has been taken into consideration in two different manners, and applied to two relevant problems of railway operations. The additional objectives could be included as extra constraints in the optimisation model, or a multi-objective optimisation procedure could be designed on top of the existing model, considered as a black box. An application of the former approach to consider green wave operations thus scheduling trains while delivering energy efficient train profiles, and according to simpler dispatching actions, is reported in Article V. The second approach is followed in [18, 22] and discussed later in Section 1.5.

1.4 Scope and Outline

This thesis presents an innovative contribution to the investigation of the research objectives presented, concerning the application of advanced mathematical methods for real-time railway dispatching, with particular focus on complex, large and busy railway networks.

Instead of taking care of the unpredictable status of the network (i.e., traffic perturbations, delays, infrastructure disruptions) during the planning stage, a greater flexibility of operations can be achieved by managing proactively traffic in real-time. We assume that the main decisions about train orders, timings and routes could be deferred from a planning perspective to a control horizon, in order to deliver a better railway service [54].

We therefore focus on the railway infrastructure manager, which has the task to deal with the real-life unpredictability of operations, by identifying and resolving conflicts during operations; we mostly ignore connected issues related to the train operating companies that may influence train circulations, like rolling stock and crew management [50]. We show anyway how approaches can be setup to include additional interests in the optimisation model, to incorporate goals of train operating companies or passengers [22].

This work focuses on reducing impact of train conflicts and delay propagation, and not on how to reduce primary delays by e.g. improving maintenance of tracks and rolling stock, reducing incidents at stations or at level crossings. The problem of managing railway traf-

fic is affecting on many ways a lot of customers and stakeholders, including passengers, freight operators, train operating companies, the railway infrastructure manager. For this reason, a choice of objective function is more a problem of political choice rather than a technical one, as more or less every indicator proposed so far has advantages and disadvantages.

Single aggregated indicators (such as average delay, punctuality) suffer from having a very simple and unidimensional view on the problem; many different solutions with completely different outcome would result in similar values of this indicator. On the other hand, they are easy to understand and compare. In fact, many performance indicators should be considered to evaluate a particular solution proposed and take informed dispatching decisions. The detailed approach presented return solutions on which any indicator based on microscopic train movements can be computed. If detailed passenger data would be available, including Origin-Destination pairs of travelling passengers, and/or characterization of value of time and travel behaviour for passengers, it would be possible to measure the actual waiting time and travel time for the whole group of passengers and the entire public transport system.

This thesis follows instead a dispatching point of view, i.e. the limitation of consecutive delays is the main concern, as they are the immediate consequence of dispatching actions.

Moreover, it is based on the work of D'Ariano, that focuses mostly on reduction of maximum consecutive delay. In fact, for this particular objective function, it is possible to design powerful algorithms that perform an exhaustive search on the solution space. The availability of a tight lower bound is fundamental to deliver quickly good solutions even for large instances. In fact, approaches based on a similar modelling exist that are able to compute good lower bounds for different objective functions, but have a practical relevance only for problems with much less traffic, simpler infrastructure and shorter time horizon than those presented in this thesis [45].

Moreover, the two objective functions of maximum and average consecutive delays are not conflicting when reducing delay propagation, in the sense that reduction of the maximum consecutive is experimentally found to have good effects on the average consecutive delay. This is especially evident in iterative settings that have been proposed for handling the compound rescheduling an rerouting procedure [21], or speed management [34]. As shown in the following chapters, experimental analysis has in particular shown that the optimal solution minimizing the maximum consecutive delay leads to schedules that are better according to a variety of solution quality indicators when compared to other solutions.

On the other hand, the flexibility of the underlying model has been exploited to take into account many other possible objective functions and, in order to overcome a single point of view on the dispatching problem, and take into account multiple objectives and

different operating policies, additional properties of solutions are investigated, concerning Research Objective 5.

We restrict our analysis to safe railway operations, with regard to the existing safety rules and Automatic Train Protection systems, and do not focus on conflicts that result in Signals Passed At Danger (SPADs) or other incidents, assuming that drivers will always comply with the safety system (see e.g. [62]).

Other limitations include a laboratory setting, i.e., no on-line flow of information is considered. The stochastic behavior of real-life operations is limited to random entrance times, dwell times, infrastructure disruptions and speed limitations for trains. To this extent, a problem instance can be defined by a deterministic realization of expected entrance times of trains at the starting time of traffic prediction, that could be computed for instance by sampling probability distributions fitted to real data.

In fact, for an event scheduled far in time, the expected value of entrance time is the best estimate possible. A more sophisticated approach would exploit stochastic programming techniques to find solutions that take into account variability of entrance parameters as in [64]. Anyway, the complexity of the problem, the difficulty on modelling precisely all stochastic factors of railway circulation, and the hard time constraints on the computation time seriously limit the applicability of such an approach to real life instances.

Otherwise, the uncertainty in the expected entrance time due to a series of unmodelled dynamics can be studied with a closed loop approach. This would make it possible to evaluate the robustness of the overall setup against variations and errors in the definition of the instance, due to unmodelled dynamics and uncertainty of expected future, and the accuracy of the simulation of traffic movements, and the whole integrated rescheduling approach. In this sense, a possible structure of a closed loop implementation has been sketched in Chapter 3.

Anyway, such a work is beyond the scope of this thesis, as it needs an on-line coupling with an external validated simulator or with real operations. Moreover, closing a loop between automatic dispatching and feedback from operations needs to be complemented with good algorithms able to find solutions in the direct loop. This work was a contribution on improving the possibilities of the scheduling algorithms on the direct loop, to address large and complex instances.

A preliminary work on the direction of studying the influence of variability of relevant parameters, including expected entrance times is reported in [13]. From the COMBINE project [36], the results from closed loop experiments of advanced train management system confirm the trends found for the open-loop setting. Further work is therefore suggested on this direction to show the added value of a rescheduling system when put into operations, by setting-up a closed loop with an on-line feed of data from an external

simulator or the field.

The improvements reachable through advanced dispatching on the side of the railway infrastructure manager would complement the promising results achieved from the planning point of view by the railway operating companies through advanced mathematical models and algorithms for timetabling, crew and rolling stock scheduling problems. Together, advanced planning and optimal dispatching will lead to smaller delays experienced directly by passengers, fewer connections missed, and more reliable and attractive railway transport for both passengers and goods.

To fill the gap between current operations and advanced dispatching, the decision support system ROMA is presented, that has been used to model, implement and evaluate advanced rescheduling algorithms. To this end, this work combines detailed mathematical modelling of railway operations, state-of-the-art scheduling techniques [29], innovative rerouting algorithms [21] and detailed modelling of railway operational constraints to deliver dispatching solutions that are feasible from the point of view of microscopic train movements, quantitatively evaluated according to many performance indicators, and optimal at local and global level concerning delay propagation.

Particular interesting situations have been investigated more deeply in this thesis, where the task of the dispatcher is most challenged due to the difficulty of the problem. Determinant factors that have been dealt with are:

- the intrinsic complexity of operations, in connection with dynamic traffic management principles needing advanced real-time control;
- the enlarged solution space of the dispatching problem for dense networks and large stations with frequent and heavily interacting traffic;
- the exponential growth of instance size when addressing large networks, with heavy mixed traffic and long time horizons while keeping microscopic detail;
- a novel coordination problem between multiple dispatching solvers in distributed multi-area dispatching to deliver solutions that are feasible at local level, and optimal at global level;
- extra degrees of freedom, and additional policy constraints taken into account as secondary objectives while dispatching railway traffic.

All chapters of this thesis, that contain the main findings, are articles already published or submitted for publication; they are self-contained as they come with their own introduction and conclusion. The overall structure and the relations between them are shown in Figure 1.1. The research project resulted in additional contributions that have not been

included for the sake of conciseness and coherence; a discussion on the most important contributions of the works here included, and of those not included, is next presented in Section 1.5.

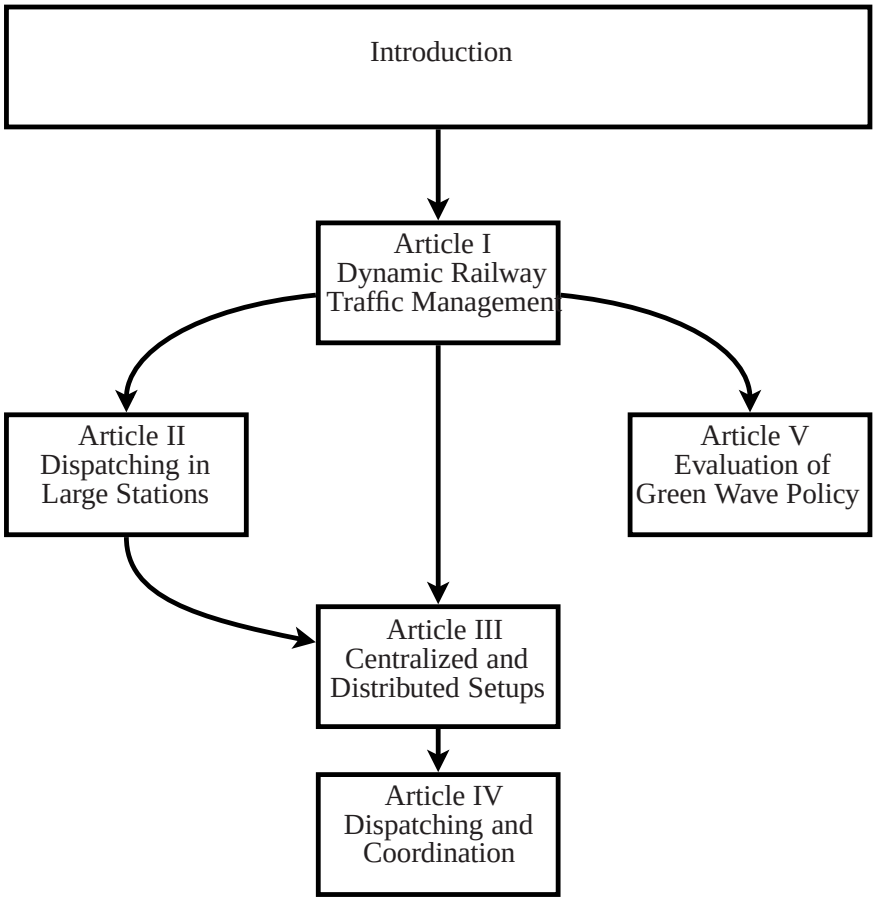


Figure 1.1: Structure of the thesis

Article I reports the results of the evaluation of advanced dispatching principles performed by using the dispatching support tool ROMA. A general introduction to the dispatching problem and solution techniques is reported, as well as a general overview of the core procedures of the ROMA system, and general applicability implications.

F. Corman, A. D'Ariano, D. Pacciarelli, M. Pranzo, Railway Dynamic Traffic Management in complex and densely used networks. In *Intelligent Infrastructures* (editors: Negenborn, Lukszo, Hellendoorn). Intelligent Systems, Control, and Automation: Science and Engineering, Volume 42, pages 377-404. Springer, 2010.

Article II reports on the application of advanced dispatching procedures on a complex

station with dense traffic and a complicated interlocking area. The extensive study and evaluation of dispatching procedures on algorithmically complex instances is complemented with applicability implications, i.e., a setup is proposed for a system interface to the dispatcher and a connection to operations.

F. Corman, A. D'Ariano, M. Pranzo, I.A. Hansen, Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area. In *Proceedings of the 3rd International Seminar on Railway Operation Modelling and Analysis* (editors: Hansen, Wendler, Weidmann, Luethi, Rodriguez, Ricci, Kroon). Zürich, Switzerland, 2009, and to appear in *Transportation Planning and Technology*, 2010.

Article III introduces a significant step towards handling large and difficult instances, by proposing a decomposed setup in which the problem of dispatching traffic over a large area is divided into two smaller dispatching problems to be coordinated together. An investigation of the characteristics of the distributed setup, a description of the distributed framework that has been designed and implemented, and quantitative assessment of centralized and distributed algorithms are there reported.

F. Corman, A. D'Ariano, D. Pacciarelli, M. Pranzo. Centralized versus distributed railway traffic control. In *Proceedings of the Eleventh international Conference on Advanced Systems for Public Transport (CASPT09)* (editors: Lo, Lam, Wong, Leung). Hong Kong, China, 2009, and accepted for publication in Public Transport: Planning and Operations, 2010.

Article IV continues the study on distributed dispatching, by comparing extensively centralized and distributed algorithms over a large railway area, studied from the point of view of different decompositions, multiple time horizons and various infrastructure availability scenarios, to quantify the benefits of decomposition in terms of solution quality and computation speed.

F. Corman, A. D'Ariano, D. Pacciarelli, M. Pranzo. Dispatching and coordination in multi-area railway traffic management. Submitted for publication.

Article V studies the application of a green wave policy to dispatch trains during perturbed operations, by introducing a suitable model, exploiting existing algorithms and assessing the benefits from the point of view of delay propagation, energy efficiency, and computational complexity.

F. Corman, A. D'Ariano, D. Pacciarelli, M. Pranzo, Evaluation of green wave policy in real-time railway traffic management, presented at *10th International Conference on Applications of Advanced Technologies in Transportation*. Athens, Greece, 2008, and published in *Transportation Research Part C*, Vol. 17(6), December 2009, pp 607-616.

1.5 Main contributions

The main achievements of the PhD project from a theoretical and methodological point of view, main findings and applicability implications are now outlined. The real-time train dispatching problem is modelled and solved in order to provide a decision support system for dispatchers managing and controlling traffic in real-time. The work of D'Ariano [29] has been used as a starting point, and further extended by additional studies, implementation and evaluation of algorithms and procedures addressing innovative principles. D'Ariano's work aimed at delivering a flexible model for railway traffic optimisation, and fast and effective scheduling algorithms working in a centralized manner; real-life instances comprising railway corridors were considered even though the test cases did not include any large station.

The work of D'Ariano suggested further research focusing on complex instances and large networks, study of severe disruptions, evaluation of dynamic railway traffic management strategies, and applicability (closed loop with operations, link with simulators or similar systems, feed from real data). In fact, when the algorithmic complexity of the problem is increased or difficult instances are considered, the performance of existing scheduling algorithms decreases substantially, resulting in suboptimal solutions or, even worse, failing in finding a feasible solution. Many factors affecting the ability to find optimal solutions for such difficult classes of instances have been studied.

A number of methodological improvements have been required to allow the solution of large and complex instances, considering a larger amount of degrees of freedom for dispatching operations, denser traffic and larger areas. This thesis keeps and extends the same structure and underlying model of D'Ariano, in particular concerning the alternative graph model [47], which proved to be a very valuable and flexible modelling tool, able to take into account additional operational constraints, practical requirements due to routes in large and complex interlocking areas of stations, and coordination constraints to guide the distributed multi-area dispatching towards a solution. Similarly, existing advanced optimisation algorithms such as a branch and bound scheduler [32] have been used as a basis and extended to take care of particular situations here investigated.

The point linking the two consecutive PhD projects can be found in the study of dynamic rerouting strategies, enlarging greatly the complexity of the dispatching problem. The research has been carried out jointly, first producing a paper that has been awarded at the RAS competition of INFORMS [14], and that has been published in the thesis of D'Ariano [29]. Meanwhile, research was going on, addressing and improving particular aspects concerning computational speed and solution performance; the final version of the algorithms and their evaluation can be found in [21], that is discussed in the following; note that the article is not included in this thesis in order to keep the thesis concise, and avoid to result in a partial overlap with the thesis of D'Ariano.

Dynamic rerouting is introduced to better exploit the available capacity at stations and interlocking areas. Issues from previous research have been addressed which resulted in intelligent metaheuristics that optimise routing of trains in combination with advanced rescheduling actions. More in detail, a generic tabu search scheme has been introduced to escape from local minima found by a simpler local search algorithm; two new routing neighborhoods have been considered and algorithmic enhancements evaluating in an approximated way the effectiveness of a move made the search procedure much quicker. When extensively tested under perturbed and heavily disrupted traffic conditions, the tabu search procedure results in great improvements in solution quality and remarkably reduced computation times, compared to the branch and bound algorithm of [32] and the local search algorithm of [31]. Moreover, for small instances for which the proven optimum is known, the new algorithms allow to close the optimality gap in a few seconds.

Research Objective 1: Evaluation of Dynamic Traffic Management Strategies

Article I [20] introduces extensively the topic of advanced dynamic traffic management, and reports on the assessment of different configurations of the ROMA system and various dynamic railway traffic management strategies, using the advanced mathematical programming techniques developed so far. The results show the effectiveness of advanced optimisation algorithms in handling flexible operations, with respect to simple and local dispatching procedures. ROMA can be applied to compute effective dispatching solutions for any given rail infrastructure and timetable, managing dense traffic in complex railway networks and under severe traffic disturbances, and also in the relevant case of a timetable that is not conflict-free. The impact of dynamic railway traffic management principles (e.g., flexible departure times at scheduled stops, dynamic train reordering and rerouting) is large, and depending on traffic density, infrastructure characteristics, and capacity available. Computational results quantify the interesting improvements contributed by the proposed principles; these benefits are the largest when the dynamic management strategies are used in combination with advanced dedicated algorithms.

Research Objective 2: Complex Areas

Dense train traffic results in very complex rescheduling instances, specially when considering busy and complicated interlocking areas of large stations, where many ordering decisions have to be taken between several trains with intersecting paths and different stopping platforms. To overcome this problem, a novel approach has been proposed by which the alternative graph formulation of the Conflict Detection and Resolution problem has been extended with the concept of aggregated block, in order to model routes in interlocking areas. As a result, it has been possible to take precisely into account station routes

considering the detailed topology of the interlocking areas that result in bottlenecks near to large stations, and to retain the microscopic detail needed to represent exactly incompatibilities of train movements at the level of track sections. Finally, this approach results in an improved lower bound for the scheduling algorithms by introducing special artificial resources for the routes, that translate to virtual machines in the job shop problem. As a result, simpler scheduling instances are actually solved, for which shorter computation times are achieved.

The extended approach has been evaluated on one of the most complex and busy station areas of the Dutch railway network. In fact, the large amount of data required for modelling the station area at microscopic level required automatic procedures to translate and generate timetable and infrastructure data from the Dutch InfraAtlas infrastructure format and DONS timetable database. A statistical description of perturbed operations is considered based on a vast set of realization data, and used to generate a large experimental set of traffic disturbances. The experiments assess the effectiveness of the advanced dispatching support tool, that delivers quickly dispatching solutions that are significantly better than the ones obtained by keeping the scheduled orders, by the ARI-like automatic route setting procedure or by the simple First-Come First-Served heuristic.

Those findings are reported in full in [25, 26], here reported as Article II. A closely related research regarded the detailed mathematical model of train operations on short track sections in interlocking areas, that was needed to properly manage large stations. The research resulted in a publication [27], discussed in the following, but not included in the thesis as rather specific, and sharing a lot of background with Article II.

To properly model trains running on short track sections in interlocking areas, a detailed mathematical model of train movements is fundamental, in order to compute exact blocking times, and detect precisely train conflicts. In particular, three different models of route reservation procedures (i.e., locking the route before the passage of the train, and releasing it after the train has cleared it) were considered: sectional lock, sectional release; route lock, route release; and route lock, sectional release. The latter corresponds to the most common situation for railway operations, but results in a particularly difficult job shop problem with sequence-dependent setup times. The sectional-lock, sectional-release model results in a better exploitation of available capacity, but requires more advanced systems to guarantee train separation. On the other hand, a route-lock route-release procedure provides an upper bound on the minimum train separation while being always feasible from the point of view of current operations. Computational experiments show that when advanced rescheduling algorithms are used, the gap between the three formulations is limited, thus providing a relevant evidence to use a simpler, yet always feasible route-lock, route-release mechanism in modelling railway traffic in station areas.

Research Objectives 3 and 4: Large Areas

In order to have a better understanding of network-wide effects in strongly interconnected railway networks, the dispatcher must look further ahead, in time and space. A large geographical size results in a large amount of trains interacting heavily in busy networks with dense traffic. Advanced train scheduling algorithms base their decisions on actual status of traffic flow and disturbances at the global level to achieve better performance with respect to local dispatching rules. On the other hand, mathematical models are seriously challenged when addressing large problem instances with many trains, block sections and stations. In fact, the dispatching problem is a complex NP-hard problem, and even the problem of deciding whether a feasible schedule exists or not is NP-hard [47]. For this reason, known exact solution algorithms might need an exponentially growing computation time to find optimal solutions, as the problem size increases in the amount of constraints and variables considered.

As a consequence, the quality of the solutions provided by centralized schedulers for large and complex instances is limited by the time available for computation of a new schedule; it is hard to take into account the microscopic detail needed to model feasible operations, a global view on the network, and solve the resulting problem in a short computation time. An innovative approach has been therefore proposed, that considers large dispatching problems divided in smaller problems to be solved independently.

Article III [16, 23] reports how the centralized decision support system ROMA is integrated into a new distributed framework, supporting coordinated solution of decomposed problems. To keep the computational burden low, and be able to deliver solutions in a timely manner, the framework has been designed to exploit high performance computing techniques, like concurrent and parallel execution. Moreover, it features a generic setup, i.e., different decompositions of large problems into subproblems can be evaluated, with a variety of underlying models and objective functions. The result is a comprehensive system that can be used to provide support to operational decisions at the level of dispatchers and the network traffic controllers coordinating them, by computing better solutions for large railway traffic management instances.

A domain-dependent decomposition has been used, aiming at reproducing the actual control structure of dispatchers and network traffic controllers. In fact, an empirical finding is that the main effects of control decisions are experienced in the vicinity of the resource over which the decision is taken. Dispatching problems can be thus successfully decomposed on a geographical basis into smaller interrelated problems to be solved independently. A bottom-up decomposition is used, that results in the local systems effectively dispatching trains by optimal scheduling, while a coordination level is in charge of harmonizing the solution. Regarding the coordination level, suitable aggregate information (times and orders computed by each local area) prove to be enough to synchronize and

coordinate at global level the solutions when combined with simple but effective coordination heuristics. The final solution minimizes consecutive delays at global scale for all trains running in the network considered.

Computational experiments on a real-world test case comprising a complicated and densely occupied area assess the performance of centralized and distributed approaches for railway traffic optimisation. Train schedules that are globally feasible are computed to handle perturbations characterized by multiple delayed trains and various levels of infrastructure availability. The distributed system yields better schedules in terms of delay propagation, in computation times that are always shorter; this twofold improvement is evident when advanced and reliable scheduling algorithms are used at local level.

The generic framework has been further used to study how scalable are local and global dispatching algorithms in their performances, to investigate the coordination problem, and to assess in depth the benefits of centralized and distributed control structures, in terms of feasibility of solutions, solution quality, and computation time [19]. The results are presented in [24], here reported as Article IV. The distributed approach has been extended to the general case of k areas, and a new asynchronous dispatching algorithm, called Job Greedy Heuristic, is introduced that schedules one train per time and combines good solution quality with a higher reliability of finding a solution than simpler algorithms.

An extensive campaign of experiments compares the centralized and distributed procedures on a large real-world railway network spanning hundreds of kilometers, considering fast dispatching rules, sophisticated scheduling heuristics, a state-of-the-art exact algorithm [32] and the new coordination procedures. The effects of different network divisions, different types of traffic disturbances and track blockage situations, and increasing time horizons of traffic prediction are compared. When dealing with short time horizons of traffic prediction, advanced optimisation algorithms are able to compute good quality solutions for delay minimization in a very short computation time. The solutions of the distributed setup are consistently better concerning delay propagation, compared to centralized approaches. Furthermore, when considering longer time horizons of traffic prediction, severe traffic disturbances and blocked tracks, the distributed approach presents a better feasibility performance than the centralized one.

More innovative approaches have been proposed recently to simulate the behavior of complex systems where multiple actors with conflicting goals interact. Train drivers, the dispatchers supervising them, and the network traffic controllers further in charge of coordinating dispatchers may be viewed as autonomous agents communicating, cooperating and negotiating in a common environment according to self-organized or hierarchical structures [39]. A preliminary investigation resulted in the conclusion that the current benefits of agent-based techniques are limited to modelling the interaction of multiple operators with conflicting goals and could be in principle suitable for proactive train traffic control. However, the ability to consistently result in feasible dispatching plans is hard to

be achieved by self-organized systems. Further studies in coordinating multiple entities should be done, so that quantitative advantages of decentralized or distributed railway traffic management systems can be evaluated, and motivate the introduction of such advanced approaches in the railway world [30].

Research Objective 5: Policies

Two complementary approaches have been used to include multiple interesting objectives and real-life policies in the research. The alternative graph model can be extended with additional constraints representing a particular operating policy; or a systematic multi-objective setup could be defined, that considers the dispatching procedures as black boxes addressing the conflict detection and resolution problem, and calling them iteratively while taking into account a second objective at a higher level.

A direction of research followed the former approach, and focused on evaluating a green wave speed management policy while dispatching trains in complex railway networks, and on defining the conditions under which it would be preferable in operations. To do so, two detailed models of the conflict detection and resolution problem are introduced, based on a flexible alternative graph formulation, the blocking time theory, and considering additional constraints on headway and running times of trains. A green wave policy restricts the only dispatching actions to dwell time extensions to hold and reschedule trains; no other explicit retiming or reordering decision is considered. Trains run therefore at their scheduled speed, facing only green signals between two planned stops and without unplanned braking or acceleration due to the signalling and safety system; this results in smoother and more energy efficient operations. Otherwise, the CDR problem can be solved assuming fixed train speed profiles, that are iteratively updated taking into account the actual aspects of the signalling system, as explained in [34]. This latter approach can exploit more flexibility in train speed management, but would require the implementation of a speed regulation system to carefully control train speeds.

Extensive computational experiments evaluate the two policies for two practical dispatching areas of the Dutch railway network with different characteristics of size, type of traffic, capacity level and traffic density. A green wave policy can reduce both train delays and energy consumption, without requiring sophisticated speed optimisation methods, when the network is not operated very close to physical capacity. However, advanced solution algorithms for solving the dispatching problem are needed, specially to optimally manage bottlenecks station areas where all the dispatching actions are to be done. In addition, a multitude of factors like the infrastructure characteristics, rolling stock used, traffic conditions, and tools available for traffic management should also be taken into account when considering the best approach for a specific case.

Article V [15, 17] reports the main findings about the study on green wave traffic management. A different approach to a problem considering multiple objectives and operating policies addresses management of passenger connections [18, 22]. From the point of view of the thesis, it is roughly equivalent to the one presented in Article V, as its main goal is to show and prove that other additional interests and complexity could be taken into account while dispatching train traffic; in order to keep the thesis concise and more coherent, the paper is not included in the thesis. On the other hand, its main contributions and achievements are discussed in what follows.

Cancellation of passenger connections is one of the main sources of passenger dissatisfaction, specially for long distance travels. In fact, transfer connections are relevant to passenger satisfaction but do not affect the feasibility of railway operations; cancellation of some scheduled connection reduces delay propagation at the expenses of the delay of passengers affected by the missed connection. The Bi-objective Conflict Detection and Resolution (BCDR) problem aims at finding a set of feasible train schedules that resolve the trade-off between the minimization of train delays and the maximization of respected transfer connections. Two heuristic algorithms are introduced to compute the Pareto front of non-dominated solutions, i.e., the set of solutions such that no other solution exists with a better value of both objective functions. CDR problems with fixed connections are iteratively solved by the dispatching system ROMA within a local search framework addressing the objective of maximization of the connections satisfied. A computational study based on a complex and densely occupied Dutch railway network during disturbed operations shows that both algorithms proposed explore the Pareto front quickly and effectively, finding the exact Pareto front for practical size instances. Results show that keeping even few critical connections between train services may have a strong impact on punctuality, therefore stressing the importance of good planning of connections. On the other hand, good dispatching actions balancing delay minimization and maximization of satisfied connections should be computed and implemented during on-line operations to deliver attractive railway services.

1.6 Practical relevance

The main practical focus of this research is on the role of the railway infrastructure manager, and more specifically on the dispatchers, that are in charge of managing railway traffic in real-time during operations. A summary of the main achievements of the thesis from a practical point of view is here reported.

So far, existing decision support systems for dispatching have been mainly investigated in academic environments; in actual operations only few systems go further than monitoring the actual situation, with a limited forecast of arrival times based on fixed running times

between stations, possibly reporting dispatching actions suggestions with no or little quality measure. Minimization of delays is insufficiently taken into account, neglecting both the impact on passengers and on consecutive delays. For instance, the German project DisKon (see e.g. [42]) resulted in one of the most advanced systems for railway traffic management that has been setup and evaluated in simulated operations. Real-time train position data are collected via UIC datagrams; based on this information, asynchronous algorithms use train priorities to optimise locally train traffic. Modifications to the current plan are computed by shifting or bending train paths along train lines; expected train arrival and waiting times can be computed, e.g. to update passenger connection plans. A comparison of Dutch and German dispatching systems based on different mathematical models, train dynamics, signalling systems is reported in [65].

This thesis presents an advanced decision support system, called ROMA, to significantly improve the task of the dispatcher. Its main focus is on managing difficult instances, i.e., dense traffic in large and saturated railway networks, busy station bottlenecks, to reduce delay propagation during serious disturbances on a network-wide scale. In those cases, the traffic situation becomes easily too complex and deviates much from the off-line determined timetable to be fully understood even by experienced dispatchers. Advanced decision support systems are required, as existing route setting procedures or simple dispatching rules come with no guarantee of optimality when the complexity of operations results in computationally difficult problems (i.e., bidirectional dense traffic, deadlock-prone situations, shortage of capacity in the short term, trains running closely behind each other) that have to be investigated in a very short time. The algorithms considered in ROMA are able to deliver a solution for computationally complex instances ensuring microscopic feasibility (and not having a macroscopically approximated solution with fixed running times) based on a forecast of the future (to take into account expected conflicts and potential deadlocks) that is precise and reliable (provided that correct data are available to simulate train dynamics) and takes into account global information and global view of traffic (and not myopic local information that may lead to suboptimal decisions or deadlocks), and finally, quantitatively better than what a very approximated and rough model (as a simple heuristic or the solution that a dispatcher may have on his mind) will deliver.

Compared to previous work on rescheduling, and specially the research done by D'Ariano [29], a greater level of complexity has been approached, in order to address the cases challenging the most the comprehension and the experience of the dispatcher. The improvements to the system result in the current possibility to consider retiming, reordering and rerouting actions to optimally solve instances with dense traffic concentrated in a bottleneck area (density), high traffic volume running on a large network (size), and additional constraints modelling particular operational policies. The results at all level of complexity quantify the benefits of using advanced algorithms against simple dispatching rules or the simulated current practice.

In particular, some interesting situations have been investigated further. Serious disruptions of train traffic ask for strong actions specifying a completely new plan of operations; for instance, so called emergency timetables are used in the Netherlands during strong disruptions. In those situations, dispatchers can hardly understand and quantify the precise global effects of local operational choices. Advanced decision support systems can result in a great help when handling seriously disrupted traffic. Multiple scenarios of high computational difficulty can be evaluated within strict time limits and at a very detailed level, making the dispatchers aware of the possible consequences of different control actions when they need it most [12].

A possible roadmap for application of the advanced dispatching system into operations is now discussed. In the current status of the system, we include stochasticity to describe train delays and perturbations; however, parameters and realization of delays are characterised as deterministic input data to the system, i.e., exactly known at the beginning of the time horizon of traffic prediction. The impact of small stochastic variations in input data, unmodelled dynamics or inaccurate parameter values toward solution quality of advanced dispatching solutions has been studied in [13]. From computational assessment, the optimised solution algorithms considered in ROMA offer a better delay reduction than simple dispatching rules, and also prove to be more robust to variations in input parameters than myopic rules based only on local information.

A further step would evaluate the advanced mathematical models proposed using real-time data of position and speed of trains. Such data can be either derived from currently available information like train describer data, combined with intelligent data mining tools [28], or measured directly by devices as GPS and communicated via channels such as GSM-R. By doing so, it will be possible to forecast precisely the circulation of trains in the short-term future. In actual operations, a closed control loop will address uncertainty in position, speed and parameters considered, with the dispatching support system updating successively its prediction in reaction to undergoing changes of the status of the network. In this way the full potential of advanced dispatching can be evaluated and compared with the decisions of human experienced dispatchers, at local and global level.

To develop such systems, interesting lessons can be learnt from a train driver assistance system that is now being introduced in the Netherlands [4]. In this approach, the status of the block sections along the expected train path is transmitted to the driver, basically giving him an extended sight. Even though no optimization approach is considered, just by having available such information, drivers can anticipate braking and coasting decisions, saving energy and reducing delay.

Acceptance by dispatchers and train drivers is a crucial factor. The dispatcher needs to be supported during operations by having a broader view of the current status of the network compared to what it is available now, with the expected conflicts identified at microscopic level, and suggesting potential solutions. For this reason, the interface must be designed

simple, unambiguous, and easily comprehensible, in order to hide the underlying complexity of the algorithms and avoid overwhelming the decision process when reporting required information on large areas, with dense traffic, and for longer time horizons [26]. A dispatching decision support system could either present a solution in detail, or deliver a set of possible solutions, each being more suitable according to one of the multiple objectives that are relevant in railway operations, other than reduction of delay propagation.

In fact, most real-life problems are characterised by many conflicting objectives, related to multiple actors, interests and operational requirements. Considering only a particular aspect of the problem might have a negative effect on the others, and benefits on one side could easily be hidden by (sometimes unavoidable) drawbacks on different sides; a clear combined view of the multiple objectives is required to avoid sub-optimal solutions. To take into account additional objectives and deliver more attractive dispatching solutions, two alternative approaches have been investigated, and proved to be applicable to consider extra objectives when optimising railway traffic. An approach addresses scheduling trains according a green wave traffic policy, in fact combining two important dispatching objectives: the minimization of train delays and of energy consumption. In this way, the benefits of that particular policy could be determined in relation to the characteristics of train traffic, timetable, and infrastructure [17]. A different systematic approach concerns the joint problem of reducing delay propagation (mainly a goal of the railway infrastructure manager) while avoiding to cancel passenger connections (a problem affecting railway operating companies). The proposed algorithms compute the Pareto front area of non-dominated solutions for this relevant multi-objective problem [22].

1.7 Recommendations for future work

The research done within this project is a follow-up of the PhD project of D'Ariano. Many future directions of research, earlier suggested by D'Ariano, have been investigated in this work; nonetheless a lot of issues are still to be tackled.

The principles of dynamic railway traffic management should be fully evaluated in a real-life pilot to quantify in operations the benefits that simulations have shown. Moreover, real-life application is required to get full feedback from all actors involved (i.e., passengers, dispatchers, railway infrastructure operators, and train drivers), and convince infrastructure managers that focus on proactive control via advanced techniques creates more opportunity for improvement than considering offline planning only. In fact, very frequent, reliable and dense services, well integrated in a multi-modal and heterogeneous transport framework can only be achieved with the introduction of advanced proactive dispatching systems, combined with limited infrastructure investments.

Additional policies and requirements from real-life operations could be considered and evaluated by advanced mathematical models, as many real problems have multiple stakeholders with multiple objectives, most of the times conflicting with each other. The ROMA system has already a unique position in being able to deliver solutions that optimise orders, routes, passing times and speeds of trains when considering complex and difficult instances, as this thesis has shown. Addressing more complex multi-objective setups requires careful modelling and advanced optimisation procedures, that might need to consider more degrees of freedom to come up with a final feasible solution; for this situation of increased computational load, advanced algorithms are needed. Other approaches could rely on multi-agent-based simulations, cooperative or competitive game theory. Such models may generate future insights on the interactions resulting from the balance of conflicting goals of the different stakeholders at a system-wide perspective.

A variety of fields share similarities with railway traffic management, and can benefit from advancements here discussed, like scheduling of public transport operations, airline operations management, scheduling and rescheduling of supply chains and production plants. Moreover, optimisation models and algorithms proposed can be applied to control other infrastructure systems, like power networks, waterway networks, telecommunications networks, infrastructures for the shipping of other commodities like gas or oil, and more in general road networks. Conversely, the capacity management strategies from those latter fields could be studied in railway traffic management.

Further study should concern the innovative approach here introduced based on decomposing and coordinating local solvers to address large instances. The effects of disturbances in large networks can be investigated more extensively, studying for instance various subdivisions of the network into a number of local areas, different aggregated information to be used at global level, frequency of data exchange between local solvers. Moreover, methodological, algorithmical and experimental issues of the distributed approach could be examined, including the loss in solution quality inherently introduced by taking into account the division in multiple solvers. In fact, when decomposing, the concept of centralized optimum is lost in favour of weaker concepts, like that of Nash equilibrium between the decomposed solvers. It is not yet known how to optimally coordinate multiple independent solution processes in order to reach better solutions, or to deliver quickly feasible solutions. From a practical point of view, advanced coordination algorithms could be considered, driven for example by ad-hoc heuristics, generic metaheuristics, exhaustive search procedures, robust optimisation algorithms, iterative approaches based on Lagrangian relaxation and control theory, distributed constraint satisfaction, artificial intelligence methods, or multi agent systems. The outcome could be further investigated theoretically, using an operations research or game theory perspective.

Bibliography

[1] *Eurobarometer*, 53, September 2000.

[2] A. Al-Ibrahim. *Dynamic Delay Management at Railways: A Semi-Markovian Decision Approach*. PhD Thesis, Tinbergen Institute, The Netherlands, 2010.

[3] T. Albrecht and S. Oettich. A new integrated approach to dynamic schedule synchronisation and energy-saving train control. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, and S. Sone, editors, *Computers in Railways VIII*, pages 847–856. WIT Press, Southampton, UK, 2002.

[4] T. Albrecht and J. Van Luipen. What role can a driver information system play in railway conflicts? In H. J. Van Zuylen and F. Middelham, editors, *Proceedings of the 11th IFAC Symposium on Control in Transportation Systems*. Delft, the Netherlands, 2006.

[5] N. Berends and N. Ouburg. Beschrijving ARI-functionaliteit. Technical Report 20, ProRail Internal Specification (in Dutch), Utrecht, The Netherlands, 2005.

[6] G. Caimi. *Algorithmic decision support for train scheduling in a large and highly utilised railway network*. PhD thesis, ETH Zürich, 2009.

[7] A. Caprara, L. G. Kroon, M. Monaci, M. Peeters, and P. Toth. Passenger Railway Optimization. In C. Barnhart and G. Laporte, editors, *Handbooks in Operations Research and Management Science*, volume 14, pages 129–187. 2006.

[8] M. Carey and I. Crawford. Scheduling trains on a network of busy complex stations. *Transportation Research, Part B*, 41 (2):159–178, 2007.

[9] Y. H. Chou, P. F. Weston, and C. Roberts. Dynamic Distributed Control for Real-time Rescheduling of Railway Networks. In I. A. Hansen, A. Radtke, J. Pachl, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

[10] Commission of the European Communities. *White Paper. European transport policy for 2010: time to decide*. 2001.

[11] Commission of the European Communities. *Mid-term review of the European Commission's 2001 Transport White Paper*. 2006.

[12] F. Corman, A. D'Ariano, and I.A. Hansen. Disruption handling in large railway networks. In B. Ning, C. A. Brebbia, and N. Tomii, editors, *Computers in Railways XII*. WIT Press, Southampton, UK, 2010.

[13] F. Corman, A. D'Ariano, G. Medeossi, and G. Longo. Robustness of advanced train dispatching solutions under disturbances. In *Proceedings of the 12th World Conference on Transport Research*. Lisbon, Portugal, July 2010.

[14] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. A tabu search algorithm for rerouting trains during rail operations. In *INFORMS Annual Meeting*. Seattle, Washington, U.S.A., 2007. First prize winner at RAS Student Paper Competition. Presentation available at the web site http://www.informs-ras.org/.

[15] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. 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, May 2008.

[16] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Centralized versus distributed railway traffic control. In *Proceedings of the 11th International Conference on Advanced Systems for Public Transport (CASPT09)*. Hong Kong, China, 2009.

[17] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Evaluation of green wave policy in real-time railway traffic management. *Transportation Research, Part C*, 17 (6):607–616, 2009.

[18] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Impact of rescheduling trains with transfer connections in disturbed traffic conditions. In *European Conference of Operational Research*, July 2009.

[19] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Optimization and coordination in real-time railway traffic management. In G. Fusco, editor, *Proceedings of the 1st International Conference on Models and Technologies for Intelligent Transportation Systems*. Aracne, Rome, 2009.

[20] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Railway dynamic traffic management in complex and densely used networks. In R.R. Negenborn, Z. Lukszo, and J. Hellendoorn, editors, *Intelligent Infrastructures*, volume 42 of *Intelligent Systems, Control and Automation: Science and Engineering*. Springer, 2010.

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

[22] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Bi-objective conflict detection and resolution in railway traffic management. *Trasportation Research, Part C*, 2010, (accepted for publication).

[23] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Centralized versus distributed systems to reschedule trains in two dispatching areas. *Public Transport: Planning and Operations*, 2010 (accepted for publication).

[24] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Dispatching and coordination in multi-area railway traffic management. 2010 (submitted for publication).

[25] F. Corman, A. D'Ariano, M. Pranzo, and I.A. Hansen. Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area. In Hansen, Wendler, Weidmann, Luethi, Rodriguez, Ricci, and Kroon, editors, *Proceedings of the 3rd International Seminar on Railway Operations Modelling and Analysis*. Zürich, Switzerland, 2009.

[26] F. Corman, A. D'Ariano, M. Pranzo, and I.A. Hansen. Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area. *Transportation Planning and Technology*, 2010 (accepted for publication).

[27] F. Corman, R. M. P. Goverde, and A. D'Ariano. Rescheduling dense train traffic over complex station interlocking areas. In R.K. Ahuja, R. Moehring, and C. Zaroliagis, editors, *Robust and Online Large-Scale Optimization*, number 5868 in Lecture Notes in Computer Science, pages 369–386. Springer - Verlag Berlin Heidelberg, 2009.

[28] W. Daamen, R. M. P. Goverde, and I. A. Hansen. Non-Discriminatory Automatic and Distinct Registration of Primary and Secondary Train Delays. *Networks and Spatial Economics and Spatial Economics*, 9 (1):47–61, 2009.

[29] A. D'Ariano. *Improving Real-Time Train Dispatching: Models, Algorithms and Applications*. PhD Thesis, TRAIL Thesis Series T2008/6, The Netherlands, 2008.

[30] A. D'Ariano and F. Corman. Existing and future approaches on railway traffic control from an agent-based perspective. In H. J. Van Zuylen, editor, *Proceedings of the 10th TRAIL Congress*. Delft University Press, 2008.

[31] A. D'Ariano, F. Corman, D. Pacciarelli, and M. Pranzo. Reordering and local rerouting strategies to manage train traffic in real-time. *Transportation Science*, 42 (4):405–419, 2008.

[32] A. D'Ariano, D. Pacciarelli, and M. Pranzo. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183(2):643–657, 2007.

[33] A. D'Ariano and M. Pranzo. An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. *Networks and Spatial Economics*, 9 (1):63–84, 2009.

[34] A. D'Ariano, M. Pranzo, and I. A. Hansen. Conflict resolution and train speed co-ordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 8(2):208–222, 2007.

[35] Dijksma and van Hijum. Motie van de leden Dijksma en van Hijum. *Tweede Kamer der Staten-General*, 2006.

[36] M. Giannettoni and S. Savio. 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, and S. Sone, editors, *Computers in Railways IX*, pages 695–704. WIT Press, Southampton, UK, 2004.

[37] R. M. P. Goverde. *Punctuality of Railway Operations and Timetable Stability Analysis*. PhD thesis, TRAIL Thesis Series T2005/10, The Netherlands, 2005.

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

[39] N. R. Jennings and M.J. Wooldridge. *Agent technology - foundations, applications and markets*. Springer, 1998.

[40] A. Kauppi, J. Wikström, B. Sandblad, and A. W. Andersson. Future train traffic control: control by re-planning. *Cognition, Technology & Work*, 8 (1):50–56, 2006.

[41] L. Kroon, D. Huisman, E. Abbink, P.J. Fioole, M. Fischetti, G. Maróti, A. Schrijver, A. Steenbeek, and R. Ybema. The New Dutch Timetable: The OR Revolution. *Interfaces*, 39 (1):6–17, 2009.

[42] A. Kuckelberg and E. Wendler. Real-time asynchronous conflict solving algorithms for computer aided train dispatching assistance systems. In J. Allan, E. Arias, C. A. Brebbia, C. Goodman, A. F. Rumsey, G. Sciutto, and N. Tomii, editors, *Computers in Railways XI*, pages 555–563. WIT Press, Southampton, UK, 2008.

[43] M. Lüthi. *Improving the Efficiency of Heaviliy Used Railway Networks through Integrated Real-Time Rescheduling*. PhD thesis, ETH Zürich, 2009.

[44] M. Lüthi, G. Medeossi, and A. Nash. Structure and simulation evaluation of an integrated real-time rescheduling and train control system for railway networks. *Networks and Spatial Economics*, 9 (1):103–121, 2009.

[45] C Mannino and A Mascis. Optimal real-time traffic control in metro stations. *Operations Research*, 57 (4):1026 – 1039, 2009.

[46] X. Mao, A. ter Mors, N. Roos, and C. Witteveen. Coordinating competitive agents in dynamic airport resource scheduling. In P. Petta, J. P. Mueller, M. Klusch, and M. Georgeff, editors, *Proceedings of the Fifth German Conference on Multiagent System Technologies (MATES2007)*, volume 4687 of *Lecture Notes in Artificial Intelligence*, pages 133–144. Springer, September 2007.

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

[48] M. Mazzarello and E. Ottaviani. A traffic management system for real-time traffic optimisation in railways. *Transportation Research, Part B*, 41 (2):246–274, 2007.

[49] A. Nash and D. Huerlimann. Railroad simulation using OpenTrack. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, and S. Sone, editors, *Computers in Railways IX*, pages 45–54. WIT Press, Southampton, UK, 2004.

[50] L. Nielsen, L. G. Kroon, and G. Maróti. A rolling horizon based framework for rolling stock rescheduling. In Hansen, Wendler, Weidmann, Luethi, Rodriguez, Ricci, and Kroon, editors, *Proceedings of the 3rd International Seminar on Railway Operations Modelling and Analysis*. Zürich, Switzerland, 2009.

[51] NS, ProRail, Railion, and V&W. Benutten en bouwen: Het plan van de spoorsector. Technical report, Nederlandse Spoorwegen, ProRail, Railion, Ministerie van Verkeer en Waterstaat, 2003.

[52] J. Rodriguez. A constraint programming model for real-time train scheduling at junctions. *Transportation Research, Part B*, 41 (2):231–245, 2007.

[53] J. Rodriguez. 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, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

[54] A. A. M. Schaafsma. *Dynamisch Railverkeersmanagement; besturingsconcept voor railverkeer op basis van het Lagenmodel Verkeer en Vervoer*. PhD thesis, Delft University of Technology, 2001. In Dutch, English summary.

[55] A. A. M. Schaafsma and M. M. G. P. Bartholomeus. Dynamic Traffic Management in the Schiphol bottleneck. In I. A. Hansen, A. Radtke, J. Pachl, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

[56] A. Schöbel. A model for the delay management problem based on mixed-integer programming. In *Proceedings of the 1st Workshop on Algorithmic Methods and Models for Optimization of Railways and Electronic Notes in Theoretical Computer Science*, volume 50 (1), pages 1–10. Crete, Greece, 2001.

[57] A. Schöbel and S. Scholl. Line planning with minimal traveling time. In *5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS)*, 2005.

[58] W. Schwanhäußer. *Die Bemessung der Pufferzeiten im Fahrplangefüge der Eisenbahn*. PhD thesis, Veröffentlichungen Verkehrswissenschaftlichen Institut RWTH Aachen, 1974. In German.

[59] J. Törnquist. Railway traffic disturbance management — An experimental analysis of disturbance complexity, management objectives and limitations in planning horizon. *Transportation Research, Part A*, 41:249–266, 2007.

[60] T. J. J. Van Den Boom and B. De Schutter. On a model predictive control algorithm for dynamic railway network management. In I. A. Hansen, A. Radtke, J. Pachl, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

[61] J. van den Top. Dynamic traffic management: planning with uncertainty to offer certainty. In I. A. Hansen, F. M. Dekking, R. M. P. Goverde, B. Heidergott, and L. E. Meester, editors, *Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis*. Delft, the Netherlands, 2005.

[62] J. van den Top. *Modeling Risk Control Measures in Railways*. PhD thesis, TRAIL Research School, 2010.

[63] P. Vansteenwegen and D. Van Oudheusden. Decreasing the passenger waiting time for an intercity rail network. *Transportation Research, Part B*, 41:478–492, 2007.

[64] M. J. C. M. Vromans. *Reliability of Railway Systems*. PhD thesis, Erasmus Universiteit Rotterdam, 2005.

[65] S. Wegele, F. Corman, and A. D'Ariano. Comparing the effectiveness of two real-time train rescheduling systems in case of perturbed traffic conditions. In J. Allan, E. Arias, C. A. Brebbia, C. Goodman, A. F. Rumsey, G. Sciutto, and N. Tomii, editors, *Computers in Railways XI*, pages 535–544. WIT Press, Southampton, UK, 2008.

[66] S. Wegele, R. Slovák, and E. Schnieder. Real-time decision support for optimal dispatching of train operation. In I. A. Hansen, A. Radtke, J. Pachl, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

[67] E. Wendler. The scheduled waiting time on railway lines. *Transportation Research, Part B*, 41:148–158, 2007.

[68] T. A. White. The development and use of dynamic traffic management simulations in North America. In I. A. Hansen, A. Radtke, J. Pachl, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

Chapter 2

Railway dynamic traffic management in complex and densely used networks

Abstract *This chapter is the first thorough assessment of a full implementation of the concept of dynamic traffic management in combination with advanced optimization tools. In the last years, several studies on partial implementations of this concept have been carried out reporting promising results. The development of new strategies for railway traffic control experienced an increasing interest due to the expected growth of traffic demand and to the limited possibilities of enhancing the infrastructure, which increase the needs for efficient use of resources and the pressure on traffic controllers. Improving the efficiency requires advanced decision support tools that accurately monitor the current train positions and dynamics, and other operating conditions, predict the potential conflicts and reschedule trains in real-time such that consecutive delays are minimized. We carry on our study using an innovative computerized railway traffic management system, called ROMA (Railway traffic Optimization by Means of Alternative graphs). An extensive computational study is carried out, based on two complex and busy dispatching areas of the Dutch rail network. We study practical size instances and different types of disturbances, including train delays and blocked tracks. Our results show the high potential of ROMA as a support tool to improve punctuality through intelligent use of the rail infrastructure and efficient use of the available transport capacity.*

This chapter is an edited version of the article

F. Corman, A. D'Ariano, D. Pacciarelli, M. Pranzo, Railway Dynamic Traffic Management in complex and densely used networks. In *Intelligent Infrastructures* (editors: Negenborn, Lukszo, Hellendoorn). Intelligent Systems, Control, and Automation: Science and Engineering, Volume 42, pages 377-404. Springer, 2010.

Reprinted with kind permission from Springer Science+Business Media.

2.1 Introduction

The aim of railway traffic control is to ensure safety, regularity, reliability of service and punctuality of train operations. Railway business strongly needs to improve the quality of service and to accommodate growth while reducing the costs. The punctuality analysis represents an important measure of rail operation performance and is often used as standard performance indicator. As reported in Goverde [9], in the autumn of 2001 the punctuality of the Dutch railway system decreased to below 80% (percentage of trains arriving at scheduled stops with a delay < 3 min). In 2003, a report by four major companies operating in the Netherlands [21] indicated a punctuality level of 95% as a target to reach within year 2015, despite the expectation of a significant increase of traffic intensity and the limited budget available to build new rail infrastructure. This chapter addresses such challenging target that can only be achieved through intelligent use of the existing rail infrastructure and efficient use of the available transport capacity. We describe traffic management strategies and dispatching support systems that can be used to improve punctuality of railway operations under disturbed traffic conditions.

Performance management is usually achieved by railway managers by carefully designing an off-line timetable and operating in real-time with strict adherence to it. However, train operations are intrinsically stochastic and traffic needs to be dynamically managed. When the scheduled railway traffic is disturbed, decisions have to be taken that modify the plan of operations in order to reduce delay propagation.

In the Netherlands, new pro-active approaches have been proposed to construct and manage the timetable. Two currently adopted strategies are the following:

- Development of robust timetables including specific running time supplements and buffer times to handle minor disturbances. This strategy requires to develop reliable estimation of delay propagation (see, e.g., [29]) in such a way that the amount of disturbances absorbed by the time reserves is increased.

- Development of dynamic traffic management strategies in which less details are fixed during the planning of activities and more control decisions are left to the dispatchers (see, e.g., [24]). This approach requires to develop effective traffic control procedures to fully exploit the enlarged degree of freedom.

To a certain extent these two strategies are complementary, since they both aim to improve punctuality of train operations at a microscopic level. In fact, minor train delays represent the vast majority of all delays and their influence on the level of train service can be minimized by careful management, provided that detailed and reliable information is available. This chapter focuses on the potential of the latter strategy and evaluates the

performance of given timetables for varying the degree of freedom and the traffic control procedures.

Dynamic traffic management strategies offer an interesting possibility to improve railway services by operating flexible timetables in which each train has to fit in a time window of arrival at a given set of feasible platforms/passing tracks. A specific platform and the exact arrival/departure times are then defined in real-time by the traffic controllers, which are therefore required to perform more actions with respect to non-flexible timetables. Under this strategy, traffic controllers have enhanced possibilities to react to unexpected events by adapting the timetable to the status of the rail network. Due to the strict time limits for computing a new timetable in case of disturbances, they usually perform manually a few modifications, such as adjustments of train routes, orders and speeds, while the efficiency of the chosen measures is often unknown [14, 31].

Dispatching systems support human dispatchers to manage traffic flow (as shown in Figure 2.1). Existing support systems compute rescheduling solutions on the basis of local information, i.e., they operate only “on the spot” and “now” and may implement simple dispatching rules. More advanced traffic management systems take into account the whole traffic in a larger area, detecting future conflicts affecting train movements, scheduling automatically trains in the whole area by using global information and suggesting possible changes of train orders or routes to the dispatcher, as well as displaying advisory speeds to train drivers. A comprehensive review on the related literature is reported in D’Ariano [4].



Figure 2.1: A train dispatcher at the traffic control center (Source: ProRail).

Most of the computerized decision support systems developed, so far, can provide fairly

good solutions for small instances and simple perturbations. However, they cannot deal with heavy disturbances in larger networks as the actual train delay propagation is only roughly estimated and does insufficiently take into account interactions among trains in the whole network. Therefore, extensive control actions are necessary to obtain globally feasible solutions.

In this chapter we compare a simple dispatching procedure with an advanced traffic management system. The simple procedure is a first come first served rule, a common practice in railway real-time management. The advanced system is the recently developed software, ROMA (Railway traffic Optimization by Means of Alternative graphs), that makes use of an optimization tool based on global information on the future evolution of the train traffic. This tool can be applied to various types of disturbances (such as multiple delayed trains, dwell time perturbations, block sections unavailability, and others) within a short computation time. The mathematical model and algorithms will be briefly described in Section 2.4. For more information, we refer the interested reader to, e.g., [2, 4, 5, 6, 7].

The innovative scientific contribution of ROMA is characterized by a combination of blocking time theory (see, e.g., [12, 22]) for the recognition of timetable conflicts in case of disturbances and a general discrete optimization model, based on the alternative graph formulation of [15], for the real-time evaluation of train reordering and rerouting in rail networks, while the costs of the different options are measured in terms of maximum and average delays at stations and other relevant points within the investigated network.

Computational experiments are based on two complex and densely occupied dispatching areas of the Dutch rail network, namely the Schiphol bottleneck area [13] and the Utrecht station area. The former is a dispatching area subject to high frequency passenger traffic, while the latter consists of a complex set of routes, heterogeneous traffic and less dense traffic conditions. A large set of disturbances are proposed for increasing values of train delays, multiple track blockage and different time horizons of traffic prediction. For each perturbed situation we generate several feasible schedules by using different configurations of the ROMA system. This allows us to quantify the effects of different traffic management strategies, in terms of train delays and computation time.

The next section presents the traffic management problem and related terminology. Section 2.3 describes the basic strategies for dynamic traffic management. Section 2.4 deals with the architecture of the ROMA system, while Section 2.5 reports on our computational experiments. The last section discusses the main achievements and gives some future research directions.

2.2 Problem description

In its basic form a rail 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 between two consecutive trains (see the example of Figure 2.2) depend on their speeds (running time of the involved trains), the braking rate of the second train (considered in the approaching time to the next signal), the train length of the first train (clearing time), the signal spacing (including sight and reaction time) and the swithing time. In case of technical or human failures, ATP ensures safe rail operations. Specifically, 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 is a track segment between two main signals, that governs the train movements, and may host at most one train at a time.

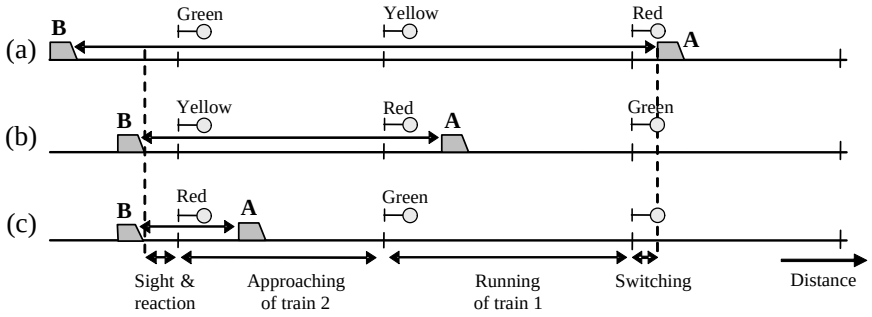


Figure 2.2: Track occupation of two trains (A and B) in case of different signal aspects.

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 Figure 2.2(c)), a yellow signal aspect means that the subsequent block section is empty but the following block section is still occupied by another train (see Figure 2.2(b)), and a green signal aspect indicates that the next two block sections are empty (see Figure 2.2(a)). 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 [8, 9, 12].

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). 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 or complex station interlocking areas, the routes for the individual trains need to be setup before entering and cleared after leaving, which takes a certain switching time.

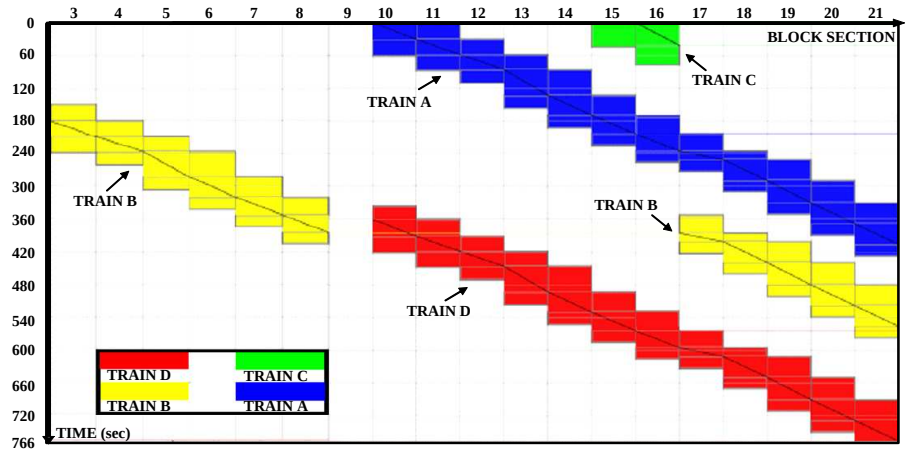


Figure 2.3: A timetable with four trains (A, B, C and D) running on a network of 21 block sections.

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 between the trains running in the network, 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 (see, e.g., [20]). Railway timetable design usually includes *recovery times* and *buffer times* between the train routes. The recovery time is an extra time added to the travel time

of a train between two stations, which corresponds to planning train speed smaller than maximum. Recovery times can be utilized in real-time to recover from delays by running trains at maximum speed. The buffer is an extra time inserted in the timetable between consecutive train paths besides the minimum headway, which prevents or reduces the propagation of train delays in the network.

We consider a timetable (see, e.g., the illustrative example of Figure 2.3) which describes the movements of all trains running in the network during a given time period of traffic prediction, 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 considered, such as minimum transfer times between connected train services. This is the time required to allow passengers to 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. Similarly, crew scheduling constraints impose a minimum time elapsed between the arrival of a train carrying (part of) the crew of another train and the departure of the latter train. 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, unexpected events occur during operations, which cause delays with respect to the operations scheduled in the timetable. The delay may propagate causing a domino effect of increasing disturbances. In Figure 2.4, the delay of one train (train C) propagates to the other trains running in the railway area (trains A and B). We define an *entrance perturbation* as a set of train delays at the entrance in a dispatching area, due to the propagation of delays from previous dispatching areas. A *disruption* is the modification of some infrastructure characteristics, such as the temporary unavailability of one or more block sections, which causes alterations in the train travel times and routes. 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.

A *conflict* occurs when two or more trains claim the same block section simultaneously, and a decision on the train ordering has to be taken (see, e.g., the disturbed traffic situation of Figure 2.4). A set of trains cause a *deadlock* when each train in the set claims

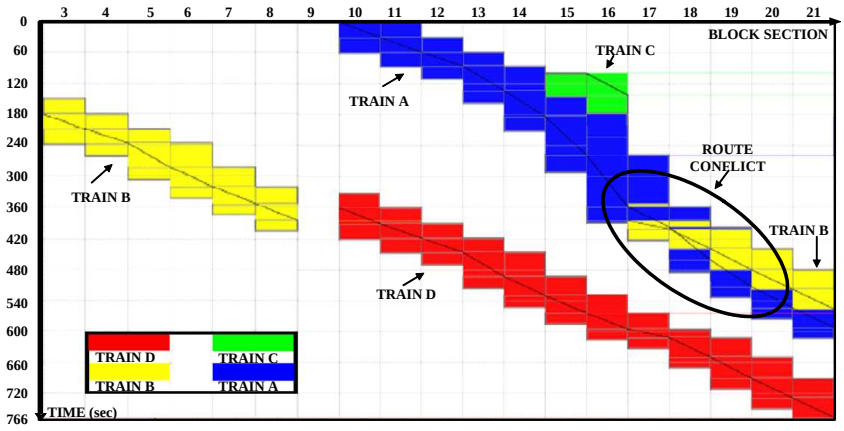


Figure 2.4: A route conflict between trains A and B caused by train C (see timetable in Figure 2.3).

a block section ahead which is not available, due either to a disruption or to the occupation/reservation for another train in the set.

Real-time railway traffic management copes with temporary 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 of traffic prediction. The task of dispatchers is to regulate traffic in a given dispatching area with the main objective of minimizing train delays in such a way that the new schedule is compliant with rail operating rules and with the entrance position of each train. The latter information is taken into account in the computation of the *release time* of each train that is the expected time, with respect to the starting time t_0 of traffic prediction, at which the current train enters 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 (see the train delays shown in Figure 2.5), 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.

The real-time railway traffic management problem is the following: 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, compatible with their initial positions and such that the selected train routes are not blocked, each train enters the network at its release time, no train departs from a relevant point before its minimum scheduled departure time, rolling stock



Figure 2.5: Display of train delays for a disturbed traffic situation (Source: Nederlandse Spoorwegen).

constraints and connected train services are respected, and trains arrive at the relevant points with the smallest possible knock-on delay.

2.3 Dynamic traffic management strategies

The standard practice in railway traffic management consists of the off-line construction of a timetable and the real-time control of trains with strict adherence to the timetable [1]. This rigid control practice makes timetable development a complex problem in which a compromise between capacity utilization and timetable robustness has to be provided [30].

Schaafsma [24] suggested the 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 (i.e., the maximum number of trains which may be operated through a line per time period). 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 further refined within the Dutch railway undertakings in the last years (see, e.g., [17, 18, 25, 26, 28]). The resulting RDTM principles consist basically of the following dynamic information for the management of congested areas:

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., low variability of travel times and connections. This would allow a greater possibility to 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/specification of train orders in real-time would allow to reduce delays if the dispatcher can find good schedules within strict time limits of computation.

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 use the infrastructure with more flexibility.

With an RDTM strategy, the plan of operations is only partially defined off-line in the timetable and then fixed in real-time, based on the actual status of the network and on the current train positions. The increasing degree of freedom left to real-time control leads to a larger workload for traffic controllers who must take several real-time actions. Thus, computerized traffic management systems are necessary to support dispatchers to exploit at best the opportunity offered by RDTM concepts.

Middelkoop and Hemelrijk [17] investigate the possible effects of introducing the RDTM principles in real-time traffic management by using the macroscopic simulation tool SIMONE. In a follow up paper, Middelkoop and Loeve [18] report on a computational analysis using the microscopic simulation tool FRISO.

Schaafsma and Bartholomeus [26] describe the first implementation of some of the RDTM principles at the Schiphol bottleneck of the Dutch rail network (see the static and dynamic information systems of Figure 2.6). The authors limit the assessment to routing flexibility and to train resequencing, using the first come first served rule at some specific railway junctions to resequence train movements.

D'Ariano et al. [7] compute optimal schedules with ROMA to resequence train movements, and assess the benefits of flexible departure times and flexible train sequencing based on the Schiphol area but using different, more challenging, timetables than in [26] and [27].



Figure 2.6: Routing flexibility by static (left side) and dynamic (righ side) information (Source: [26]). Note that "5 of 6 " is Dutch and translates to "5 or 6 " in English.

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 assess the full implementation of the three RDTM principles on different networks and timetables in order to evaluate their potential and limitations more in general. This need motivates the present chapter. In the next section we illustrate the decision support system used to carry on this assessment.

2.4 Decision support system

This section describes the implementation of the decision support system called ROMA (Railway traffic Optimization by Means of Alternative graphs) for railway traffic management. The ROMA 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 railway traffic management problem is divided into three subproblems: *(i)* data loading and exchange of information with the field, *(ii)* assignment of a passable route to each train in order to avoid blocked tracks and *(iii)* definition of optimal train routes, orders and specification of the exact arrival and departure times at stations as well as at a set of relevant points in the network, such as specific junctions and passing points. The ROMA system addresses the resolution of the three subproblems.

Figure 2.7 presents the ROMA system architecture, which is composed of interrelated procedures. A human dispatcher can interact with the decision support system by adding/removing constraints or changing the timetable. We now describe the function of each procedure and how the three introduced subproblems are solved:

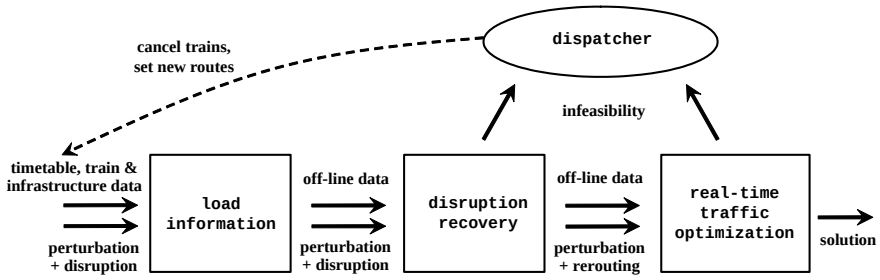


Figure 2.7: Architecture of the decision support system.

- 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 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.). We assume that the exact speed and location of each train are updated in real-time. Hence, the impact of inaccurate train data is supposed to be negligible.

- Disruption recovery (subproblem *(ii)*): Given a default route and a prioritized set of rerouting options, find a passable route for each train by avoiding the blocked tracks in the 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 with the aim of minimizing train delays in the network.

The real-time railway traffic management system must be able to efficiently detect and solve the conflicts arising in the rail network during perturbed operations. The traffic optimization procedure identifies the potential headway and route conflicts with a high level of accuracy (considering the physical characteristics of the rolling stock and infrastructure used by each train) 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 (each train travels with the minimum traversing time on each block section). To alter the original timetable as little as possible, we focus here on the development of conflict resolution actions in the dispatching area under study. Large timetable modifications and cancelation of train routes are among the possible dispatching measures but they are not performed automatically. The decision support system solution is suggested to the traffic controller before its actual implementation. After checking

the suggested solution, the traffic controller may either 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. 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 decision support system is computing a solution. However, since the decision 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 in a fourth subsection we point out the limitations and approximations of the proposed approach.

2.4.1 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 movements is the availability of a detailed and accurately updated data set. Precisely, running times and setup 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, 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 decision support system before execution. All this information is stored before the other procedures start.

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. 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 also includes a prioritized list of routing options, chosen by the dispatcher. 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. We assume that the drivers follow 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.

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

2.4.2 Disruption recovery

The *disruption recovery* procedure checks if there are unavailable block sections in the network (i.e., track blockage situation of Figure 2.8 (a)), which make some train route unpassable. This activity corresponds to the resolution of subproblem *(ii)*. For each train, this procedure discards disrupted 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 route*. The default route of each train and the set of remaining passable routes are then given to the real-time railway traffic optimization procedure of the ROMA system.

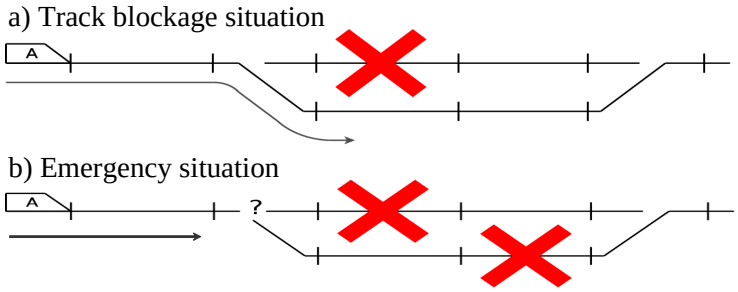


Figure 2.8: Disrupted railway corridors.

Since the ROMA system is only allowed to select the route of each train from a given set, when no passable route is available for a train, the system requires external support by the human dispatcher (e.g., emergency situation of Figure 2.8 (b)). In case of a heavy disruption, the human dispatcher authorizes ROMA to use an emergency timetable in which train routes are strongly modified, e.g., enabling a specific train to reverse its running direction according to a specific movement authority given by the traffic controllers.

2.4.3 Real-time traffic optimization

The *real-time railway traffic optimization* procedure is the decisional kernel of the decision support system. This procedure is responsible for detecting and solving train conflicts while minimizing the train delay propagation. 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 trains.

The conflict detection and resolution problem can be formulated as a job shop scheduling problem with no-store and no-swap constraints. Mascis and Pacciarelli [15] show 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 an alternative graph formulation of the conflict detection and resolution prob-

lem. 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. In general, other train operators' objectives could also be considered 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.

We now present a small railway network with two trains running at different speeds. Figure 2.9 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 A is a slow train running from block section 3 to block section 9 and stopping at platform 6. Train A can enter a block section only if the signal aspect is yellow or green. Train B is a fast train running from block section 1 to block section 9 through platform 7 without stopping. Train B can enter a block section at high speed only if the signal aspect is green. At t_0, we therefore assume that train B requires two empty block sections.

Figure 2.10 presents an alternative graph formulation for the traffic situation of Figure 2.9. 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 trains

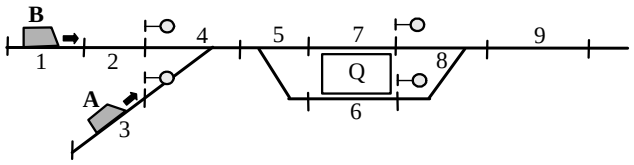


Figure 2.9: Example of a small railway network with two trains.

A and 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 the heads of trains A and 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 train A from the scheduled stop Q, while the length of arc $(AQ, A8)$ (not depicted in Figure 2.10) is its scheduled dwell time.

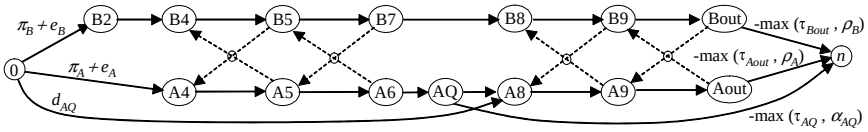


Figure 2.10: Alternative graph formulation of the example in Figure 2.9.

We next discuss the objective function for the formulation of Figure 2.10. Let τ_{AQ} be the earliest possible arrival time of train A at Q computed according to its initial position and speed, its assigned route and following a maximum speed profile in the empty network (i.e., by disregarding the presence of other trains). Let α_{AQ} be the scheduled arrival time of a train A at Q in the timetable, which can be infeasible in case of real-time disturbances. The length $-\max\{\tau_{AQ}, \alpha_{AQ}\}$ is the *modified due date* of train A at the scheduled stop. Let ρ_A and ρ_B be the scheduled exit times of trains A and B, the modified due dates at the exit of the network ("out") are $-\max\{\tau_{Aout}, \rho_A\}$ and $-\max\{\tau_{Bout}, \rho_B\}$. The resulting makespan $l^S(0, n)$ corresponds to the maximum consecutive delay computed for all trains at their relevant points.

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 alternative graph model (see, e.g., [4, 5, 6, 7]).

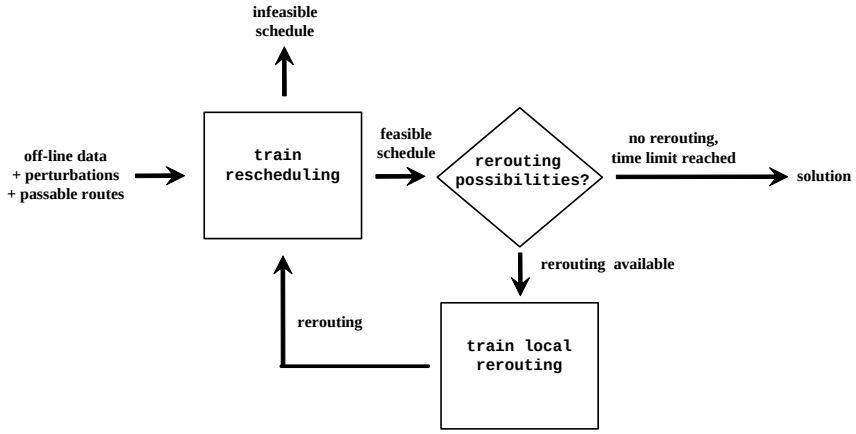


Figure 2.11: Architecture of the real-time traffic optimization procedure.

The real-time 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 2.11. 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 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 direct impact on the level of punctuality, this chapter compares two scheduling algorithms:

- Branch and Bound (BB): This is an exhaustive algorithm that explores all the reordering alternatives and chooses the one minimizing the maximum consecutive delay. Here we consider a truncated branch and bound [6] that returns near-optimal schedules for practical size problems within a short computation time.
- First Come First Served (FCFS): This is a well-known dispatching rule which gives

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 in the timetable.

We also implemented 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 novel Tabu Search (TS) algorithm, in order to escape from local minima [2]. Since the combinatorial structure of the conflict detection and resolution problem is similar to that of the job shop scheduling problem with routing flexibility, we focus on the tabu search approach that achieved very good results with the latter problem [16].

If the real-time railway traffic optimization procedure is unable to find deadlock-free and conflict-free schedules, the dispatcher has to carry out other types of timetable modifications such as introduction of new train routes, application of short-turning of trains in case of track blockage or even cancelation of train services at some stations (e.g., connections between passenger trains).

2.4.4 Discussion

The ROMA system is designed to help the dispatchers to cope with disturbances in the traffic flow by suggesting adjustments to the timetable of each train in terms of routing and timing, and by resequencing the trains at the entrance of each merging/crossing point. A strong point of ROMA is the simultaneous management of all trains running in an area, which allows maximization of punctuality and best exploitation of the available rail infrastructure (subproblem *(iii)*). ROMA is able to optimize railway traffic flow also in case of severe traffic disturbances, including the presence of blocked tracks and routes (subproblem *(ii)*), i.e., when emergency timetables are required and dispatchers need more support in their task.

The proposed decision support system is still a laboratory tool whose input/output interface is still limited to the loading of static (infrastructure and timetable) and dynamic (train positions and speeds) data gathered from the field (subproblem *(i)*). Besides, train speed profiles are computed separately from the scheduling problem in a preprocessing step. In order to incorporate ROMA within an advanced traffic management system, technical implementation issues concerning the practical operation, such as data transmission, communication of delays and the realization of the proposed dispatching measures, have still to be implemented, as well as the computation of new train dynamics when changing

the train orders at conflict points. These issues are outside the scope of this chapter (we refer the reader to, e.g., [11, 19, 23]).

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 measured times is computed and could be used, too, as input data for the proposed decision support system (see, e.g., [3, 10]).

2.5 Computational experiments

This section presents the experiments performed to evaluate the ROMA system over a large sample of real-life instances. The aim of the study is to assess to which extent train delays could be minimized by choosing suitable dispatching actions and dynamic traffic management strategies. We compare the solutions obtained by simple and advanced algorithms. ROMA runs on a PC equipped with a processor Intel Pentium D (3 Ghz), 1 GB RAM and Linux operating system. Each run of the BB algorithm is truncated after 10 seconds of computation (the best-known solution is often found during the first few seconds of computation), while the FCFS algorithm takes less than one second of computation. The whole time allowed to the real-time optimization procedure to compute a solution is limited to 60 seconds in order to be compatible with real-time operations.

2.5.1 Description of the test cases

This subsection describes the two dispatching areas under study. Both infrastructures offer interesting possibilities for 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 station areas, where trains may be allowed to stop at different nearby platforms. There are several potential conflict points in each dispatching area that are merging and crossing points along each traffic direction.

Schiphol dispatching area

The dispatching area around Schiphol tunnel is shown in Figure 2.12. The network consists of 86 block sections, 16 platforms and two traffic directions. The rail 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), the station of Nieuw Vennep, the shunting yard of Hoofddorp station, and two stations in Amsterdam, namely Amsterdam Lelylaan and Amsterdam Zuid WTC. The two traffic directions are largely independent except around Amsterdam Lelylaan station and at the border of Hoofddorp shunting yard. There are two intermediate stations: Hoofddorp and Schiphol.

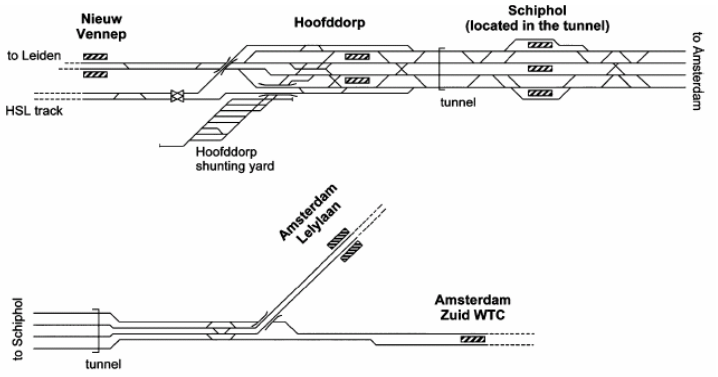


Figure 2.12: Schiphol dispatching area (Source: [13]).

We use an experimental timetable for passenger trains, designed to face the expected increase in traffic through this bottleneck area in the next years. This challenging timetable is very close to capacity saturation of this 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 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 was 20 trains per direction [13]. In 2009, the hourly timetable of Schiphol included 24 trains per direction [27]. We chose the more challenging timetable with 27 trains per direction in order to assess the effectiveness of ROMA under even more dense traffic conditions.

Table 2.1: Description of the Schiphol instances.

Num. of Delayed Trains	Max Entrance Delay	Num. of Blocked Tracks	Time Horizon Length
1 / 3 / 5	100 / 300 / 600	0 (100%) / 1 (82%) / 2 (67%)	15 / 30 / 60

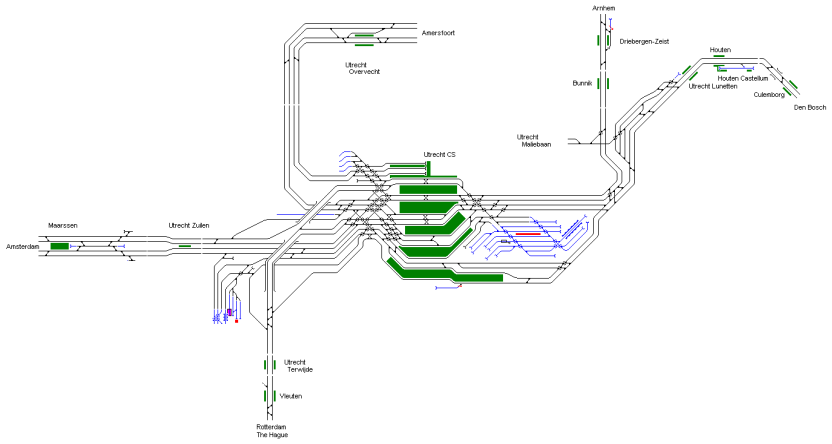


Figure 2.13: Utrecht dispatching area (Source: ProRail).

For some trains in the timetable, we consider alternative platform stops at Schiphol station. This flexibility is only applied to nearby platforms in order to limit passengers' discomfort. In total, there are 111 routes available for train rerouting.

Table 2.1 describes the disturbances for the Schiphol dispatching area. The first column indicates the number of trains delayed at their entrance in the network (we only delay the trains that enter the studied dispatching area in the first 15 minutes of traffic prediction), the second column the maximum values of entrance delay (in seconds), the third column the number of blocked tracks in the network, plus the corresponding percentage of passable train routes. The chosen blocked tracks are unavailable platforms at Schiphol station, causing some trains to be rerouted on another available platform in order to continue their trip. Finally, the fourth column of Table 2.1 reports the length of the time horizon of traffic prediction (in minutes). Three instances are generated for each value of columns 1, 2, 3 and 4, yielding a total of 81 entrance delay configurations.

Utrecht dispatching area

The railway network around Utrecht Central station is shown in Figure 2.13. Five main lines converge to Utrecht, connecting the North and South regions of The Netherlands to the lines to the West and the East. The network considered is delimited by the following stations: Utrecht Overvecht on the line to Amersfoort, Driebergen-Zeist on the line to Arnhem, Culemborg on the line towards Den Bosch, Vleuten on the line to Rotterdam and the Hague plus Maarssen on the line towards Amsterdam. In total, the diameter of the dispatching area is around 20 km long.

Table 2.2: Description of the Utrecht instances.

Num. of Delayed Trains	Max Entrance Delay	Num. of Blocked Tracks	Time Horizon Length
1 / 3 / 5	100 / 300 / 900	0 (100%) / 1 (93%) / 2 (91%)	15 / 30 / 60

Utrecht Central station is one of the most complex railway areas in The Netherlands, including more than 600 block sections and a very complicated and densely occupied interlocking area, defining a large amount of inbound and outbound routes. Most of the trains have a scheduled stop at one of the 20 platform tracks. The total amount of travelers at Utrecht Central Station is around 150.000 per day.

We use a provisional 2008 timetable that is cyclic with a cycle length of one hour. The trains are mostly for passenger services, operated by NS (Nederlandse Spoorwegen), except for a few freight trains. The timetable schedules up to 80 trains in a peak hour and provides connections between passenger services, coupling and splitting of rolling stock for intercity and local services coming from/going to Rotterdam, the Hague or Amersfoort, as well as re-use of rolling stock for commuter services towards Utrecht Overvecht and Culemborg. For each train in the timetable, we consider the possibility of rerouting trains to nearby platforms. This flexibility results in a total amount of 228 alternative train routes.

Table 2.2 describes the disturbances for the Utrecht dispatching area. Also in this case, three instances are generated for each value of columns 2, 3 and 4, yielding a total of 81 entrance delay configurations.

2.5.2 Dynamic traffic management strategies

Table 2.3 shows the effects of implementing various RDTM principles in combination with the ROMA system. Column 1 reports the reordering algorithms (“FCFS” is the First Come First Served rule and “BB” is the Branch and Bound algorithm of [6]) while Column 2 reports the rerouting strategy (“Default” means that ROMA selects for each train the default route, “TS” means that the route is chosen with the tabu search of [2]). Column 3 indicates whether departure flexibility is used or not by the ROMA system. Precisely, we consider one minute of flexible departure time for all trains at their scheduled stops.

Each row of Table 2.3 reports the average results on the 81 instances of the tables 2.1 and 2.2 for the two dispatching areas. Columns 4 and 5 show, respectively, the maximum and average consecutive delays for the Schiphol instances. The last two columns present the delays for the Utrecht instances. In both cases, delays are expressed in seconds and computed for all trains at their scheduled stops and at their exit from the network.

We next comment on the delay impact of applying different combinations of the RDTM

principles for the two dispatching areas under study.

Table 2.3: Average results on various configurations of the ROMA system.

Dynamic Traffic Management Strategies			Schiphol Dispatching Area		Utrecht Dispatching Area	
Reordering Algorithm	Rerouting Strategy	Departure Flexibility	Max Delay (sec)	Avg Delay (sec)	Max Delay (sec)	Avg Delay (sec)
FCFS	Default	No	473	68.9	252	7.0
FCFS	TS	No	420	58.6	222	5.1
FCFS	Default	Yes	466	65.2	181	3.6
FCFS	TS	Yes	386	54.9	159	2.4
BB	Default	No	313	63.4	221	6.6
BB	TS	No	255	46.2	213	4.8
BB	Default	Yes	307	63.4	154	3.2
BB	TS	Yes	249	45.6	151	2.4

As for the reordering algorithms, the BB algorithm provides by far better results with respect to the FCFS dispatching rule since the former algorithm chooses the train orders on the basis of global information on the delay propagation. Comparing the average results of all instances, the delay reduction is more evident for the Schiphol dispatching area (36% and 12% in terms of, respectively, maximum and average consecutive delays) than for the Utrecht dispatching area (9% and 6% in terms of, respectively, maximum and average consecutive delays). This is likely due to the fact that the Schiphol timetable is more dense of trains that run with short time headways.

As for the rerouting strategies, the TS algorithm exhibits a better capacity to keep delays small with respect to the default routes for both networks, even if rerouting is only allowed as alternative platforming of trains at their scheduled stops. Comparing the average results for all instances, the delay reduction is more evident at the Utrecht dispatching area (28% in terms of average consecutive delays) rather than at the Schiphol dispatching area (21% in terms of average consecutive delays) since Utrecht Central Station presents a larger number of alternative platforms.

As for the departure flexibility, its introduction in the Utrecht dispatching area allows to halve the average consecutive delays. On the other hand, the results in the Schiphol dispatching area are by far less effective (3% in terms of average consecutive delays). The different performance is probably due to the longer dwell times at Utrecht Central Station, leading to more possibilities to improve train punctuality by means of flexible departure times.

We have shown that each individual RDTM principle is an effective measure in managing train traffic, limiting the propagation of train delays. However, the best results are achieved when using all the three RDTM principles. For both networks, the worst configuration is the basic FCFS rule (see the first row of Table 2.3) while the best configuration is the one using the BB algorithm, the TS algorithm and flexible departures (see the last row

of Table 2.3). The gaps between these two configurations in terms of average consecutive delays are 34% for the Schiphol dispatching area and 66% for the Utrecht dispatching area.

2.5.3 Effects of increasing perturbations

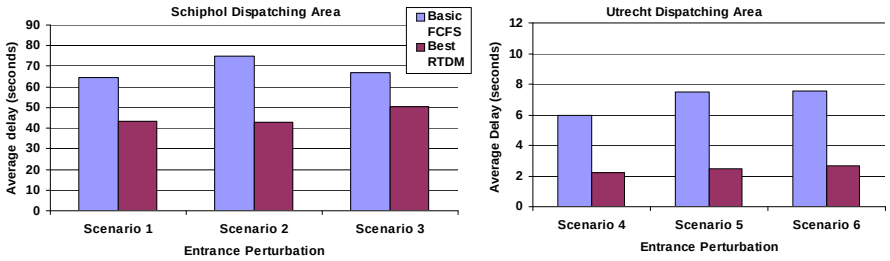


Figure 2.14: Comparison of the basic FCFS rule with the best RDTM configuration (BB, TS and flexible departures) for six types of entrance perturbation scenarios.

Figure 2.14 presents the average results obtained for six types of entrance perturbation scenarios. Three entrance perturbation scenarios (1, 2 and 3) are tested for the Schiphol dispatching area: 1 train delayed by 100 seconds (scenario 1), 3 trains delayed by 300 seconds each (scenario 2), 5 trains delayed by 600 seconds each (scenario 3). Similarly, three types of entrance perturbation scenarios (4, 5 and 6) are tested for the Utrecht dispatching area: 1 train delayed by 100 seconds (scenario 4), 3 trains delayed by 300 seconds each (scenario 5), 5 trains delayed by 900 seconds each (scenario 6). For each of type of scenario, the average results on 9 instances are reported, in terms of average consecutive delays, for the basic FCFS rule and the best RDTM configuration.

For all tested scenarios and for both the dispatching areas, the gap between the basic FCFS and the best RDTM is evident. While the results obtained from the best RDTM present an average consecutive delay slightly increasing with the number of late trains, the behavior of the basis FCFS is more erratic since this is a local decision rule and its output is less predictable.

2.5.4 Effects of increasing disruptions

Figure 2.15 presents the average results obtained for six types of track blockage scenarios. Three scenarios (1, 2 and 3) are tested for the Schiphol dispatching area: all tracks available (scenario 1), 18% tracks blocked (scenario 2), 33% tracks blocked (scenario

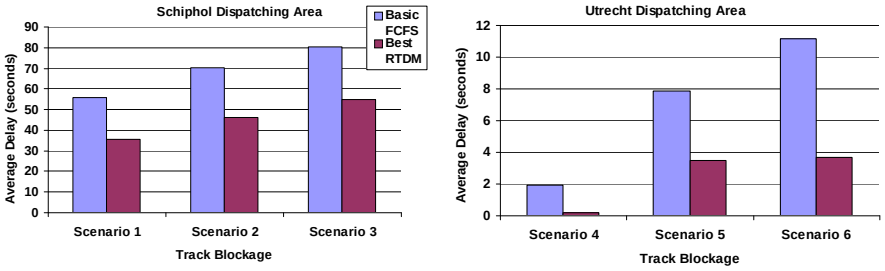


Figure 2.15: Comparison of the basic FCFS rule with the best RDTM configuration (BB, TS and flexible departures) for six types of track blockage scenarios.

3). Similarly, three scenarios (4, 5 and 6) are tested for the Utrecht dispatching area: all tracks available (scenario 4), 7% tracks blocked (scenario 5), 9% tracks blocked (scenario 6). For each of type of scenario, the average results on 27 instances are reported, in terms of average consecutive delays, for the basic FCFS rule and the best RDTM configuration.

In general, there is a considerable gap between the FCFS rule and the best RDTM configuration. For the Schiphol instances, the track blockage increases the average consecutive delays for both configurations of the ROMA system while their gap remains quite similar. For the Utrecht instances, scenario 5 has a stronger impact on delays compared to scenario 4, while scenario 6 has a limited impact compared to scenario 5 for both the system configurations, even if the best RDTM configuration presents a much smaller increase of delays.

2.5.5 Effects of increasing time horizons

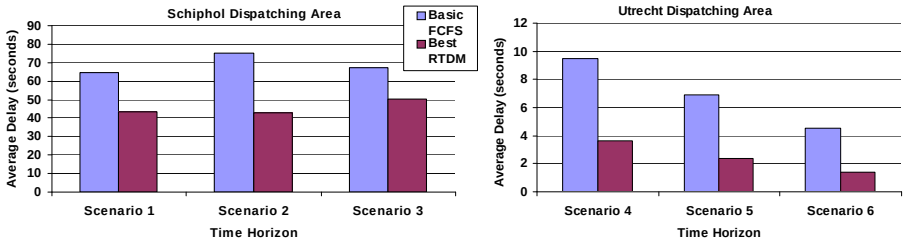


Figure 2.16: Comparison of the basic FCFS rule with the best RDTM configuration (BB, TS and flexible departures) for six types of time horizon scenarios.

Figure 2.16 presents the average results for six scenarios obtained by varying the time horizon and the test site. Three time horizon scenarios (1, 2 and 3) are tested for the

Schiphol dispatching area: 15 minutes with 19 running trains (scenario 1), 30 minutes with 31 running trains (scenario 2), 60 minutes with 54 running trains (scenario 3). Similarly, three scenarios (4, 5 and 6) are tested for the Utrecht dispatching area: 15 minutes with 22 running trains (scenario 4), 30 minutes with 39 running trains (scenario 5), 60 minutes with 80 running trains (scenario 6). For each scenario, the average results on 27 instances are reported, in terms of average consecutive delays, for the basic FCFS rule and the best RDTM configuration.

The results obtained for different time horizons also show a considerable gap between the two system configurations. For the Schiphol dispatching area, enlarging the time horizon of traffic prediction results in larger average consecutive delays. The very small time reserves in the Schiphol timetable cause a propagation of train delays in the network, even if the best RDTM configuration is able to strongly limit this effect. For the Utrecht dispatching area, there is a different trend of the average consecutive delays. Their values decrease appreciably, for both the system configurations, when the time horizon of traffic prediction is enlarged. This is mainly due to the larger amount of time reserves included in the Utrecht timetable with respect to the Schiphol one.

2.6 Conclusions

This chapter presents the performance of different configurations of the ROMA system and various RDTM strategies. The results show the effectiveness of using advanced optimization algorithms with respect to simple and local dispatching procedures. ROMA can be applied to compute efficient dispatching solutions for any given rail infrastructure and timetable, also when the timetable is not conflict-free. This fact enables its usage when managing dense traffic in complex railway networks and under severe traffic disturbances, such as when emergency timetables are required and traffic controllers need support to solve conflicts.

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. These benefits are the largest when the principles are used in combination with advanced traffic management algorithms.

Future research should address the integration of the proposed system into a larger framework, enabling to cope with several dispatching areas. To this end, it is important to address the decomposition of large problems into smaller problems to be solved by local dispatching systems, and their coordination may ensure globally viable and effective solutions for the whole rail network.

Acknowledgements

We thank the Dutch infrastructure manager ProRail (specially R. Hemelrijk, D. Middelkoop and L. Lodder) for providing the instances. This work is partially supported by the programs “Towards Reliable Mobility” of the Transport Research Centre Delft and by the Italian Ministry of Research, Grant number RBIP06BZW8, project FIRB “Advanced tracking system in intermodal freight transportation”.

Bibliography

[1] A. Caprara, L. G. Kroon, M. Monaci, M. Peeters, and P. Toth. Passenger Railway Optimization. In C. Barnhart and G. Laporte, editors, *Handbooks in Operations Research and Management Science*, volume 14, pages 129–187. 2006.

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

[3] W. Daamen, R. M. P. Goverde, and I. A. Hansen. Non-Discriminatory Automatic and Distinct Registration of Primary and Secondary Train Delays. *Networks and Spatial Economics and Spatial Economics*, 9 (1):47–61, 2009.

[4] A. D'Ariano. *Improving Real-Time Train Dispatching: Models, Algorithms and Applications*. PhD Thesis, TRAIL Thesis Series T2008/6, The Netherlands, 2008.

[5] A. D'Ariano, F. Corman, D. Pacciarelli, and M. Pranzo. Reordering and local rerouting strategies to manage train traffic in real-time. *Transportation Science*, 42 (4):405–419, 2008.

[6] A. D'Ariano, D. Pacciarelli, and M. Pranzo. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183(2):643–657, 2007.

[7] A. D'Ariano, D. Pacciarelli, and M. Pranzo. Assessment of flexible timetables in real-time traffic management of a railway bottleneck. *Transportation Research, Part C*, 16 (2):232–245, 2008.

[8] E. Goddard. Overview of signalling and train control systems. In *The 9th Institution of Engineering and Technology Professional Development Course on Electric Traction Systems*, pages 336–350. Manchester, UK, 2006.

[9] R. M. P. Goverde. *Punctuality of Railway Operations and Timetable Stability Analysis*. PhD thesis, TRAIL Thesis Series T2005/10, The Netherlands, 2005.

[10] R. M. P. Goverde and I. A. Hansen. TNV-Prepare: Analysis of Dutch Railway Operations Based on Train Detection Data. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, and S. Sone, editors, *Computers in Railways VII*, pages 779–788. WIT Press, Southampton, UK, 2000.

[11] S. Hailes. Modern telecommunications systems for train control. In *The 11th Institution of Engineering and Technology Professional Development Course on Railway Signalling and Control Systems*, pages 185–192. 2006.

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

[13] R. Hemelrijk, J. Kruijer, and D. K. De Vries. Schiphol tunnel 2007. Description of the situation. Technical Report 22, Holland Railconsult, Utrecht, 2003.

[14] A. Kauppi, J. Wikström, B. Sandblad, and A. W. Andersson. Future train traffic control: control by re-planning. *Cognition, Technology & Work*, 8 (1):50–56, 2006.

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

[16] M. Mastrolilli and L. M. Gambardella. Effective Neighborhood Functions for the Flexible Job Shop Problem. *Journal of Scheduling*, 3 (1):3–20, 2000.

[17] A. D. Middelkoop and R. Hemelrijk. Exploring the effects of Dynamic Traffic Management. In *Proceedings of the Dagstuhl Seminar on Algorithmic Methods for Railway Optimization*. Schloss Dagstuhl Wadern Germany, 2004.

[18] A. D. Middelkoop and L. Loeve. Simulation of traffic management with FRISO. In J. Allan, C. A. Brebbia, A. F. Rumsey, G. Sciutto, S. Sone, and C. J. Goodman, editors, *Computers in Railways X*, pages 501–509. WIT Press, Southampton, UK, 2006.

[19] G. Neil. On board train control and monitoring systems. In *The 9th Institution of Engineering and Technology Professional Development Course on Electric Traction Systems*, pages 211–241. Manchester, UK, 2006.

[20] L. Nie and I. A. Hansen. System analysis of train operations and track occupancy at railway stations. *European Journal of Transport and Infrastructure Research*, 5 (1):31–54, 2005.

[21] NS, ProRail, Railion, and V&W. Benutten en bouwen: Het plan van de spoorsector. Technical report, Nederlandse Spoorwegen, ProRail, Railion, Ministerie van Verkeer en Waterstaat, 2003.

[22] J. Pachl. *Railway Operation and Control*. VTD Rail Publishing, Mountlake Terrace, 2002.

[23] A. J. D. Santos, A. R. Soares, F. M. De Almeida Redondo, and N. B. Carvalho. Tracking trains via radio frequency systems. *IEEE Transactions on Intelligent Transportation Systems*, 6 (2):244–258, 2005.

[24] A. A. M. Schaafsma. *Dynamisch Railverkeersmanagement; besturingsconcept voor railverkeer op basis van het Lagenmodel Verkeer en Vervoer*. PhD thesis, Delft University of Technology, 2001. In Dutch, English summary.

[25] A. A. M. Schaafsma. Dynamic traffic management - innovative solution for the Schiphol bottleneck 2007. In I. A. Hansen, F. M. Dekking, R. M. P. Goverde, B. Heidergott, and L. E. Meester, editors, *Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis*. Delft, the Netherlands, 2005.

[26] A. A. M. Schaafsma and M. M. G. P. Bartholomeus. Dynamic Traffic Management in the Schiphol bottleneck. In I. A. Hansen, A. Radtke, J. Pachl, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

[27] V.A. Schaafsma, A.A.M.and Weeda. Operation-driven Scheduling Approach for Fast, Frequent and Reliable Railway Services. In Hansen, Wendler, Weidmann, Luethi, Rodriguez, Ricci, and Kroon, editors, *Proceedings of the 3rd International Seminar on Railway Operations Modelling and Analysis*. Zürich, Switzerland, 2009.

[28] J. van den Top. Dynamic traffic management: planning with uncertainty to offer certainty. In I. A. Hansen, F. M. Dekking, R. M. P. Goverde, B. Heidergott, and L. E. Meester, editors, *Proceedings of the 1st International Seminar on Railway Operations Modelling and Analysis*. Delft, the Netherlands, 2005.

[29] M. J. C. M. Vromans. *Reliability of Railway Systems*. PhD thesis, Erasmus Universiteit Rotterdam, 2005.

[30] E. Wendler. The scheduled waiting time on railway lines. *Transportation Research, Part B*, 41:148–158, 2007.

[31] T. A. White. The development and use of dynamic traffic management simulations in North America. In I. A. Hansen, A. Radtke, J. Pachl, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

Chapter 3

Reordering and rerouting in a complicated station area

Abstract *During operations railway traffic experiences disturbances that cause conflicts between train paths or even deadlock situations. Dispatchers need actions to restore feasibility and limit spreading of delays through the network. To help them in such task, the dispatching support tool ROMA (Railway traffic Optimization by Means of Alternative graphs) has been implemented in a laboratory environment. This paper reports on enhancements to the underlying train dispatching model as well as to the solving algorithms studied in order to tackle the increased complexity of busy stations with multiple conflicting paths and high service frequencies. Advanced train reordering and rerouting techniques are compared with straightforward rules and the current approach in the Netherlands. Extensive computational studies based on accepted statistical distributions of train delays assess the effectiveness of the ROMA tool in terms of solution quality and computation time.*

This chapter is an edited version of the article

F. Corman, A. D'Ariano, M. Pranzo, I.A. Hansen, Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area. In *Proceedings of the 3rd International Seminar on Railway Operation Modelling and Analysis* (editors: Hansen, Wendler, Weidmann, Luethi, Rodriguez, Ricci, Kroon). Zürich, Switzerland, 2009, and to appear in *Transportation Planning and Technology*, 2010.

3.1 Introduction

Railway traffic management is mainly directed towards the implementation of an existing plan of operations (off-line timetabling) and its adjustment due to disruptive events as quickly as possible (real-time dispatching). The timetable is characterized by departure and arrival times of each train at station platforms and/or at relevant merging and crossing points. The assignment of routes, platforms and passing times may require months, during which several variants are analysed in depth under economical and operational constraints. In real-time, unforeseen events may disrupt the timetable and thus the resolution of route conflicts and other infeasibilities is required.

The real-time dispatching process is to determine feasible (conflict-free and deadlock-free) train movements minimizing timetable deviations. An accurate prediction of the effects of delays and other disturbances requires modelling the evolution of train traffic in sufficient detail and considering the actual state of the network, both the dynamic behaviour of circulating trains and the dispatching measures used to control traffic. Hence, the precise delay propagation cannot be predicted by dispatchers, especially in case of complicated station areas, high density traffic and severe disturbances. Furthermore, railway managers are looking for decision support systems that enable their operators to determine implementable control actions as quick as possible. For these reasons, there is a need for developing more sophisticated and efficient decision support tools to forecast the network delay propagation for individual dispatching measures.

In this context, we implemented a laboratory dispatching support tool, called ROMA [6, 9, 10, 12]. This tool is designed to support dispatchers in the railway traffic monitoring, control and reaction to various types of disturbances (such as multiple delayed trains, dwell time perturbations, block sections' unavailability, and others) within a short computation time. The dynamic traffic control may coordinate the speed of successive trains on open track (retiming), avoid expected route conflicts (reordering) and allow for dynamic use of platform tracks or alternative paths along a line (local rerouting).

The innovative scientific contribution of our approach is characterized by a unique combination of the blocking time theory [14] for the recognition of route conflicts in case of disturbances and a general discrete optimization model, based on the alternative graph formulation [19], for the real-time evaluation of train reordering and rerouting in railway networks. The feasibility of the rescheduling options is verified in a very limited computation time by dynamically updating the corresponding headway distances, train speeds and blocking times, while the costs of the different options are measured in terms of maximum and average delays at stations and other relevant points within the investigated network.

In this paper, we extend the laboratory dispatching support tool ROMA to cope with more

saturated railway networks and complex infrastructure configurations. This further level of detailed information enables accurate computation of train trajectories and routes in complicated and busy station areas with several intersecting paths to lines and platforms. To this end, the following additional features have been implemented in ROMA:

- Automatic generation of timetable and infrastructure data (using an interchange format similar to RAILML, based on the Dutch InfraAtlas infrastructure format and DONS timetable database) in order to enable modularity in data handling and improve the information flow;
- Alternative graph formulation of the real-time train dispatching process in presence of many block sections and routes within interlocking areas;
- Modification of the algorithms to detect and solve possible conflicts on short block sections by taking into account incompatibilities between train routes.

We also discuss the interactions between dispatchers and decision support tool and the insertion of the tool in a dynamic setting with actual operations (a discussion on practical issues concerning the implementation, applicability and accuracy of the rescheduling actions can also be found e.g. in [18, 20]).

Our dispatching support tool is applied to the management of a complex and busy station area of the Dutch railway network, Utrecht Central Station, and to evaluate its performance. Disturbed traffic conditions are simulated, according to a statistical description based on realization data of train traffic. For each disturbance the tool provides a series of dispatching actions for a traffic control period of one hour. The experiments aim at evaluating the impact of the dynamic traffic control strategies by assuming that times, orders, routes and connections of all trains at all stations could be adjusted to produce a better feasible schedule. The computational results intend to demonstrate the computational effort of the proposed test cases and to assess the effectiveness of the advanced dispatching support tool. The dispatching solutions are presented in terms of train delays. Furthermore, the solutions obtained by the optimization are compared with those computed by using straightforward (local and on the spot) dispatching rules or no dispatching action (timetable solution).

The paper is organized as follows. Section 2 gives a brief literature review of models and algorithms for railway traffic management. Section 3 presents the real-time train dispatching process, a detailed model developed for the resolution of possible conflicts between trains with extensions to the management of complex station areas and the train dispatching support tool based on that model. A discussion on the real-time application of this tool is also proposed, along with a description of the possible interactions with the human operator (dispatcher). Section 4 reports the computational results obtained for the

Utrecht station area. Section 5 concludes the paper with a discussion on the current state of development of the dispatching support system and further research directions.

3.2 Related literature

Existing models for solving routing and scheduling problems can be classified according to two levels of approximation: macroscopic models and microscopic models. Off-line timetabling usually relies on macroscopic models, while microscopic approaches are mandatory when dispatching train traffic in real-time.

To keep complexity low at the planning stage, macroscopic approaches model a railway network as a simplified series of links connecting stations. A fixed running time is required to travel between two stations, and a fixed headway time is imposed between consecutive trains on the same link or platform at stations. The time variables are normally bounded to full minutes. Several works on timetabling use a formulation based on the periodic event scheduling problem by Serafini and Ukovich [24], which assumes infinite capacity at stations and a rough model of headway times and safety system.

For the Dutch railways, DONS is the macroscopic tool adopted to design timetables. A network scheduler module, called CADANS (see e.g. [23]), determines a feasible cyclic network timetable, while having fixed running and minimum headway times, and neglecting capacity at stations. A second level module, called STATIONS (see e.g. [29]), manages the routing of trains in complex station areas. The model takes into account incompatibility between routes according to predefined safety constraints and builds a graph of incompatibilities in which a maximum weight node packing corresponds to a feasible routing solution, while neglecting the impact of signalling and train length on blocking times.

Carey and Crawford [5] present a sophisticated model and a novel heuristic procedure to assess the benefits of an existing draft timetable for a network of busy stations. The precise track layout of stations and incompatibility of conflicting routes and platform tracks occupations are taken into account in the model, while train separation is formulated as a fixed minimum headway distance. The heuristic solves the route conflicts along the tracks by selecting the least immediate delay cost.

Caimi et al. [4] solve ordering and routing problems in station areas simultaneously, by building a large conflict graph that takes into account multiple scheduling possibilities for each train. A fixed point iteration algorithm has been implemented to compute a feasible solution in a reasonable computation time. The procedure assumes that trains have fixed running and headway times in interlocking areas, while acceleration, braking, dwell time extensions, as well as variations of train length, are not discussed.

A greater level of detail is needed to properly control railway traffic during operations. Accurate train positions, speeds, and acceleration and braking time losses have to be computed for a reliable prediction of the trains trajectories including blocking times. The speed profile of trains has to be computed according to the actual speed limits and the corresponding acceleration and deceleration rates (see e.g. [27]). The signalling system with actual signal aspects needs to be modelled along corridors and in station areas, while a precise layout of interlocking areas is required to take into account incompatibilities between routes.

The resolution of the real-time dispatching process is a demanding task, specially in a network of stations with multiple merging and crossing lines, and experienced dispatchers usually limit the degree of freedom by looking for simple solutions, that often may be sub-optimal. Among the recent contributions, Jacobs [17] has developed a train rescheduling model based on detection of route conflicts with high precision, by means of blocking time theory, with the objective of reducing the running time extensions. A heuristic procedure based on train priorities is applied to build up incrementally a dispatching plan, solving infeasibilities in an asynchronous and locally optimal way.

Rodriguez [22] studies rerouting and reordering possibilities for a small network with up to 12 trains. A job shop scheduling model with additional state resources constraints is proposed to detect and solve route conflicts. Synchronization constraints are used to keep train running with sufficient headway distances, even in case of yellow or red signal aspects. However, variability of train speed profiles is not considered. Dispatching solutions are computed by constraint-based programming in a short computation time.

Törnquist and Persson [25] propose a mixed integer linear programming formulation to manage disturbed traffic conditions by means of train reordering and rerouting. They assume a fixed headway time between trains and fixed running times along segments between stations and relevant interlocking areas. The objective is to minimize a cost function based on train delays. Results on a real network with few delayed trains show that there are still good margins for improving punctuality of trains.

D'Ariano [9] uses an alternative graph formulation to perform train reordering and rerouting. Blocking time theory is adopted to check whether the required minimum headway distances between trains are respected. The possibilities of optimizing the routing of trains are explored by means of metaheuristics while the problem of scheduling trains in complex areas is solved by a branch and bound algorithm within a given time of computation. Numerous results on two practical dispatching areas with small stations and several railway corridors are reported to assess the effectiveness of the proposed approach.

Most of the previous work on automated rescheduling has investigated simplified railway networks and implicit representation of incompatibilities in interlocking areas. The complexity of managing station areas with dense traffic and multiple conflicting routes

is acknowledged by dispatchers, specially in case of large disturbances. Hence, the development of a decision support system for rescheduling in complex interlocking areas is of great relevance to improve quality of railway operations. In order to proper manage traffic in complicated interlocking areas and provide accurate solutions to the dispatchers, detailed microscopic information is to be taken into account.

Therefore, there is a need to develop methods that use a rather detailed description of the rail network that is able to model the possible reordering and rerouting possibilities along tracks and inside main station areas. The solution of such instances of increased complexity should be found without losing necessary details. Realistic train separation rules based on signal spacing, interlocking of routes and blocking time theory should be used, as the use of fixed headway times is too rough in case of dense traffic and multiple interactions between several inbound and outbound routes at stations. Moreover, the trade-off between solution quality of the rescheduling process and time required to compute dispatching solutions is of crucial importance in the practical setting of a decision support tool and must not be underestimated.

3.3 Real-time train dispatching

3.3.1 Problem description

A railway network is composed of stations connected by lines and tracks, that are operationally divided in block sections. The passage of a train through a block section is called *operation* and a sequence of operations to be traversed by a train is called *route*. At any time a route is passable if all its block sections are available, i.e., there is no blocked track along the route. The timing of a route specifies the starting time t_i of each operation in the route. Each operation requires a travelling time, called *running time*, that is computed according to the dynamics of each rolling stock, namely the parameters about maximum speed and acceleration ratios, speed restrictions on the infrastructure, and the driver behaviour according to signalling system constraints (e.g. in case of a red signal the train must brake to a complete stop and then re-accelerate).

According to standard railway safety regulations [2], no more than one train at a time is allowed to occupy any block section (conflict-free condition); the three-aspect fixed block signalling system and the Automatic Train Protection (ATP) system are used to ensure a safe headway between successive trains and to generate an automatic brake in case of accidents or technical failures errors. This headway translates to a *setup time*, required to modify the signalling aspect and to change the route after the tail of a train has released the track segment. The minimum headway time between two successive trains depends on their speeds, the braking rate of the following train, the length of the preceding train,

the distance between signals and the time needed to setup a new route. A train speed profile is acceptable when the acceleration rate, deceleration rate, maximum speed of the actual rolling stock and the minimum headway time between trains are respected.

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 should 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. During the service of a line rolling stock completes a number of round-trips and may change their composition. In the Netherlands, train services may be coupled and uncoupled during operations, by combining one train with another train or by splitting into two distinct train units. Railway timetables include a technical service time at terminal stations, which is the time between the arrival of a train and the start of a new service using the same rolling stock.

Various types of perturbations to the timetable occur during operations. Examples of disturbances may include infrastructure breakdowns, delay of trains entering the dispatching area from the previous one, speed limitations due to technical failure or extended dwell times at stops. Such disturbances may lead to potential conflicts or even deadlocks. A potential *conflict* is when two or more trains claim the same available block section simultaneously, and a decision on the train ordering has to be taken. In that case the movement authority is given to only one of the trains involved at a time. In case of short headway distances, the other trains are forced by the signalling and train protection system to decrease their speed or to even completely stop on the open track or within the interlocking area. The conflict resolution may therefore cause some train delays. A set of trains cause a *deadlock* when each train in the set claims a block section ahead which is not available and cannot be made available, due either to an infrastructure breakdown or to the occupation/reservation for another train in the set.

Following the notation of [9, 13], the *total delay* is the positive 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. An *initial* (or *primary*) *delay* is directly caused by late departures, failures or disturbances and exceeds available running time margins until the next potential conflict or timetable reference point. A *consecutive* (or *knock-on*) *delay* is caused by the interdependence between trains running in the network and in principle can

be minimized by proactively managing railway traffic.

Figure 3.1 shows two traffic situations that illustrate the problem addressed in this paper. In both situations, the railway network is composed of 10 block sections and a station with 2 platform stops on block sections 3 and 4 respectively.

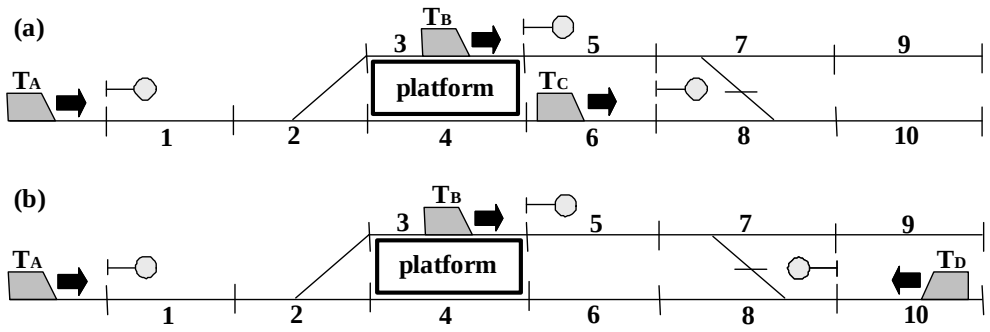


Figure 3.1: Examples: (a) situation with no deadlock; (b) situation with potential deadlocks

In Figure 3.1(a), three trains (T_A, T_B and T_C) are running in this simple network. T_A is a fast train with no platform stop and three alternative routes: R_A^1 = [$A1$, $A2$, $A3$, $A5$, $A7$, $A9$], R_A^2 = [$A1$, $A2$, $A3$, $A5$, $A7$, $A8$, $A10$] and R_A^3 = [$A1$, $A2$, $A4$, $A6$, $A8$, $A10$]. T_B is a fast train with a platform stop on block section 3 and two alternative routes: R_B^1 = [$B3$, $B5$, $B7$, $B9$] and R_B^2 = [$B3$, $B5$, $B7$, $B8$, $B10$]. Train T_C is a slow train with only one route: R_C = [$C6$, $C8$, $C10$]. In this traffic situation there is no deadlock in the network for all possible route combinations. However, the train routes share a number of block sections (e.g. R_A^3 and R_C share block sections 6, 8 and 10).

During disturbed operations, trains face initial delays and the conflict-free condition requires train ordering and routing decisions in order to compute a feasible schedule and to limit the propagation of consecutive delays. For the situation of Figure 3.1(a), let consider that the dwell time of T_B is strongly delayed and the running time of T_C is considerably larger than the scheduled one. A critical routing decision is to select a route to T_A. In fact, R_A^1 and R_A^2 face at least a potential conflict with R_B^1 and R_B^2 on the stopping point (i.e., block section 3) while R_A^3 face at least a potential conflict with R_C on non-stopping points (i.e., block sections 6, 8 and 10). A critical ordering decision is to select a precedence between R_B^2 and R_C on block section 8.

In Figure 3.1(b), three trains (T_A, T_B and T_D) are running in the network. T_A and T_B have the same routes as for the previous situation. T_D is a fast train with no platform stop and two alternative routes: R_D^1 = [$D10$, $D8$, $D6$, $D4$, $D2$, $D1$] and R_D^2 = [$D10$, $D8$, $D7$, $D5$, $D3$, $D2$, $D1$]. In this traffic situation there are various sets of routes that lead to

deadlocks. For instance, R_A^3 is always infeasible while R_A^2 and R_B^2 are only feasible if T_A and T_B are scheduled after T_D on block sections 8 and 10.

3.3.2 Problem formulation

The real-time dispatching process can be approached by retiming trains, i.e., by using running time supplements that are included in the timetable. The train sequence at junctions and merging points may also be adjusted (reordering) to the actual delay situation, e.g. the trains can be rescheduled in the order they arrive. A further degree of freedom is to change locally the route used by a train (rerouting), e.g. an empty platform can be used instead of causing a delay while waiting for a still occupied track.

This section introduces a detailed formulation of the real-time process of changing dwell times as well as train orders and routes in order to solve potential conflicts and to avoid deadlocks in case of disturbed operations. This is called Conflict Detection and Resolution (CDR) and can be partitioned into two subproblems: *(i)* a scheduling problem, for which the starting time of each operation is determined, and *(ii)* a rerouting problem, for which a route is associated to each train among a set of rerouting possibilities. In what follows, we refer to Conflict Detection and Resolution with Fixed Routes (CDRFR) to denote the scheduling problem with a single route and a fixed speed profile assigned to each train, with the objective of minimizing the consecutive delays.

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. In job shop scheduling [21], 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.

In terms of the CDR problem, jobs correspond to running trains and machines to block sections of the signalling and train control system. The processing time of an operation can be used to represent the running time of the train on the corresponding block section. Since a block section cannot host two trains simultaneously, there is a potential 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 potential 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 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.

Mascis and Pacciarelli [19] introduce alternative graphs to model variants of job shop scheduling problems. An alternative graph is a triple $\mathcal{G} = (N, F, A)$ being N a set of nodes, F a set of fixed directed arcs, and A is a set of pairs of alternative directed arcs. A graph selection S is a set of alternative arcs chosen from A such that at most one arc is selected for each pair. A feasible solution to the scheduling problem is a graph selection that is complete (exactly one arc from each alternative pair is chosen) and consistent (there are no positive length cycles in the graph).

In the alternative graph formulation, the CDRFR problem is to find 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. A positive length cycle represents a deadlock situation, i.e., an operation preceding itself. In general, negative length cycles allow to model more general scheduling situations [19]. To summarize, a conflict-free and deadlock-free schedule for the CDRFR problem is associated with a complete consistent selection in the alternative graph $\mathcal{G}(S)$.

An alternative graph gives all possible schedules once train routes have been fixed. When addressing the CDR problem, the set of routes can be changed by selecting a different set of fixed arcs (i.e., a different train route). The set F will be the decision variable taking care of the route chosen and $A = A(F)$ is a consequence of the choice for F. For a chosen route set F_i the solution to the associated CDRFR problem $S(F_i)$ is a complete consistent selection in $A(F_i)$. The solution to the CDR problem will be $(F, S(F))$.

All the relevant times associated with the running of a train in a track segment can be modelled in the alternative graph formulation by using blocking time theory (see e.g. [13]). The blocking time is the time interval in which a block section is exclusively reserved to a train and blocked for other traffic. A virtual blocking time overlap arises if the scheduled minimum headway distance between two train paths is not respected.

Several additional railway constraints may be easily included in the alternative graph model by a suitable choice of arcs and weights. Minimal connection time between different train services can be represented by an arc expressing the minimal required time distance between two operations. Coupling and decoupling of rolling stock can be represented by additional time relations between the services that are going to be split or combined.

Aggregation of block sections in complex station areas

A large amount of operations needs to be considered in the interlocking areas of complex stations. As a result, many variables and constraints affect the starting time of each operation. Due to the computational complexity of the problem, it is very difficult to compute suitable solutions to large scheduling instances in a short execution time. We next present an efficient procedure to aggregate information with the goal of reducing the number of decision variables and constraints.

An *aggregated block* is a sequence of consecutive block sections along the route that a train may traverse one after each other (see Figure 3.2). The sequence of corresponding operations $o_i, \ldots, o_m$ can be modelled as an operation of the aggregated block. The aggregated block operation is completely defined by the set of operations performed over the involved block sections. Precisely, the running time over an aggregated block is determined by the sum of running times associated to the involved block sections. The setup time over an aggregated block operation is computed as follows. At the time a given train running in the interlocking area clears the last block section of its aggregated block, all block sections of the aggregated block are released simultaneously, and thus become available to other trains. In this way, each train can start running over its aggregated block without being hampered by other trains.

When dealing with complicated interlocking areas the use of aggregated block operations allows to reduce the size of the problem. In fact, all the decision variables on the starting times of the individual operations in an aggregated block are taken into account at once. However, since too much aggregation of information leads to oversimplification, the procedure is to be restricted to the case when physical layout characteristics and operational rules constrain the individual operations to be related to each other. For instance, aggregating the whole path of a train into a single operation would incorrectly restrict the capacity of a railway corridor to be used by a single train at a time.

A convenient aggregation approach is to start and end aggregated blocks in correspondence to the main signals. In fact, when dealing with disturbances and short headway distances between trains, a red signal aspect may be shown, due to interlocking rules, to all the following trains having the same route or any other route with some shared block section. In this situation, the main signal would give a movement authority to each train only when all the block sections of its claimed route are free. In case of sectional release, block sections are released one at a time as the tail of the current train has cleared it.

This paper presents a more conservative approach that is based on the common dispatching practice for which a route is released only when the current train has left the last block section of its route. So doing, only one train at a time is effectively constrained to be scheduled in an aggregated block and complementary information about incompat-

ibilities between routes should be associated with the derived aggregated blocks. For a quantitative comparison of aggregated and disaggregated approaches we refer the reader to [7].

A similar aggregation procedure is performed when modelling the track between two stations as a simple link (this is usually applied for timetabling purposes). In this case, compatibility between different paths is enforced by a minimum fixed headway time between trains in conflicting routes. However, we consider a more accurate model of the interactions between trains in case of disturbed operations, i.e., we have to compute the speed trajectories of all the trains involved in route conflicts and their actual blocking times.

An example of aggregated blocks is described in Figure 3.2. We refer to an interlocking area connecting three platform tracks (1, 2, 3) to two open tracks (HA and HN). In Figure 3.2(a), there are nine block sections that are grouped into 6 aggregated blocks (HA-1, HA-2, HA-3, HN-1, HN-2 and HN-3), connecting each platform with each open track. The graph of Figure 3.2(c) depicts the incompatibilities between the six aggregated blocks, corresponding to the six nodes of the graph. Each arc connects two nodes that are incompatible. As only one train at a time is allowed to be scheduled in an aggregated block, each aggregated block is thus incompatible with itself. The other incompatibilities are given by the track layout. In total, there are eighteen incompatibilities out of all the possible relations, i.e., only three pairs of blocks are not mutually excluding each other (<HA-1, HN-2>, <HA-2, HN-3>, <HA-1, HN-3>).

As shown in the example of Figure 3.2, we propose an incompatibility graph [7] that is able to model incompatibilities between aggregated blocks that are due to the complex track topology of interlocking areas. A more compact representation relies on the introduction of *virtual machines* and on their association to aggregated blocks (see Figure 3.2). A virtual machine is a machine of the job shop problem that is used to keep track of the incompatibilities, such that any two aggregated blocks are compatible if there is no virtual machine associated to both of them. In other words, virtual machines represent all the incompatibilities between conflicting routes in a complicated interlocking area.

The number of virtual machines is given by the number of aggregated blocks that happen to be compatible with at least another aggregated block, i.e., the number of nodes of the incompatibility graph that are not connected with all the other nodes. The procedure adopted in this paper is to scan the incompatibility graph and search for the virtual machines needed to model all the incompatibilities between the aggregated blocks. The virtual machines are then introduced in the alternative graph formulation of the CDR problem, allowing to translate the characteristics of the non-aggregated model into the aggregated one.

In the illustrative example of Figure 3.2, all the incompatibilities between the six ag-

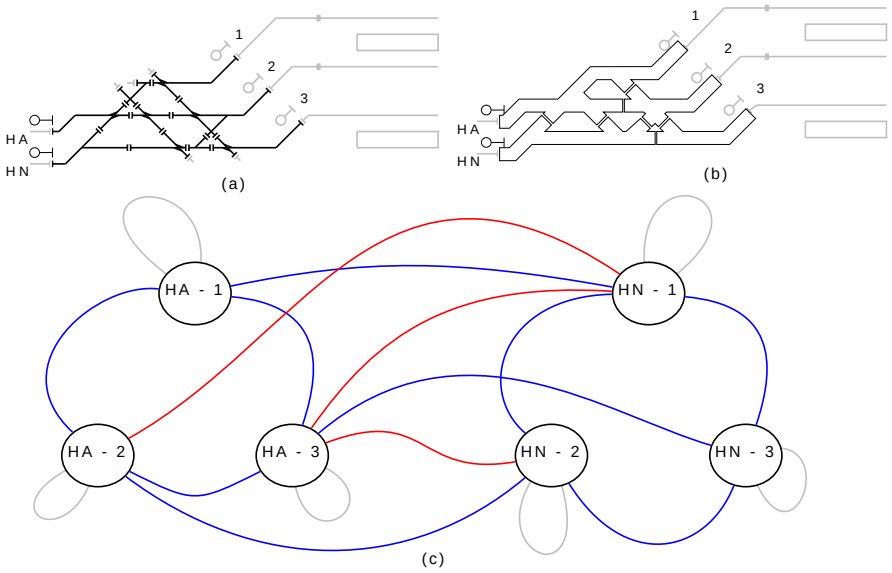


Figure 3.2: Block section and aggregated blocks: (a) block sections; (b) virtual machines associated to the aggregated blocks; (c) incompatibility graph between the aggregated blocks.

gregated blocks can be expressed by means of four virtual machines only, since the aggregated blocks compatible with each other are HA-1, HN-2, HA-2 and HN-3. In Figure 3.2(b), the virtual machines are represented as composition of block sections.

3.3.3 Decision support tool

The ROMA software is a decision support tool designed to assist traffic controllers in the evaluation of real-time dispatching solutions. ROMA is implemented in C++ language and uses the AGLibrary developed by the “Aut.Or.I.” Research Group of Roma Tre University.

For any situation of timetable deviation, such as multiple delayed trains, ROMA performs the following main procedures:

- Assigning a feasible route to each train such that there are no blocked tracks;
- Defining optimal train orders, routes, and times such as the exact arrival and departure times at stations, and the passing times at a set of relevant points (e.g. stations, junctions and the exit point of the network);

- Ensuring a minimum required time headway between the exit of a train from a block section and the entrance of the subsequent train into the same block section while maintaining acceptable speed profiles.

The three procedures are solved automatically by ROMA by means of the modules described in Figure 3.3. So far, the tool is a laboratory version and the field layer is only simulated. We next give a brief description of each module.

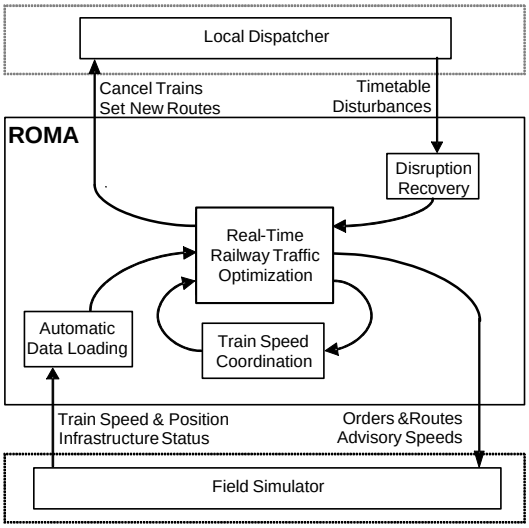


Figure 3.3: ROMA dispatching support tool architecture.

The `automatic data loading` module is in charge of collecting information regarding on-line positions and speeds of trains, infrastructure availability status, timetable, and rolling stocks. Accurate running and setup times are also computed for all the trains running in the network. The off-line information about the infrastructure layout is loaded directly from the InfraAtlas database in use at ProRail (the Dutch infrastructure manager). The timetable is translated from the DONS format by a dedicated automatic procedure. The data is stored in a format compatible with the current RAILML specifications that are a standard interface for railway data in Europe.

Regarding the on-line information, we suppose there are GPS sensors on board of trains such that information flows over a GSM-R channel to the dispatching control center. A more comprehensive method could be to combine GSM-R for actual train speed data and an automatic data mining tool, such as TNV-Conflict [8], for track occupancy and clearance data.

The `disruption recovery` module is in charge of providing a feasible route for each train when one or more block sections are unavailable to traffic, e.g. due to infrastructure disruptions. A list of train routes is given to this module, that select the alternative

most similar to the scheduled one. If no predefined route is available for a given train, the local dispatcher is asked to introduce new routes or cancel train services manually.

The `real-time railway traffic optimization` module is the optimization core of the ROMA tool. The module is able to exploit retiming, reordering and rerouting strategies. The first two degrees of freedom are tackled by solving a given CDRFR problem, while the third is addressed by the CDR problem. We next introduce the CDRFR and CDR algorithms used in this paper (a detailed description can be found in [6, 9, 10, 11]).

Several heuristic methods can be used to solve the CDRFR problem. A straightforward method is to impose the order prescribed by the timetable, another one is to follow the simple rule First Come First Served (FCFS), i.e., to assign priority to the train that claims the concerned block section first. In the Dutch dispatching practice an automatic route setting system, called ARI [3], is usually adopted as far as the delay is less than a predefined threshold. Specifically, the movement authority is given on the basis of the FCFS rule for tracks crossing each other while for merging tracks the orders specified in the timetable are followed. In case of larger delays, expert dispatchers take train ordering decisions directly with the support of a list of what-if scenarios. In order to implement the ARI system in an automatic way and for any kind of delay, we set up a list of priority rules for the latter case.

In order to perform an exhaustive search in the space of CDRFR solutions, a truncated Branch and Bound (BB) algorithm has been implemented [11]. This algorithm computes near-optimal solutions within a time limit of computation compatible with operations (a discussion about relevant times will be presented in Section 3.3.5). A good starting solution, or upper bound, is found by a set of heuristics for solving the CDRFR problem, such as the FCFS rule and other simple algorithms based on alternative graph properties.

A tight lower bound for the CDRFR problem based on the Jackson Preemptive Schedule [16] is computed for each (possibly virtual) machine. Using a level of description lower than the aggregated blocks improves the result by taking into account all the incompatibilities; on the other hand, the virtual machines have been introduced as the minimal set describing the incompatibilities, and hence are the best possible choice of job shop machines on which to compute the lower bound. The lower bound on each machine is computed in $O(z\ log\ z)$ steps, where z is the number of trains scheduled on the machine.

After computing a good solution for the CDRFR problem, a Tabu Search (TS) algorithm has been developed to search for alternative routes potentially leading to better schedules [6]. 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. Whenever a better schedule is found, the new route is set as default route and the search is repeated until a time limit of computation is reached.

Since the alternative graph model assumes deterministic blocking and waiting times, train trajectories are only feasible in absence of disturbances. In fact, the impact of deceleration and acceleration for hindered trains is not taken into account in the formulation of the CDR problem. The `train speed coordination` module is needed to ascertain whether a safe space headway between trains is respected and to update the speed profiles of trains according to the actual signal aspects. Speed coordination among consecutive trains is achieved by iteratively adapting the speed trajectory of trains and by updating the blocking and waiting times in the corresponding alternative graph, such that the resulting train schedules comply with the constraints of the signalling and safety system [13].

3.3.4 Interaction between decision support tool and dispatcher

The ROMA dispatching support tool computes train routes, orders and advisory speeds. However, before the implementation of the proposed actions the dispatcher receives a detailed forecast of the future traffic flow and train delays, and should recognize and acknowledge the changes in the timetable. The real-time interaction between the decision support tool and the dispatcher needs to be simple, clear and fast since the dispatcher has to decide which solution should be implemented among a set of possible solutions. In case of small perturbations, a dispatching solution could be presented in terms of the only relevant actions that differ from the scheduled ones, avoiding unnecessary corrections of the paths of on-time trains. In other more disturbed traffic conditions with multiple delayed trains, several timetable modifications may be needed to recover from delays and infeasible traffic situations. In this case, the dispatcher needs to be informed of the reasons for a particular modification of advisory speeds, arrival/departure times, and train routes and sequences.

Figure 3.4 depicts an interface between ROMA and the dispatcher. Blocking time graphs are useful to represent, visually, the future evolution of the train traffic. The main limitation is that only one corridor at a time can be investigated in sufficient level of detail (e.g. Figure 3.4 shows four virtual overlaps of blocking times for trains C15931 and A3031 that are running on the same line) while the visualization of route conflicts and route booking actions is quite complex in station areas. So, specific points of interest, where a train would experience knock-on effects, and the suggested rescheduling solutions are highlighted with a sufficient amount of detailed information (e.g. location of potential conflicts, suggested orders and routes, expected delay) to let the dispatcher understand the proposed dispatching actions and their impact on railway operations.

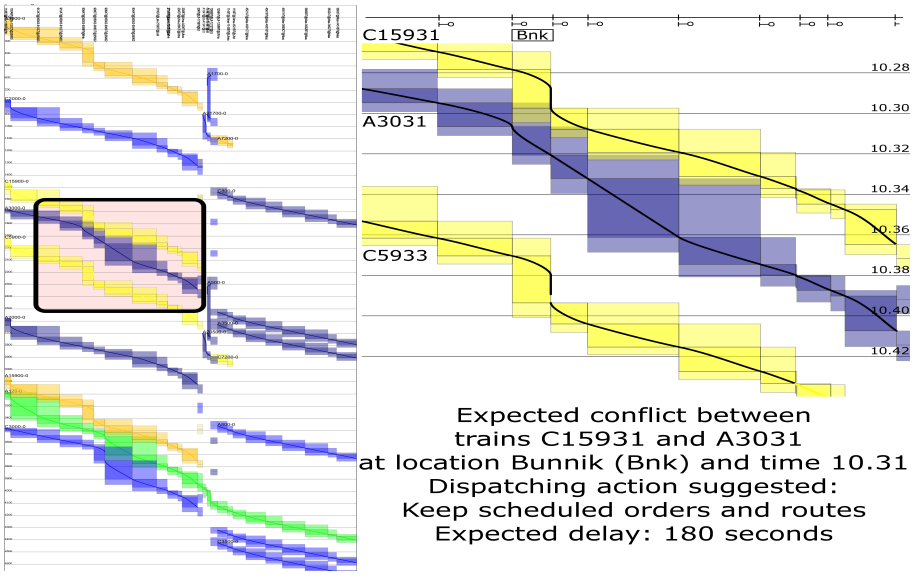


Figure 3.4: An interface based on the blocking time plot of the proposed control actions.

3.3.5 Real-time dynamic setting of the tool

This subsection discusses how our decision support tool could be used in a dynamic setting with operations. We observe that the applicability of the proposed dispatching solutions is influenced not only by the accuracy of the dispatching support tool in representing the actual and future traffic situations, but also by the reliability of the control actions (i.e., the ability to forecast their outcome). For these reasons, we next focus our attention on the following time intervals necessary to close the loop with operations (see Figure 3.5):

t_0 : time at which the current position and speed of each train are updated;

t_1 : time at which ROMA starts computing a dispatching solution;

t_2 : time at which ROMA returns a dispatching solution;

t_3 : time at which the ROMA solution is accepted by the dispatcher;

t_4 : time at which the dispatching actions are implemented;

t'_0 : next time t_0.

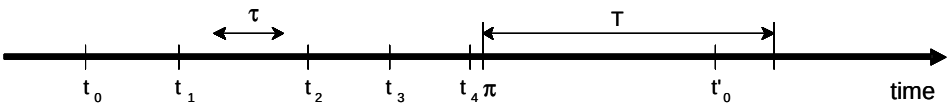


Figure 3.5: Time intervals to apply the dispatching support tool during operations.

The above times can be interrelated as follows. The time between t_0 and t_1 is needed to record actual train positions and speeds and communicate these to the traffic control center. At time t_0, we make use of the best prediction of all train positions and speeds obtained before that time. This can be transmitted via GSM-R on-board units of trains or by interpolating real-time train detection and signalling data. The time step between t_1 and t_2 is needed by ROMA to reconstruct the current traffic conditions, simulate the future evolution, detect possible conflicts and provide solutions. The time between t_2 and t_3 is used by the dispatcher to check the dispatching solutions given by ROMA and, eventually, to compare those with other dispatching options. The time between t_3 and t_4 considers the delay due to the transmission of the control actions as well as the time needed to implement the dispatching actions in practice, such as switching signals and setting up routes.

We now introduce the starting time π of the traffic prediction, the time horizon length T and the time τ to compute a dispatching solution, and describe how to set them up with respect to the other times. ROMA provides control actions in the time interval between π and π+T. It is assumed that no relevant unplanned action will occur from t_0 to π and the traffic flow in the network is determined exactly. However, an error between the simulated traffic and the actual traffic always exists due to the dynamic nature of the real-time operation and the inherent inertia of the dispatching process. If the dispatching support tool is not able to model the current status of the network with sufficient precision, the control actions suggested by the tool after π might be sub-optimal, obsolete or even infeasible. Moreover, the suggested actions would be physically applicable only if π is larger than t_4. From a practical point of view, the longer the interval t_4 - t_0 is, the larger is the error between the simulated and actual network status. To limit this error, the dispatching tool must achieve a sufficient precision in simulating the status of the system within the required computation and communication times.

Since the available time to compute a dispatching solution is rather limited, the time horizon of traffic prediction T may be also limited. So, the computation of downstream possible conflicts too far away in time should be avoided since the prediction uncertainty increases. However, the computation of optimal dispatching solutions requires to take into account global information regarding train traffic. To this end, the dispatching tool should be able to manage traffic in a large interconnected area with dense traffic.

Another important issue to study is the frequency of rescheduling. In fact, t'_0 is a variable time since the dispatching support tool may be run periodically or event-based (discussions on rescheduling under uncertainty can be found e.g. in [1, 15, 26]). In general, important parameters for choosing the frequency of rescheduling are the traffic prediction horizon T, the accuracy of the simulation procedure and the robustness against random variations in the dynamic traffic flow evolution. The more often the dispatching support tool is used, the less is the divergence between the train operations simulated by the tool

and the real traffic conditions. On the other hand, the dispatching support tool could be adopted when a particular condition triggers, i.e., when the error for an observed variable exceeds a given threshold or when an unplanned disruption has occurred and the current solution is infeasible.

A deeper analysis on the choice of the relevant times t_0, t_1, t_2, t_3, t_4 and their link with the time horizon T and the starting time of traffic prediction π would be necessary. To this end, experimental verification must still prove the applicability of the support tool in a real-world setting and the effectiveness and promptness of advanced dispatching measures compared to the current dispatching process.

3.4 Test case

In this section we present a comprehensive computational study to evaluate the potential of employing ROMA as support tool for traffic management in the dispatching area around Utrecht Central Station. We next describe the instances and present the obtained results.

3.4.1 Description of the instances

The topology of the railway network around the main station of Utrecht is similar to a star with 5 main directions crossing each other (see Figure 3.6). The main lines relaying the North and South regions of the Netherlands are connected to the lines to the West and the East. The network considered is delimited by the following stations: Utrecht Overvecht on the line to Amersfoort, Driebergen-Zeist on the line to Arnhem, Culemborg on the line towards Den Bosch, Vleuten on the line to Rotterdam and The Hague plus Maarssen on the line towards Amsterdam. In total, the diameter of the area is around 20 km long.

Utrecht Central Station is the most complex station area in the Netherlands and the interlocking area of the main station has, alone, around 200 block sections and more than 100 switches, leading to a large amount of inbound and outbound routes. In total, the considered area includes more than 600 block sections. In order to speed up the dispatching process, the whole network is transformed into around 200 aggregated blocks, with a total amount of around 250 virtual machines necessary to detect the incompatibilities between aggregated blocks.

Utrecht Central Station provides 20 platform tracks, most of which may be reserved to two trains at a time, for instance when coupling or splitting. Most of the platform tracks are used by through traffic, i.e., trains running in the opposite direction, even if some trains

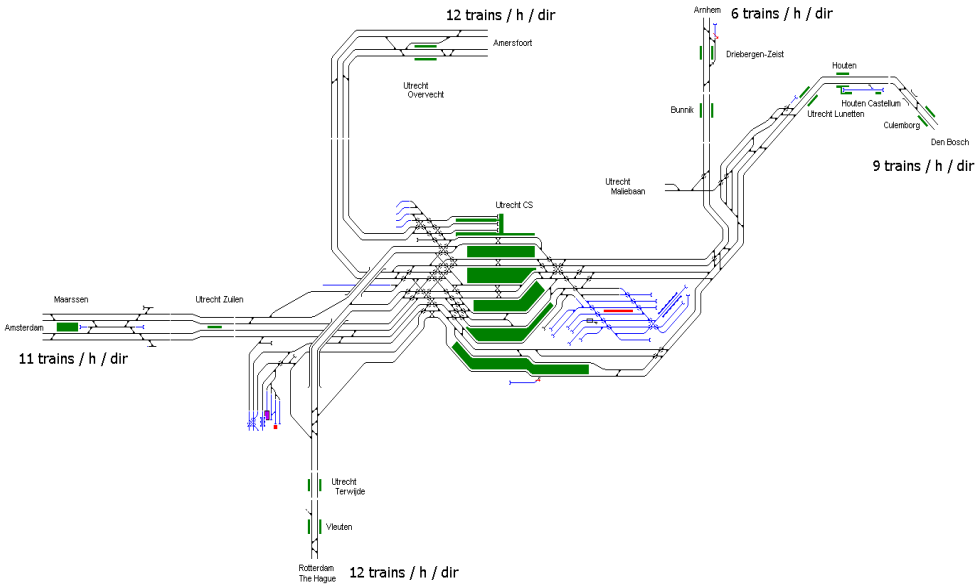


Figure 3.6: Utrecht Central Station in the center of the diagram; the scheduled hourly traffic per direction is given for each line of the dispatching area.

change direction after their stop at a platform. There are also three dead-end platforms.

For the computational experiments, we use the 2008 timetable that is periodic with a cycle length of one hour. The trains are mostly for passenger services, operated by NS (Nederlandse Spoorwegen), except for a few freight trains. The timetable schedules up to 80 trains in a peak hour and provides connections between passenger services, coupling and splitting of rolling stock for intercity and local services coming from/going to Rotterdam, the Hague or Amersfoort, as well as re-use of rolling stock for commuter services towards Utrecht Overvecht and Culemborg. The total amount of travellers at Utrecht Central Station is around 150.000 per day.

In order to limit the number of routings, we only consider two common routes for each train. So, the total number of routes is 160. Most of the alternative routes consider adjacent platforms since passengers' discomfort and extra time for transfer of passengers from one platform to another one are to be limited.

We generate timetable perturbations that cause initial delays in the network. To this end, we analyse a total amount of more than 33000 train events (arrivals, departures, dwell processes and passing times) that have been recorded at Utrecht Central Station in April 2008 by ProRail. We consider a statistical fitting procedure (as shown in [28]) to set the three parameters of the Weibull distributions, which are adopted to characterize the delays of different trains and the variation in the dwell time process (see Table 3.1).

Train Type	Disturbed Entrance Times			Dwell Time Extensions		
	Scale	Shape	Shift	Scale	Shape	Shift
Intercity	395	2.2	-315	252	2.1	4
Stoptrein	227	2.4	-198	252	2.1	4
Sprinter	235	3.0	-186	252	2.1	4
International	560	1.4	-205	252	2.1	4
Freight	1099	2.6	-885	252	2.1	4

Table 3.1: Parameters of the Weibull distributions.

Based on the calculated statistical parameters, we generate 40 random instances with disturbed entrance times and 15 random instances with dwell time extensions at Utrecht Central Station. Precisely, the average disturbance is around 30 seconds per train while the maximum disturbance is 685 seconds. These instances represent an average level of perturbed operations over a whole month. We combine each instance with the three infrastructure scenarios: *(i)* all infrastructure available; *(ii)* platform 2 in Utrecht unavailable for traffic; *(iii)* switch 1423A in Utrecht unavailable for traffic (so that platform 15 results blocked). In the given scenarios, respectively, 0%, 3% and 6% of the routes loaded into the dispatching support tool are unavailable, while 0%, 2% and 5% of the running trains have to be rerouted through the interlocking area, being still able to perform their scheduled trip with one of the available alternative routes. In total 1800 disturbance instances are tested with one hour of time horizon of traffic prediction (T) and with one minute of maximum time to compute a dispatching solution (τ).

3.4.2 Computational results

The main aim of the study is to assess how much train delays could be minimized by choosing suitable dispatching actions. We compare the dispatching solutions obtained by simple rules and advanced algorithms. Precisely, we test the following dispatching rules: the one that keeps the timetable order fixed, the local heuristic based on the FCFS rule and the ARI-like procedure. To achieve optimal traffic control actions by ROMA, we use the following two configurations: ROMA-reordering that adopts the BB algorithm to find near-optimal solutions to the CDRFR problem, and ROMA-rerouting that improves the ROMA-reordering solutions by the TS algorithm (i.e., computes better solutions to the CDR problem).

The average behaviour of the proposed dispatching procedures is shown in Table 3.2 in terms of computation time (in seconds), maximum and average consecutive delays (in seconds), average total delays (in seconds), percentage of on-time trains (named Punctuality since in the Netherlands a train is considered late if this has a total delay larger than

three minutes). Each row of the table corresponds to the average results over the 1800 instances.

Dispatching Procedure	Comp Time (s)	Max Cons Delay (s)	Avg Cons Delay (s)	Avg Total Delay (s)	Punctuality (%)
Timetable	5.8	622	50.1	94.5	83
ARI-like	5.7	446	28.2	74.3	84
FCFS	4.4	397	19	65.5	89
ROMA-reordering	5.7	296	15.1	61.2	91
ROMA-rerouting	52.3	299	14.6	60.8	92

Table 3.2: Average behaviour of the analysed dispatching procedures.

We now discuss the performance of the studied dispatching procedures on the basis of the results of Table 3.2. The solution found by keeping the train orders and routes as scheduled in the timetable has a poor quality. This is due to long waiting times for solving the route conflicts, causing a domino effect of delay propagation. The ARI-like procedure performs considerably better than the original schedule. However, the simple heuristic FCFS outperforms ARI. The use of the BB algorithm enables ROMA-reordering to achieve the best performance in terms of maximum consecutive delays. Specifically, the BB algorithm found the optimal solution to the CDRFR problem in 89% of the cases within the given time limit of computation.

When comparing the performances of ROMA-reordering to ARI, the average total delay experienced is reduced by around 18%, while a bigger reduction up to 48% is achieved in terms of average consecutive delays. With regards to the ROMA-rerouting procedure, we only note a small improvement compared to ROMA-reordering since few rerouting alternatives are explored in the given computation time (on average 54 CDRFR solutions are computed for each perturbated situation). On the other hand, significant results are obtained by ROMA-rerouting. Figure 3.7 shows the remarkable improvement of ROMA-rerouting versus FCFS in terms of average consecutive delays. The curve shows that ROMA-rerouting is, on average, around 20% better than FCFS, independently from the given initial delay.

Another important factor to consider is the number of reordering and rerouting actions needed to implement the dispatching solutions. We next compare the advanced procedures with the timetable solution. ROMA-rerouting changes around 2% of the train orders and 4% of the train routes. The other procedures also change at most 2% of the train orders. The limited number of modifications is due to the large number of instances with relatively small initial delay.

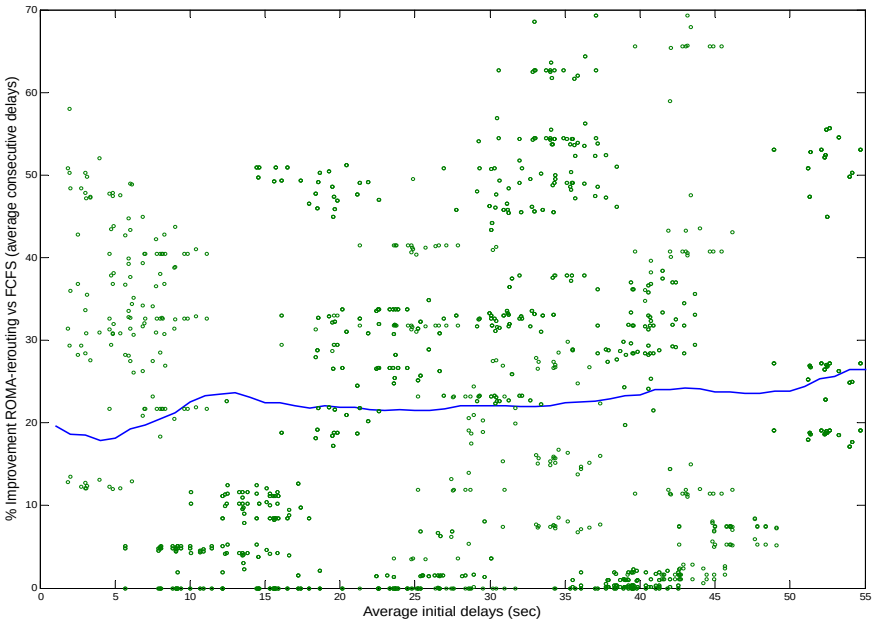


Figure 3.7: Scatter diagram improvement of ROMA-rerouting over FCFS, in terms of average consecutive delays, as function of the tested average initial delays.

3.5 Conclusions

This paper describes a laboratory train dispatching support tool that has been developed for the management of complicated and densely occupied station areas, i.e., many trains per hour dwelling at a large set of platform tracks. We use automatic scripts to convert the railway data (infrastructure, timetable and rolling stock information) from existing data formats supplied by the Dutch network infrastructure manager to the ROMA tool format. We also discuss the real-time use of the dispatching support tool, and generalize the applicability of earlier developed models and algorithms to manage distributed railway traffic in complex interlocking areas.

In order to assess the performance of the proposed algorithms, we propose a comprehensive set of experiments based on statistical distributions of train delays fitted to the realization data of a monthly period in the area of Utrecht Central Station. A large set of disturbances is generated and multiple control actions are required to restore feasibility of train operations and minimize the delay propagation. For the given set of instances, we investigate the computational effort and efficiency of advanced reordering and rerouting algorithms compared to simple dispatching procedures taken from traffic management practice. The tested instances aim at simulating the average behaviour in case of perturbed traffic for one month of operations. The reordering solutions computed by ROMA are significantly better than the ones obtained by keeping the scheduled orders, by the ARI-like procedure or by the simple FCFS heuristic. The benefit of rerouting traffic is not yet fully explored as the complexity of the scheduling problem still limits the computation time given to the rerouting task and only a limited number of alternative routes has been considered.

Future research should focus on the implementation of a closed-loop traffic monitoring and control system under the current safety regulations, and on studying the full benefits of dispatching trains in larger networks and for heavily disturbed operations.

Acknowledgements

We thank the Dutch infrastructure manager ProRail for providing the instances, D. Pacciarelli for fruitful discussions. This work is partially supported by the research program TRANSUMO “Reliable Transport Chains” and by the Italian Ministry of Research, Grant number RBIP06BZW8, project FIRB “Advanced tracking system in intermodal freight transportation”.

Bibliography

[1] H. Aytug, M. A. Lawley, K. McKay, S. Mohan, and R. Uzsoy. Executing production schedules in the face of uncertainties: a review and some future directions. *European Journal of Operational Research*, 161(1):86–110, 2005.

[2] C. Bailey. *European Railway Signalling*. Institution of Railway Signal Engineers, A & C Black, London, England, 1995.

[3] N. Berends and N. Ouburg. Beschrijving ARI-functionaliteit. Technical Report 20, ProRail Internal Specification (in Dutch), Utrecht, The Netherlands, 2005.

[4] G. Caimi, D. Burkolter, T. Herrmann, F. Chudak, and M. Laumanns. Design of a railway scheduling model for dense services. *Networks and Spatial Economics*, 9 (1)(1):25–46, 2009.

[5] M. Carey and I. Crawford. Scheduling trains on a network of busy complex stations. *Transportation Research, Part B*, 41 (2):159–178, 2007.

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

[7] F. Corman, R. M. P. Goverde, and A. D'Ariano. Rescheduling dense train traffic over complex station interlocking areas. In R.K. Ahuja, R. Moehring, and C. Zaroliagis, editors, *Robust and Online Large-Scale Optimization*, number 5868 in Lecture Notes in Computer Science, pages 369–386. Springer - Verlag Berlin Heidelberg, 2009.

[8] W. Daamen, R. M. P. Goverde, and I. A. Hansen. Non-Discriminatory Automatic and Distinct Registration of Primary and Secondary Train Delays. *Networks and Spatial Economics and Spatial Economics*, 9 (1):47–61, 2009.

[9] A. D'Ariano. *Improving Real-Time Train Dispatching: Models, Algorithms and Applications*. PhD Thesis, TRAIL Thesis Series T2008/6, The Netherlands, 2008.

[10] A. D'Ariano, F. Corman, D. Pacciarelli, and M. Pranzo. Reordering and local rerouting strategies to manage train traffic in real-time. *Transportation Science*, 42 (4):405–419, 2008.

[11] A. D'Ariano, D. Pacciarelli, and M. Pranzo. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183(2):643–657, 2007.

[12] A. D'Ariano and M. Pranzo. An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. *Networks and Spatial Economics*, 9 (1):63–84, 2009.

[13] A. D'Ariano, M. Pranzo, and I. A. Hansen. Conflict resolution and train speed co-ordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 8(2):208–222, 2007.

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

[15] K. Hozak and J.A. Hill. Issues and opportunities regarding replanning and rescheduling frequencies. *International Journal of Production Economics*, 2008. to appear.

[16] J. R. Jackson. Scheduling a production line to minimize maximum tardiness. Technical Report 43, University of California, Los Angeles, 1955. Management Science Research Project.

[17] J. Jacobs. Reducing delays by means of computer-aided 'on-the-spot' rescheduling. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, and S. Sone, editors, *Computers in Railways IX*, pages 603–612. WIT Press, Southampton, UK, 2004.

[18] M. Lüthi, G. Medeossi, and A. Nash. Structure and simulation evaluation of an integrated real-time rescheduling and train control system for railway networks. *Networks and Spatial Economics*, 9 (1):103–121, 2009.

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

[20] M. Mazzarello and E. Ottaviani. A traffic management system for real-time traffic optimisation in railways. *Transportation Research, Part B*, 41 (2):246–274, 2007.

[21] M. Pinedo. *Scheduling: Theory, Algorithms and System*. Prentice Hall, 1995.

[22] J. Rodriguez. 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, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

[23] A. Schrijver and A. Steenbeek. Dienstregelingontwikkeling voor Railned: Rapport CADANS 1.0. Technical report, Centrum voor Wiskunde en Informatica, Amsterdam, the Netherlands, 1994. In Dutch.

[24] P. Serafini and W. Ukovich. A Mathematical Model for Periodic Event Scheduling Problems. *SIAM Journal on Discrete Mathematics*, 2 (4):550–581, 1989.

[25] J. Törnquist and J. A. Persson. N-tracked railway traffic re-scheduling during disturbances. *Transportation Research, Part B*, 41 (3):342–362, 2007.

[26] G. H. Vieira, J. W. Herrmann, and E. Lin. Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of Scheduling*, 6(1):39–62, 2003.

[27] S. Wegele, F. Corman, and A. D'Ariano. Comparing the effectiveness of two real-time train rescheduling systems in case of perturbed traffic conditions. In J. Allan, E. Arias, C. A. Brebbia, C. Goodman, A. F. Rumsey, G. Sciutto, and N. Tomii, editors, *Computers in Railways XI*, pages 535–544. WIT Press, Southampton, UK, 2008.

[28] J. Yuan. *Stochastic Modelling of Train Delays and Delay Propagation in Stations*. PhD Thesis, TRAIL Thesis Series T2006/6, The Netherlands, 2006.

[29] P. J. Zwaneveld, L. G. Kroon, E. H. Romeijn, M. Salomon, S. P. M. Dauzère-Pérès, S. Van Hoesel, and H. W. Ambergen. Routing Trains Through Railway Stations: Model Formulation and Algorithms. *Transportation Science*, 30:181–194, 1996.

Chapter 4

Centralized versus distributed systems to reschedule trains in two dispatching areas

Abstract *Railway dispatchers are in charge of rescheduling trains during operations in order to limit propagation of disturbances occurring in real-time. To help the dispatchers in such task, an advanced decision support system, ROMA (Railway traffic Optimization by Means of Alternative graphs), has been recently implemented to optimize railway traffic within a single dispatching area. This paper presents a novel distributed optimization system to control trains running in a Dutch railway network that is divided into two complex dispatching areas with dense traffic, each one controlled by a single dispatcher with the support of a local ROMA. A coordination level is introduced in order to manage the interaction among the two local ROMAs. An extensive computational assessment of the centralized and distributed systems is performed by using simple and advanced train scheduling algorithms, including dispatching rules adopted during operations. The effectiveness of the distributed system is shown in terms of computation time and delay minimization for practical statistical entrance delay distributions and in presence of an increasing number of blocked platforms in the main station area.*

This chapter is an edited version of the article

F. Corman, A. D'Ariano, D. Pacciarelli, M. Pranzo. Centralized versus distributed railway traffic control. In *Proceedings of the Eleventh international Conference on Advanced Systems for Public Transport (CASPT09)* (editors: Lo, Lam, Wong, Leung). Hong Kong, China, 2009, and accepted for publication in Public Transport: Planning and Operations, 2010.

4.1 Introduction

The traffic control of nation-wide railway networks is usually managed by a set of regional traffic control centers. For instance, the control of the 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 controlling several dispatching areas.

In each dispatching area there is a dispatcher who receives real-time information on the route, current location and speed of each train, and the status 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 going to occur and solves them on the basis of experience and rules. Possible control actions include changing dwell times at scheduled stops or train orders at junctions, stations and passing points. Other control actions involve major modifications such as route change or cancelation. The main goal of dispatchers is the minimization of train delays even if dispatchers also take into account effects on the passengers.

The traffic control is often hierarchically organized in at least two decision levels. At the lower level there are line dispatchers, who control dispatching areas with a local view of the traffic flow. At the higher level the network dispatchers are responsible for the coordination of the rescheduling decisions taken by several line dispatchers with a global overview of the traffic flow.

This paper deals with the development of Decision Support Systems (DSSs) for real-time railway traffic management of railway networks composed by multiple dispatching areas. The problem of coordinating the decisions taken by dispatchers is quite underinvestigated in the literature on railway traffic management and, to the best of our knowledge, this paper is one of first attempts to deal with this problem. Therefore, we limit our investigation to the case of two dispatching areas coordinated by a network dispatcher.

Decision support systems are being developed by railway companies and research institutions to support the rescheduling process. Such automated systems are mostly based on centralized procedures that must deliver viable solutions in a short computation time. However, real-time train rescheduling is a difficult NP-hard problem [5] and the best solvers available are able to solve to optimality instances at the level of a single dispatching area. Effectively managing train traffic in a network with multiple areas is still a challenging problem, since the computation time of existing exact rescheduling algorithms may increase dramatically with the size of the network.

The computational complexity of the train scheduling problem can be limited to an acceptable level by decomposition, since local decisions are taken in a parallel fashion and

independently from each other. However, little research addressed the problem of automatically coordinating independent DSSs, each controlling a single dispatching area. The main issue of a distributed approach is therefore the coordination between the solutions of the sub-problems, since the aggregated solution must be globally feasible (see, e.g., [6, 35]).

Global feasibility requires that a number of constraints at the borders between dispatching areas is satisfied. The exit time of a train from an area must be compatible to the entrance time in the subsequent area. Train orders and routes at the border should be consistent and locally feasible solutions should not cause deadlock situations from a global perspective. Such constraints cannot be satisfied when solving the sub-problems individually, since each sub-problem has a myopic view of the entire process. Coordination action is therefore necessary and consists either in the modification of local solutions or in the addition of specific constraints to each sub-problem, so that the local solution are forced to be globally feasible.

This paper proposes a distributed framework for railway traffic control in a network of two areas. The centralized decision support system, ROMA (Railway traffic Optimization by Means of Alternative graphs) [7, 8, 10] is integrated into the distributed framework. Each dispatching area is managed by a local ROMA while the problem of coordinating adjacent areas is solved with a distributed approach. Specifically, the coordination procedure alternates a solution step, in which each local ROMA produces a solution for its local area, to a coordination step. In the latter step, each local ROMA receives information on the traffic flow at the border between areas and uses it to produce a solution compliant with that of the other area. In order to exploit the potential of distributed optimization, concurrent and parallel execution is supported by a standard communication protocol [22].

Extensive computational experiments are carried out in order to compare the performance of centralized and distributed approaches on a practical test case composed by two dispatching areas in The Netherlands. The network is a complicated and densely occupied network around Utrecht Central station, with an hourly timetable of about 80 trains per hour. Different algorithms are compared to solve the line dispatcher problem, including an exact optimization algorithm and two scheduling rules frequently used in the railway practice. Instances include multiple delayed trains and disruptions with an increasing number of blocked platforms in order to assess the performance of the centralized and distributed systems when the complexity of the instances increases. The solution quality is assessed in terms of maximum and average train delays. The computation time of both approaches is also investigated.

The main contributions of this paper are fourfold:

- A methodology is given to check the global feasibility of local schedules based on aggregated information;

- A schedule coordination procedure is introduced and tested;
- The two-level coordinated traffic control is assessed for the first time on a real network composed of two dispatching areas with dense traffic;
- A systematic study of the distributed and centralized approaches is provided for increasing traffic disturbances.

The paper outline is the following. Section 2 defines the railway terminology used throughout the paper. Section 3 provides an overview of the existing literature on train rescheduling. Section 4 briefly describes the centralized system and the mathematical model and procedures for train scheduling. Section 5 introduces the distributed system, including models and algorithms for train schedule coordination. Section 6 reports on the computational results on a real-world test case from the Dutch railway network, and deals with a variety of disturbed traffic situations. Section 7 concludes the paper and gives directions for further research.

4.2 Problem Definitions

In its basic form a railway network is composed of stations, links and block sections separated by signals. Signals control the train traffic on the routes and are located before every junction as well as along the lines and inside the stations. A block section is a track segment between two main signals and, for safety reasons, may host at most one train at a time. Signals, interlocking and Automatic Train Protection systems (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 main characteristics of most railway signaling systems is the fixed block signaling system. A train may have movement authority to enter a block section only after the train ahead has completely left it and the ATP system releases the block section. In this paper we consider the Dutch three-aspect fixed block signaling system, in which a signal aspect may be red, yellow or green. However, the discussion holds for most fixed block signaling systems. A red signal aspect means that the subsequent block section is either out of service or occupied by another train, 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. Descriptions of railway signaling systems and traffic control regulations can be found, e.g., in [12].

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 during a *service* (train run). At any time a train route is feasible if all its block sections are available for traffic. 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*. The running time is known in advance since all trains travel at their scheduled speed, which usually contains some margins to recover from small delays.

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 includes the interval 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 block section and to cover the sighting distance (see, e.g., [13, 24, 25]).

The timetable describes the movement of all trains circulating in the network by 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 can be included, such as minimum transfer times between connected train services.

Unexpected events occurring during operations may cause delays which make the timetable infeasible. The delay may propagate causing a domino effect of increasing disturbances. We define an *entrance perturbation* as a set of delayed trains at the entrance in a dispatching area, due to the propagation of delays from previous dispatching areas. An *infrastructure disruption* is the unavailability of one or more block sections, which causes alterations in the train travel times and routes. 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 railway traffic management copes with the temporary 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. The task of dispatchers is to regulate traffic with the main objective of minimizing train delays in such a way that the new schedule is compliant with railway rules and with the actual position of each train. The latter information enables the computation of the *release time* of each train with respect to the starting time t_0 of traffic

prediction, which is the expected time at which each train enters its first block section in the area under study.

The *total delay* is the positive 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* is caused by original failures and disturbances and can only be recovered by exploiting available running time reserves, i.e., with trains traveling at maximum speed. The *consecutive delays* are the additional delays generated by the dispatching measures taken in response to initial delays.

A potential *conflict* occurs when two or more trains claim the same block section simultaneously, and a decision on the train ordering has to be taken. A set of trains causes a *deadlock* when each train in the set claims a block section ahead which is not available, due either to a disruption or to the occupation/reservation for another train in the set.

4.3 Overview of the Literature

This section gives an overview on the literature on train rescheduling. In general, automated rescheduling systems are designed to support traffic controllers in the management of railway traffic in a given railway area and within a short time limit of computation. An important objective is the minimization of train delays at stations as well as at the exit from the area. Other research directions focus on the delay minimization for passengers and goods. Several factors influence the solution quality, including the level of detail considered to formulate the problem, the size and complexity of the studied area, the density of traffic, the type of disturbance and the length of the time horizon of traffic prediction. This factors play an important role and need to be carefully chosen in accordance with the chosen solution approach.

We classify the variety of approaches presented in the literature discussion in two groups: centralized and distributed systems. The first group has a single core in charge of deciding the rescheduling actions while the second group is organized in entities (agents, modules or subsystems) that take decisions and need to be coordinated.

Table 4.1 reports a list of recent contributions on centralized train rescheduling. We analyze three important factors: the level of detail (MA = macroscopic or MI = microscopic), the characteristics of the studied area (SL = Single-track Line, DL = Double-track Line, J = Junction, DA = Dispatching Area or N = Network with multiple dispatching areas) and the solution methodology. All the studied approaches consider realistic railway networks.

In general, macroscopic approaches model train movements and headway times without considering explicitly all the safety regulations. A train run on the track between two

Table 4.1: Recent contributions on centralized train rescheduling

Paper	Approach	Size	Solution Method
[1]	MA	DA	Local heuristic
[9]	MI	DA	Heuristics + branch & bound
[14]	MA	DL	Local heuristic
[16]	MI	N	Simulation + priority rules
[27]	MA	DL	Genetic algorithms
[28]	MI	J	Constraint programming + branch & bound
[29]	MI	SL	Look-ahead heuristic
[33]	MI	J	Simulation + genetic algorithms
[34]	MA	N	Mixed integer linear programming
[37]	MI	N	Simulation + genetic algorithms

stations is described by a running time on the overall track and a minimum headway between successive trains. Differently, microscopic approaches control trains at the level of block signal and offer more detailed information about train movements and actual state of the signaling system along the tracks. Detailed information are used regarding track layout, switches and signals in order to compute accurate running and headway times. Clearly, the latter approaches require detailed models and a larger computation time to compute train schedules compared to the former approach.

A main drawback of most centralized approaches is that they are heavily affected by the size of instances. A trade-off must be found between the size of the studied area and the time horizon of traffic prediction. In fact, the problem complexity increases with the number of trains, tracks and stations. If a small test case is considered, only few trains and few conflicts can be detected and solved. Thus, solutions computed with a limited time horizon of traffic prediction are myopic, since the rescheduling process does not consider conflicting trains outside the studied area and time horizon [10].

Concerning the solution method and quality, many approaches combine a simulation model with simple decision rules. The performance of such DSSs can be quite erratic. Searching for optimal solutions may result in exceedingly large computation time, so when increasing the size of the problems some speed-up is crucial in order to come up with at least a good quality solution within a reasonable time.

A time-effective approach consists of handling large problems by decomposing them into several smaller, well-structured, weakly interrelated sub-problems. The sub-problems are then solved via effective algorithms. Finally, the coordination issue among the different subsystems is addressed and solved, delivering good quality dispatching solutions in a timely manner.

Table 4.2 reports a list of contributions on distributed train rescheduling. We analyze four

Table 4.2: Recent contributions on distributed train rescheduling

Paper	Approach	Decomposition	Structure	Solution Method
[4]	MA	Junctions	DEC	Local heuristic
[15]	MA	Trains, Stations	DEC	Resource allocation
[17]	MA	Areas, Network	HIER	Local heuristic
[18]	MA	Stations, Areas	DEC	Local heuristic + priority
[19]	MA	Trains, Stations	DEC	Resource allocation
[21]	MI	Areas, Network	HIER	Local heuristic
[23]	MA	Trains	DEC	Local heuristic + priority
[26]	MA	Trains	DEC	Resource allocation + priority
[30]	MA	Trains, Stations	DEC	Local heuristic
[32]	MI	Areas, Network	HIER	Local heuristic

important factors: the level of detail (MA or MI), the entities involved in the decomposition (junctions, trains, stations, areas and/or network), the structure of the decomposition (DEC = Decentralized or HIER = Hierarchical) and the solution methodology.

With a distributed approach, different physical or logical entities, like trains, block sections or infrastructure managers, may have a certain decisional capacity and negotiate with other entities for the access to shared resources. Decisional entities may be organized at the same or at different hierarchical levels. In the latter case, entities at the higher level guide the behavior of lower level entities and the final solution can be reached after a bottom-up or a top-down negotiation. With a bottom-up negotiation, the low level entities take the main decisions and a high level entity checks the decision and, if necessary, asks for modifications to the low level entities. With a top-down negotiation, the high level entity takes the main decisions on the basis of some aggregated model and leaves detailed decisions to the low level entities. The latter may ask for modifications, for example if no feasible solution can be found given the decisions of the high level entity.

The timetabling problem is typically solved through a top-down decomposition in which tentative timetables are first produced at the higher level and then checked for feasibility on the basis of detailed models. A well-known example of this approach is given by the Dutch system DONS [36], in which CADANS [31], produces cyclic timetables at nationwide level based on macroscopic models and then STATIONS [39] assigns platforms and routes to trains in a railway station based on a more detailed model.

A recent top-down approach to timetabling is proposed by Caimi et al. [3]. With this approach, train service intentions are first generated at the higher level and then a detailed network-wide timetable is built at the lower level by decomposing the network in condensation and compensation zones. Condensation zones are main station areas while compensation zones are open tracks between stations. Detailed timetables are produced in

each zone almost independently from each other. The notion of portals is then introduced in order to coordinate the passing times of trains at the border between condensation and compensation zones.

Differently from timetabling, distributed train rescheduling is typically approached with a bottom-up hierarchical approach [17, 21, 32] or with a single level decentralized approach (see Table 4.2). In fact, the rescheduling problem is heavily constrained locally by the current train positions and by the existing timetable. Therefore, the main decisions have to be taken locally, while higher level decisional entities address the global feasibility issue of local solutions.

On the whole, in the literature on distributed approaches to train rescheduling still few decision support systems are able to quickly and effectively reschedule train movements in practical networks. Despite the efforts devoted to developing sophisticated dispatching procedures, most of the existing approaches suffer from a lack of a general methodology or are designed to solve fictitious instances of traffic flow perturbation. Specifically, from Table 4.2, among the single level decentralized approaches only Lee and Gosh [19] study a practical network, while among hierarchical approaches only Mazzarello and Ottaviani [21] and Jia and Zhang [17] report on practical experiences.

At least five critical requirements for a good system operability can be envisaged: (i) DSSs should be able to compute dispatching solutions within limited time; (ii) DSSs for line dispatchers should be able to recover local feasibility in presence of multiple train delays or disruptions, i.e., the adjusted timetable must be compliant with the actual train positions and infrastructure status; (iii) DSSs for network dispatchers should be able to check rapidly the local solutions for global feasibility; (iv) DSSs should provide good quality solutions also when dealing with complicated and dense railway networks. (v) DSSs should operate with accurate input data. If the quality of input data is poor, it is unlikely that the solutions provided by the DSSs will be effective in practice. A critical prerequisite of good DSSs is therefore the use of reliable methods for traffic prediction and data estimation.

This work contributes at filling the gap between theory and practice of train rescheduling. Specifically, we refer to the first four critical requirements described above. The basic methodology for railway traffic regulation and coordination is based on the ideas recently developed within the European project COMBINE 2 [6] and then exploited by [21, 32]. In this paper, a systematic study of the hierarchically distributed framework is provided and a comparison is carried out with the centralized approach. Differently from previous papers, an exact algorithm is used for rescheduling trains and difficult practical instances are utilized for the assessment.

4.4 Centralized Rescheduling System

This section describes the decisional kernel of the dispatching support system ROMA, that we use as a DSS for the line dispatcher. ROMA detects and solves train conflicts while minimizing the maximum and average consecutive delay in lexicographic order [8]. In this study, we use the rerouting function of ROMA only for the purpose of disruption recovery and not in the optimization module. This choice limits the need for coordination between dispatching areas to scheduling decisions only. Rerouting decisions are taken by the network dispatcher as a preliminary decision before the scheduling phase. The decisional kernel works as follows. For a given set of real-time disturbances, 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 orders and times for all trains at each block section.

The underlying model used by ROMA to formulate the conflict resolution problem is a job shop with no-store and no-swap constraints. This problem is formulated with an alternative graph [20] and by using the blocking time theory to compute arc weights (see, e.g., [25]). The main value of the alternative graph is the microscopic and flexible modeling of the network topology at the level of railway signal aspects and operational rules.

We next summarize the main aspects of the alternative graph formulation used in this paper. A more detailed description can be found, e.g., in [5, 8]. In case of fixed block signaling, each pair [train, block section] corresponds to a node in the alternative graph and the arcs between nodes represent precedence constraints. Other constraints relevant to the railway practice can be included into the model (see, e.g., [7, 9, 11, 10]). A train route is viewed as a job in the alternative graph. A job is a sequence of operations, i.e., block sections that must be traversed sequentially by the train. A fixed running time for each operation is known in advance, that is the minimum time occurring between the starting of consecutive operations. A possible additional waiting time can be necessary in order to solve train conflicts. A train schedule defines 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 that is modeled in the graph by introducing a suitable pair of alternative arcs for each pair of trains traversing the same block section. A deadlock-free and conflict-free schedule is 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 order to evaluate a train schedule, we use the maximum consecutive delay as the main performance indicator of a solution. The maximum consecutive delay can be computed as the length of the longest path in the graph when exactly one arc from each alternative pair is selected.

4.4.1 Illustrative Example

Let us illustrate the construction of the alternative graph for a small railway network with 9 block sections and three trains (A, B and C). The route of train A is given by the sequence of operations $A1$, $A4$, $A5$, $A6$, $A7$, $A9$. The route of train B is $B2$, $B4$, $B5$, $B6$, $B7$, $B8$ while the route of train C is $C9$, $C7$, $C6$, $C5$, $C4$, $C3$. The three trains share the same path from block section 4 to 7. Additionally, trains A and C share the block section 9. Trains B and C are slow trains and enter their first block section at release time 0 and 100, respectively. Their minimum running time on each block section is 20 time units. Train A is a fast train that enters its first block section at release time 210 and runs on each block section of its route with a minimum running time of 10 time units. All the setup times are fixed to 10 time units. There are only three points that are relevant for the delay computation, namely the block sections 3, 8 and 9. Trains A, B and C are scheduled to leave the network at exit time 270, 120 and 220. There is therefore no extra time added to the travel time of each train in the network.

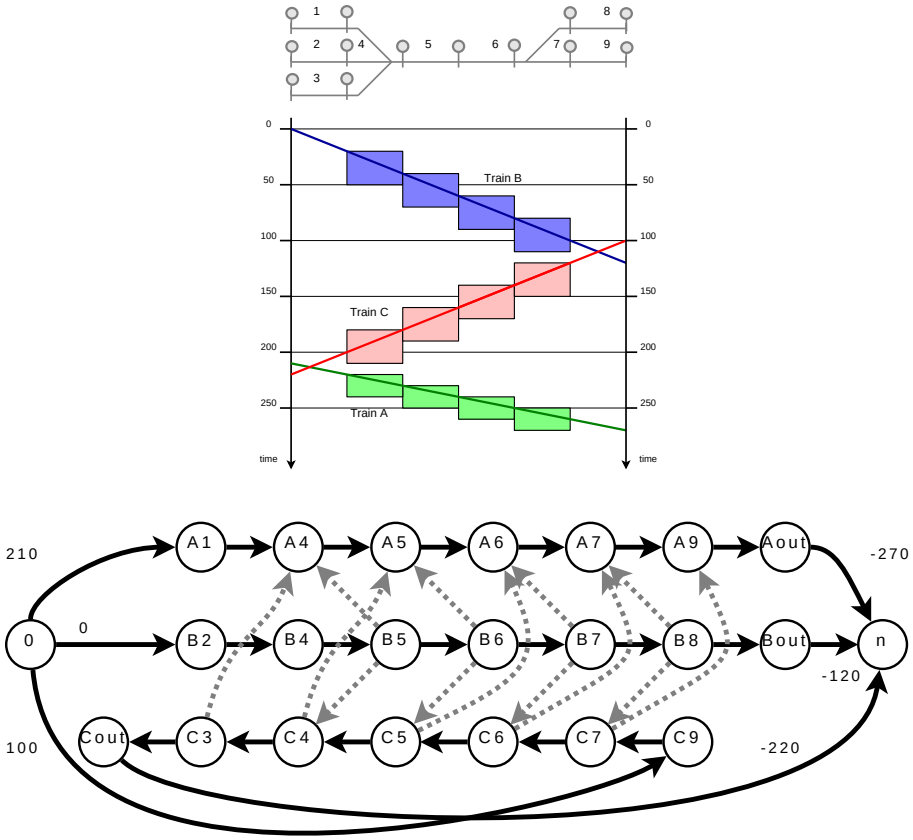


Figure 4.1: Timetable solution: space-time diagram (top) and alternative graph (bottom)

Figure 4.1 (top) shows a space-time diagram of the timetable solution. For clarity, we

only show the running and setup time of the relevant block sections. Figure 4.1 (bottom) then presents the corresponding alternative graph formulation. 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. The selected alternative arcs are represented by dashed arcs. For clarity, the running and setup times are not depicted in the alternative graph. 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. Since the initial delay is zero, the weight of the latter arcs is equal to the exit time of the associated trains from the network as specified in the timetable (but with negative weight). Since there are no delays, the longest path in the graph has length equal to 0.

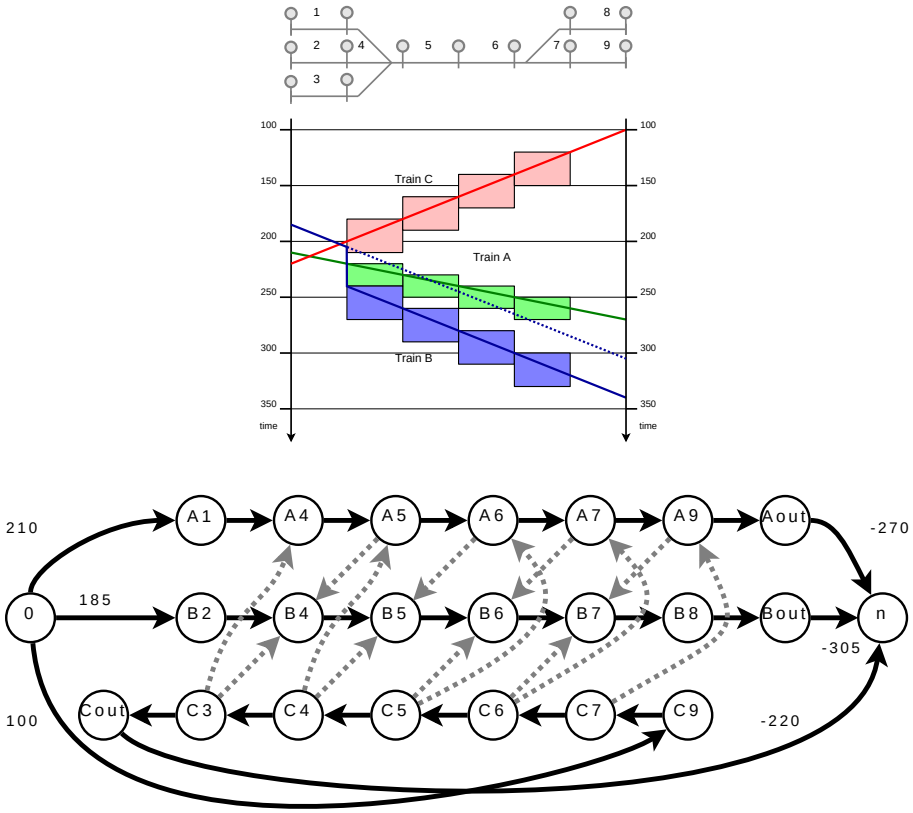


Figure 4.2: Optimal train scheduling solution to the disturbed situation

Figure 4.2 (top) shows the space-time diagram of the optimal solution to the real-time railway traffic management problem for a disturbed situation in which the entrance of train B is delayed by 185 time units. The dotted line for train B represents its free running path, computed by considering its new entrance time and disregarding the presence of the other trains. The new exit times are 270, 220 and 340 for trains A, B and C, respectively.

Figure 4.2 (bottom) presents the optimal solution in terms of alternative graphs. Since train B is delayed, its release time and its minimum exit time are updated in the graph.

Furthermore, there is an additional waiting time for train B to enter block section 4 that causes a maximum consecutive delay of 35 time units. The other trains are on time.

4.4.2 Train Scheduling Procedures

Given a timetable, the entrance position of each train in the network and the current status of the infrastructure, each train scheduling procedure implemented in ROMA is in charge of computing quickly a feasible schedule for each train, i.e., defines a feasible entrance time of all trains on each block section. The following three scheduling procedures are tested in this paper:

- First Come First Served (FCFS): This is a well-known dispatching rule which gives 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 in the timetable.
- ARI (Automatische Rijweg Instelling): This rule is a fully automated version of the route setting procedure used in the Netherlands [2]. Decisions on train orders are taken every time a conflict is detected. If both conflicting trains have an entrance delay below 3 minutes the train order is assigned according to the timetable for conflicting trains requiring the same track and according to the First Come First Served rule for conflicting trains requiring different incompatible tracks. If at least one of the conflicting trains is delayed by more than 3 minutes, conflicting trains are scheduled on the basis of train priorities: first intercity trains, then local trains and finally freight trains. For trains with the same priority, precedence is given to the train with the smallest number of scheduled stops after the conflicting point.
- Branch and Bound (BB): This is an exact scheduling algorithm that explores all the reordering alternatives and chooses the one minimizing the maximum consecutive delay. Here, we consider a truncated branch and bound [9] that returns near-optimal schedules for practical train scheduling problems within a computation time compatible with operational needs.

The resolution of the train scheduling problem formulated by alternative graphs requires a computation time that is rapidly increasing with the dimensions of the graph. Precisely, the number of disjunctions considered (alternative arcs) increases quadratically with the number of trains and linearly with the number of shared resources (block sections or stopping platforms). Furthermore, the increase in computation time is different for the various algorithms considered. The computation time of FCFS and ARI increases polynomially in the number of variables, while the one of the exhaustive search increases exponentially in the worst case.

4.5 Distributed Rescheduling System

This section addresses the problem of controlling the railway traffic in large networks. In this paper, large network means that a single centralized ROMA system is unable to control the traffic in the whole network within the short time available for computation. In what follows, a large railway network is partitioned by traffic controllers on a geographical basis, since the decomposition of the network into smaller areas should be performed statically. The ideal size of each area clearly depends on the expected traffic pattern in the controlled area, and this parameter has effects on the solution quality.

The following subsections describe the architecture of the distributed system, introduce the aggregate formulation adopted to detect global infeasibilities, present a novel procedure to coordinate the local scheduling solutions.

4.5.1 System Architecture

Figure 4.3 shows the distributed system architecture in terms of actors involved and information flow. The system is designed to support hierarchically the work of two line dispatchers and a network dispatcher.

Each area is controlled by a local ROMA, and two local ROMAs exchange information about their solutions, coordinated at a higher level. If a local ROMA is not able to compute a feasible schedule, a *local infeasibility* is found, i.e., conflicts and/or deadlocks that involve more trains over the local area. In case of local infeasibility, the line dispatcher is asked to take control actions. On the other hand, the coordination procedure may fail, resulting in a *global infeasibility*, i.e., conflicts and/or deadlocks that involve more trains over several local areas. In this other case of infeasibility, the network dispatcher is asked to recover the situation. This is achieved by imposing additional constraints at global level to the coordination level of the dispatching system or by communicating new control actions to line dispatchers.

The coordination level is in charge of coordinating the solution processes by exchanging relevant local information while detecting and solving global infeasibilities. Only aggregate data describing the interactions between trains running in adjacent areas are sent to the coordinator, that has to solve global infeasibilities (e.g. by changing train orders at the borders) and to transmit quickly the border constraints to the local ROMAs. If no global infeasibility is found, the coordination results in exchanging the relevant border information between the areas. A compact representation of variables and constraints of each area limits the size of the set of data to be managed, avoiding the coordination level to become the bottleneck of the procedure.

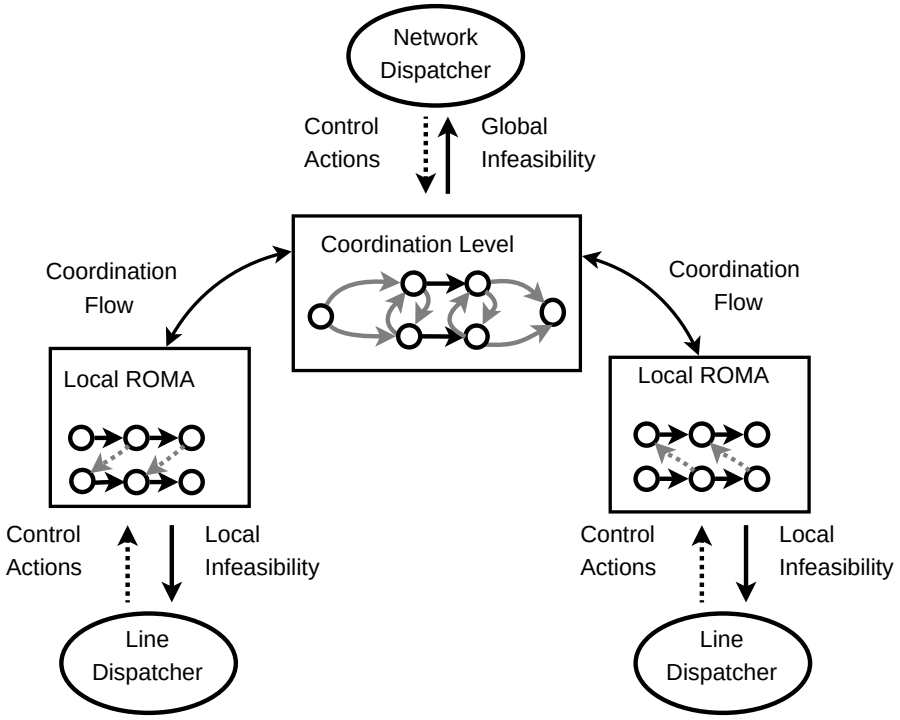


Figure 4.3: Framework of the distributed rescheduling system

A communication protocol has been implemented between the local ROMAs and the coordination level. Reliable and fast exchange of messages between the involved actors is necessary to achieve a synchronized and coordinated behavior. To this end, a standard protocol for high performance computing, Message Passing Interface [22], is used to effectively support concurrent and parallel execution of processes, as well as inter-process communication.

4.5.2 Formulation with Alternative and Border Graphs

We now describe the mathematical models for solving and coordinating the scheduling solutions of local solvers (i.e., local ROMAs). A limited amount of local information is exchanged in order to ensure that a set of locally feasible schedules is globally feasible. The information sent by the local ROMAs to the coordinator is extracted from the alternative graph representation of the traffic flow in each local area. For each train passing from an area to another one we consider *border nodes* in the alternative graph. These nodes are associated to the operations representing trains crossing the borders between areas. Note that each block section at the border between two areas is included in both areas. Similarly, the associated border nodes are duplicated.

Figure 4.4 (top) shows the space-time diagram of a solution to the real-time railway traffic management problem for the disturbed situation of the illustrative example in which train B is delayed by 185 time units. The proposed solution is obtained by dividing the railway network in two local areas, computing the optimal solution for each local area, checking the global feasibility of the local solutions. The three trains (A, B and C) traverse both the local areas, called x and y. Train A and B run from area x to area y while train C runs from area y to area x.

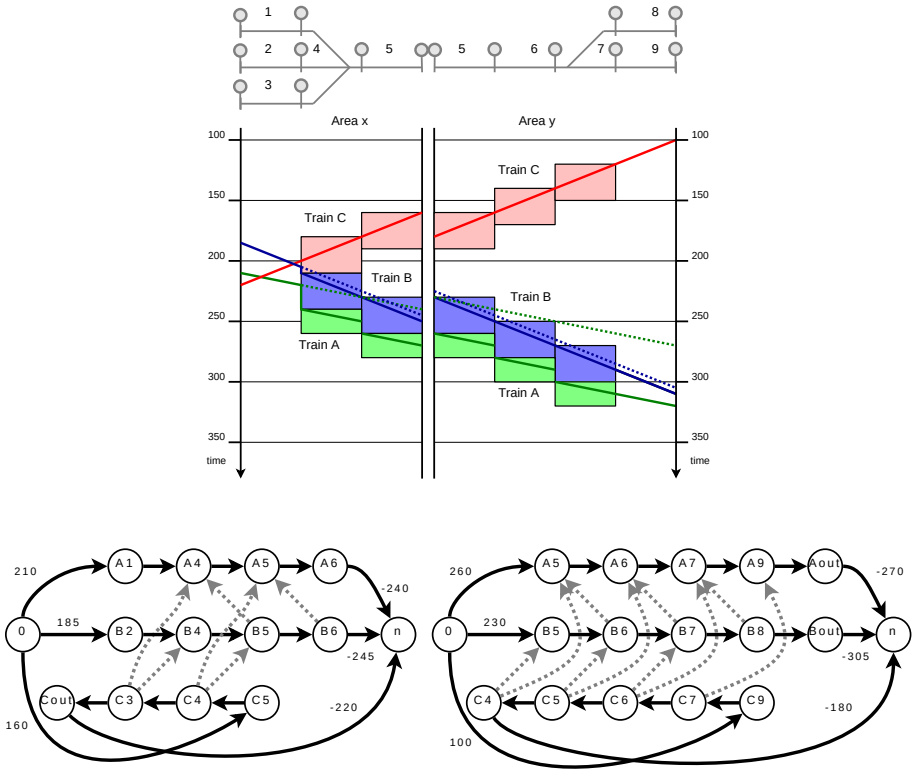


Figure 4.4: Optimal local solutions but sub-optimal global solution

Figure 4.4 (bottom) reports the local alternative graphs of the proposed solution. The border nodes in the two graphs are $A5$, $A6$, $B5$, $B6$, $C5$ and $C4$. In the left-side graph, the release time of train C is the time operation $C5$ starts in the other graph, while the minimum exit time of train A and train B is computed as their free running till the end of block section 5. In the right-side graph, the release time of train A (B) is the starting time of operation $A5$ ($B5$) in the left-side graph, while the minimum exit time of train C is computed as its free running till the end of block section 5. In both the local areas, the trains are scheduled at block section 5 in such a way that C precedes B and B precedes A. The local solutions are locally optimal schedules and also result in a globally feasible solution. However, the global solution is sub-optimal. Train B has a conflict with train C on block section 4, with an additional waiting time of 5 time units, while train A has

several conflicts with train B on block sections 4, 5, 6 and 7. The most delayed train is train A that has a maximum consecutive delay of 50 time units. This simple example shows that there is an interesting gap between the optimal solution of the global area and the solution obtained by coordinating the optimal solutions of the local areas.

Another disturbed situation is next studied for the timetable of the illustrative example. Trains A and B are delayed at their entrance in the network by 20 and 120 time units, respectively. Figure 4.5 (top) shows the space-time diagram of the optimal local solutions for the two areas. Specifically, the maximum delay in the left-side graph is equal to 30 time units since train C has an additional waiting time on block section 5, caused by train B. Similarly, the maximum delay in the left-side graph is equal to 30 time units.

In the new disturbed situation, the optimal solutions of the two areas do not match and the coordinator will detect a global infeasibility as follows. The local ROMA controlling area x decides to give precedence to train B over train C for block section 5. Hence, there is a positive length path in area x from node $B5$ to node $C5$ (see Figure 4.5 (bottom-left)). Conversely, the local ROMA controlling area y decides to give precedence to train C over B on block section 5. In this case, there is a positive length path in area y from $C5$ to $B5$. The global situation is not known by the two ROMAs, which consider perfectly feasible their respective plans. On the other hand, at higher level the coordinator detects a positive length cycle involving nodes $B5$ and $C5$, that corresponds to the deadlock situation depicted in Figure 4.5 (top). The coordinator must ask one of the two solvers to change the order between trains B and C in order to get a globally feasible solution (i.e., in order to avoid the deadlock situation).

In the remaining part of this subsection, we describe how global infeasibilities between areas can be easily detected by an aggregate formulation. Based on the proposed decomposition of the network and the modeling of border situations, we define the *border graph* $G_B = (V_B, A_B)$, which consists of the set V_B of all border nodes and the set A_B of all border arcs. Note that nodes 0 and n are also nodes of the border graph, since these nodes represent a common reference for all areas. Let b_i, b_j be two border nodes. The set of border arcs is defined as follows. If a directed path from b_i to b_j exists in one of the solutions for the local areas, the arc (b_i, b_j) belongs to A_B. The weight of (b_i, b_j) is the length of a corresponding longest directed path in this local solution. The border graph (its arcs and the arc weights) is therefore a function of the current local solutions. The following theorem shows that the local solutions produced by the local ROMAs are globally feasible if and only if there are no positive length cycles in the resulting border graph.

Theorem 4.5.1: Consider a global area composed by k local areas. Given a locally feasible solution for each local area, the union of the k solutions is globally feasible if and only if the border graph has no positive length cycles.

Proof. Consider the global alternative graph associated to the whole region. This is

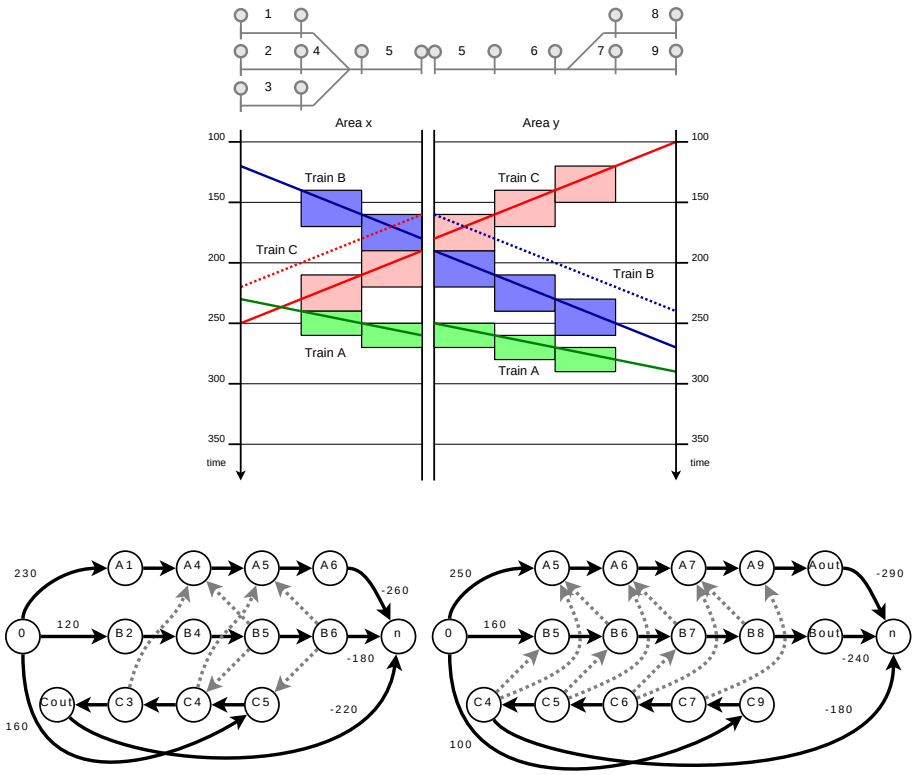


Figure 4.5: Optimal local solutions but infeasible global solution

obtained by sewing together the k local alternative graphs. By definition, a solution of the global alternative graph is feasible if and only if it contains no positive length cycles. A positive length cycle in the global graph can either belong entirely to a local area, or can involve nodes belonging to several areas. The former case cannot occur since all local solutions are feasible. The latter case occurs if and only if there is a positive length cycle involving at least two border nodes. This means that in each local area there is a directed path connecting a pair of border nodes, the total length of these paths being positive. Thus, there is a cycle in the global graph if and only if the border graph contains a positive length cycle, which concludes the proof. □

Each local ROMA transmits to the coordinator the list of all border nodes, together with the information concerning the route chosen for each train. The coordinator then matches the border nodes of different areas in order to build an aggregate representation of the interactions among the areas. Additional information computed by each local ROMA is the length of the longest path between every pair of border nodes. The coordinator exchanges with the local ROMAs only information related to the border nodes. This information is communicated by each ROMA to the coordinator that has to detect situations of global infeasibility. From Theorem 4.5.1, the coordinator simply has to build the border graph and check for the existence of positive length cycles. This can be computed in a fast way by means of existing graph search algorithms. For example, the algorithm of Floyd and Warshall requires a computing time $O(n^3)$, where n is the number of nodes of the border graph.

The left-side (right-side) of Figure 4.6 shows the border graph for the example of Figure 4.4 (Figure 4.5). In the border graph on the left-side of Figure 4.6 there is no positive length cycle. For reasons of clarity, we did not depict the (positive) weight of each border arc in the graph. On the other hand, in the border graph on the right-side of Figure 4.6 there are several positive length cycles involving two or more of the following border nodes: $B5$, $B6$, $C4$ and $C5$. The border arcs causing at least a cycle are emphasized in black in Figure 4.6.

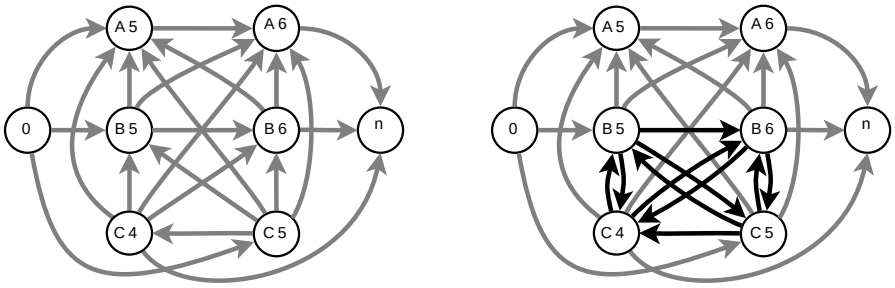


Figure 4.6: Border graphs of the feasible (left-side) and infeasible (right-side) global solutions

When a global infeasibility is detected, the coordinator may either impose precedence constraints among some trains at border nodes or impose the same value for the exit time from an area and the entrance time in the subsequent area for some train (see Section 4.5.3). Each local ROMA will then look for a new schedule in which these constraints are satisfied. After possible iterations for defining a feasible schedule, the resulting solution will be globally feasible if and only if a locally feasible solution can be found by all local ROMAs and the border graph has no positive length cycles.

4.5.3 Schedule Coordination Procedure

Figure 4.7 shows a new schedule coordination procedure that is based on an iterative exchange of information between two local solvers and the coordinator. At each step, the train scheduling problem in each local area is solved by the corresponding local solver. The coordinator checks whether the local solutions are consistent and result in a globally feasible solution, if not it communicates some constraints at the border between areas to the local solvers. The two steps are repeated until a globally feasible solution is found or a termination criteria is met. For the two areas, the local solutions are *consistent* when the local solvers compute the same times (and orders) for all operations associated to the border nodes between the two areas.

We introduce the border variables $B(i) = (T(i), O(i))$ for each scheduling iteration i so that a consistency check can be done between the local solutions. Variables $B(i)$ are used for bidirectional communication between the local ROMAs and the coordinator. A local ROMA sends a complete solution to the coordinator (times and orders). In response, the coordinator sends a partial solution (release times and partial orders) to the local ROMAs, regarding constraints at the borders that must be satisfied by the local solver at iteration $(i + 1)$. In the communication from the coordinator to the local solvers at iteration i, the border variables $T(i)$ contain the minimum entrance times at which the trains can enter the block sections at the borders. $O(i)$ is the set of partial orders for the trains at the borders, and clearly depends on $T(i)$.

For each local area, we define the border variables $B_x(i) = (T_x(i), O_x(i))$, representing times and orders of all trains running in the area x at iteration i. For two areas x_1 and x_2, the local solutions of the different areas are consistent if the values of border variables $B_{x_1}(i)$ and $B_{x_2}(i)$ are the same for all the border nodes contained in both areas.

The initial values of the border variables, $B(1)$, are computed as follows. Each local ROMA first computes the minimum starting time of each operation in its local alternative graph by running trains at maximum speed without considering the interactions among them, and communicates the exit time of every train to the coordinator. The coordinator then propagates these exit times to the neighboring area, as the minimum entrance times

Input: set $\{x_1, x_2\}$ of local areas, border variables $B(i)$
$i = 1$
while time limit of computation is not reached **do**
  for each local area x, concurrently
    impose $B(i)$ to the scheduling problem of the local area x
    solve the scheduling problem of the local area x
    if no solution to the local scheduling problem is found
      return local infeasibility to line dispatcher of area x
    else compute $B_x(i+1)$ from the local solution
  end for
  build border graph from the two local solutions
  if border graph has positive length cycles
    global_feasibility = false
  else global_feasibility = true
  if $B_{x_1}(i+1)$ and $B_{x_2}(i+1)$ are consistent and globally feasible
    return consistent and globally feasible solutions
  use update strategy to compute $B(i+1)$ from $B_{x_1}(i+1)$ and $B_{x_2}(i+1)$
  $i = i + 1$
end
return global infeasibility to network dispatcher

Figure 4.7: General sketch of the schedule coordination procedure

$T(1)$. The set $O(1)$ is left empty. Consequently, the local areas adjust the values of the border variables. This procedure converges after a finite number of iterative adjustments since the values of the border variables are updated progressively.

Given a set X of local areas and the border variables $B(1)$, the coordination procedure gets a solution from each local area, computed by one of the scheduling algorithms described in Section 4.3. The scheduling problem of each local area is decoupled from the one of other areas, so that the local ROMAs can compute their solutions by concurrent and parallel execution. If a local ROMA is not able to find a feasible schedule, a local infeasibility is reported to the line dispatcher that will have to implement other dispatching actions in its local area such as the modification of some train routes.

After computing a feasible schedule for both local areas, each local ROMA reports the new values for the border variables and the longest paths between each pair of border nodes. The coordinator collects this information, builds the border graph and analyzes the local solutions at network level. If the local solutions are globally feasible and consistent, the coordination procedure ends by returning the local solutions. If the local solutions are not globally feasible and consistent, the coordinator calls an update strategy to compute additional coordination constraints, and the overall procedure is iterated with new values for the border variables. If a given time limit of computation is reached before a globally feasible solution is found, the procedure reports a global infeasibility. In this case, the network dispatcher is asked to take scheduling decisions different from the ones computed by the local solvers, such as specify a new processing order between some operations at the area borders.

The simple update strategy implemented in this paper to compute $B(i+1)$ is as follows. An inconsistency is detected when there is at least one train crossing a border between the two areas with an exit time from an area different from the entrance time in the subsequent area. In this situation, the coordinator imposes a new minimum entrance time for the train in the subsequent area. If a train exits area x_1 and enters area x_2 and its entrance time in area x_2 is smaller than its exit time from area x_1, the coordinator propagates the exit time from the previous area to the next one. In other words, the train is forced to enter area x_2 not before the exit time from area x_1.

When the inconsistency involves train ordering, the coordinator also imposes the same order among the trains at the border in both areas, as follow. If the infeasibility involves two trains running in the same direction, the train order of the previous area is imposed to be the same in the following area. On the other hand, if the infeasibility involves two trains running in opposite directions, the train order is obtained by giving precedence to the first train reaching the border in the schedules of both local areas. In our computational experience, this simple procedure quickly converges to globally feasible solutions since only a few local decisions propagate to the other area.

4.6 Test Case

This section presents the experiments performed to evaluate the centralized and distributed systems over a large sample of real-life instances. The aim of the study is to assess how much train delays could be minimized by centralized and distributed approaches. We compare the solutions obtained by simple and advanced scheduling algorithms, further compared to the dispatching rule used in practice in the Netherlands. The procedures of the two systems are implemented on a PC equipped with two processors at 1.6 Ghz, 2 GB Ram and Windows XP operating system. For both systems the time limit of execution is 60 seconds. In the distributed system, each run of the scheduling algorithm is truncated after 10 seconds of computation.

4.6.1 Description of the Instances

The topology of the railway network around the main station of Utrecht is similar to a star with 5 main directions crossing each other (see Figure 4.8). The main lines relaying the North and South regions of the Netherlands are connected to the lines to the West and the East. The network is delimited by the following stations: Utrecht Overvecht on the line to Amersfoort, Driebergen-Zeist on the line to Arnhem, Culemborg on the line towards Den Bosch, Vleuten on the line to Rotterdam and The Hague plus Maarssen on the line towards Amsterdam. In total, the diameter of the area is around 20 km long.

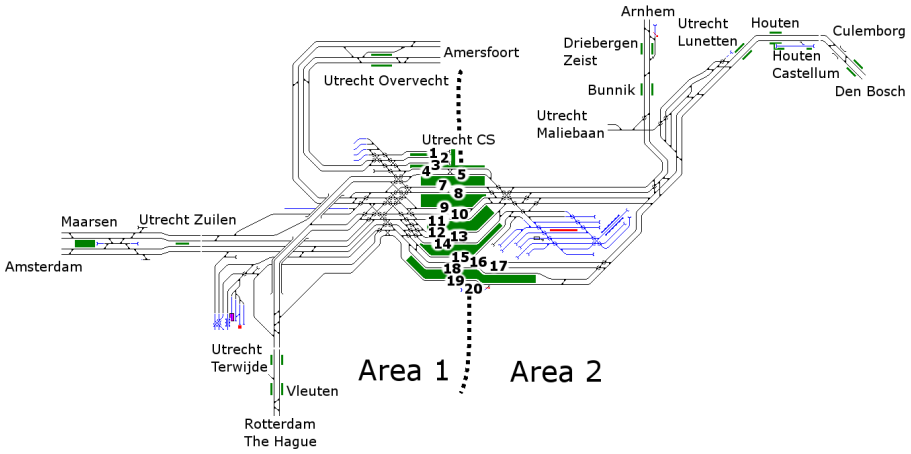


Figure 4.8: Railway network around Utrecht Central station

This is one of the most complex railway areas in the Netherlands. In total, the network includes more than 600 block sections. The interlocking area of Utrecht Central station

has, alone, around 200 block sections and more than 100 switches, leading to a large amount of inbound and outbound routes. Utrecht Central station provides 20 platform tracks, most of which might be reserved to two trains at a time. Most of the platform tracks are used by through traffic, i.e., trains running in the opposite direction, even if some trains change direction after their stop at a platform. There are also three dead-end platforms. We use one peak hour of the 2008 timetable that schedules up to 80 trains. The trains are mostly for passenger services, operated by NS (Nederlandse Spoorwegen), except for a few freight trains.

In the centralized system the network is managed as a whole, while in the distributed system the network is divided into two adjacent areas, as shown in Figure 4.8. At the borders between the two areas there are 20 platform tracks of Utrecht Central station. The operations at the area borders represent the 23 trains traversing the two areas. The other 57 trains only run on a single area. The same division of the dispatching task is also adopted in actual operations.

Practical statistical entrance delay distributions are studied that cause initial delays in the network. For the given timetable, we consider more than 33000 train events (arrivals, departures, dwell processes and passing times) that have been recorded at Utrecht Central station in April 2008 by ProRail. A statistical fitting procedure [38] is adopted to set the parameters of the Weibull distributions, which are used to characterize the delays of different trains and the variation in the dwell time process.

From the calculated statistical distributions, we investigate 90 random timetable perturbation instances with delayed entrance times and dwell time extensions. Each instance considers one hour of time horizon of traffic prediction, with 80 trains in the network. The average entrance delay is around 90 seconds per train while the maximum entrance delay is 390 seconds. These instances represent an average level of perturbed operations over one month of observation. We combine each delay instance with a set of six infrastructure scenarios that result in increasingly reduced availability of infrastructure at Utrecht Central station.

Table 4.3: Description of the disruption scenarios considered

Infrastructure Scenario	Blocked Platforms	% Unavailable Platforms	% Rerouted Trains
i	-	0	0
ii	7 8	10	10
iii	7 8 12 13 14	25	21
iv	2 3 7 8 12 13 14	35	35
v	2 3 7 8 12 13 14 18	40	45
vi	2 3 7 8 10 12 13 14 16 17 18	55	50

Table 4.3 describes the six infrastructure scenarios (see Column 1). Column 2 reports the specific platforms that are blocked and completely unavailable for traffic, Column 3 the percentage of unavailable platform tracks at Utrecht Central station, and Column 4 the percentage of trains on the hourly timetable that need to be rerouted to another station platform. The train rerouting is limited to the trains that cannot perform their scheduled trip in the station interlocking area with their original route since the assigned platform is blocked. It is worth being noted that scenarios (v) and (vi) require trains running on a single track in different traffic directions, thus resulting in an additional difficulty for the dispatching process due to possible deadlocks.

In total, we consider a set of 540 disturbance instances (i.e., all the combinations of the 90 timetable perturbations and of the 6 infrastructure scenarios). In the next subsection, we will compare the centralized and distributed approaches over all the 540 disturbance instances.

4.6.2 Computational Results

This subsection compares the global solutions obtained by the centralized and distributed systems. We consider the three scheduling procedures described in Section 4. For both the centralized and distributed systems, Table 4.4 reports the average behavior for the ARI procedure, Table 4.5 for the FCFS procedure and Table 4.6 for the BB algorithm. Each row of the tables reports the average results aggregated over the 90 timetable perturbation instances and for a given infrastructure scenario (see Column 1). Columns 2-4 describe the performance of the centralized system in terms of computation time (in seconds), maximum and average consecutive delays (in seconds). Similarly, Columns 5-7 give the same performance indicators for the distributed system. In both systems, the train delays are computed at all stations and at the exit from the overall network. The last three columns report further information on the distributed system. Column 8 presents the average number of border constraints imposed by the coordinator in order to recover the global feasibility, Column 9 the average number of scheduling iterations in the schedule coordination procedure, Column 10 the percentage of iterations that result in a global infeasibility. For each scheduling iteration, a new border graph is computed in order to check the global feasibility of the current local solutions. On average, the border graph has around 600 border nodes and 1100 border arcs. For all tested instances, there is no infeasibility and inconsistency in each final solution returned by the centralized and distributed systems. This interesting result is probably due to the choice of dividing the proposed two-area railway network at the platform tracks of Utrecht Central station.

Table 4.4 reports the average results for the implemented ARI dispatching rule. The centralized and distributed systems present global solutions that do not differ much in terms of maximum and average consecutive delays, with the distributed system performing slightly

Table 4.4: Average results on the two approaches by scheduling trains via the ARI procedure

Tests	Centralized System			Distributed System					
Infra Case	Comp Time	Max Delay	Avg Delay	Comp Time	Max Delay	Avg Delay	Coord Constr	Sched Iter	Glob Inf
i	10.5	271	10.9	4.8	276	11.9	14	1.1	0%
ii	11.9	437	22.7	7.2	437	23.3	15	1.2	4%
iii	13.7	584	32.6	7.7	561	31.5	27	2.0	17%
iv	13.9	584	47.0	8.5	561	46.8	27	2.0	17%
v	13.2	1515	66.2	8.6	1520	63.3	27	2.0	16%
vi	15.4	1515	65.2	7.7	1520	64.6	28	1.8	22%

better. This is due to the fact that ARI considers a combination of local scheduling rules and train priorities, which is only relatively sensitive to the problem decomposition and to the actual traffic situation. Both systems are able to deliver a feasible solution within 15 seconds, and the distributed system is always faster than the centralized one. Despite the solution process at the local level is heuristic, the coordination does not result in a major workload, being always able to resolve all the infeasibilities within up to 2 iterations of coordination.

Table 4.5: Average results on the two approaches by scheduling trains via the FCFS procedure

Tests	Centralized System			Distributed System					
Infra Case	Comp Time	Max Delay	Avg Delay	Comp Time	Max Delay	Avg Delay	Coord Constr	Sched Iter	Glob Inf
i	0.1	134	2.8	0.2	141	2.9	13	1.0	0%
ii	0.1	390	14.4	0.3	390	14.2	14	1.1	5%
iii	0.1	495	23.4	0.3	493	23.0	17	1.2	8%
iv	0.1	503	33.8	0.3	503	34.0	17	1.2	8%
v	0.1	1515	56.8	0.3	1520	55.3	17	1.2	8%
vi	0.1	1515	64.1	0.5	1520	62.1	28	1.8	22%

Table 4.5 gives the average results of the FCFS procedure. Also in this case, the two systems present similar global solutions in terms of consecutive delays. This is due to the fact that FCFS does also not take decisions by evaluating the propagation of train delays. Both systems are again very fast to solve the scheduling instances, even if for this heuristic the distributed system is slightly slower than the centralized one. This is in part due to the overload introduced by the coordination layer and in part to technical reasons related to the computation of the local schedules, that in practice is not completely performed in parallel due to the iterative exchange of information.

Table 4.6: Average results on the two approaches by scheduling trains via the BB algorithm

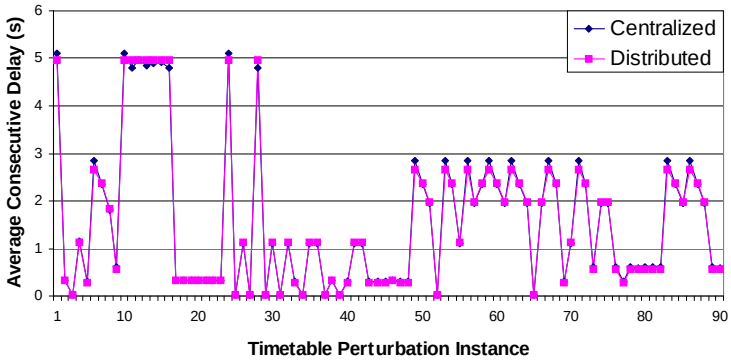
Tests	Centralized System			Distributed System					
Infra Case	Comp Time	Max Delay	Avg Delay	Comp Time	Max Delay	Avg Delay	Coord Constr	Sched Iter	Glob Inf
i	0.6	77	1.6	0.5	77	1.6	13	1.0	0%
ii	2.3	221	7.5	0.6	221	7.3	16	1.2	9%
iii	6.6	235	13.4	1.9	235	13.7	18	1.3	12%
iv	6.8	248	17.3	1.8	248	17.7	17	1.2	14%
v	60.0	1515	56.8	22.6	1520	42.0	17	1.2	11%
vi	60.0	1515	64.0	25.8	1520	43.2	24	1.5	29%

Table 4.6 presents the average results of the BB algorithm. The centralized system is more time consuming than the distributed one, the difference being more evident in case of strong disruptions. This is due to the direct impact of the instance size and complexity on the computation time of BB (the global alternative graphs have more than 5000 alternative pairs while each of the two local alternative graphs has less than 3000 alternative pairs). However, the global solutions reported for the two systems are very similar in terms of maximum consecutive delays. The distributed system reports smaller average consecutive delays, specially for the infrastructure scenarios (v) and (vi). This is probably due to the iterative use of BB in the distributed solving procedure.

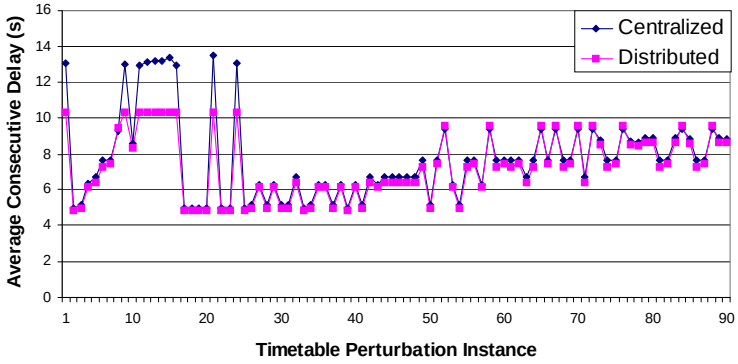
Comparing the solutions computed by the BB algorithm and by the other scheduling procedures, BB outperforms both ARI and FCFS, for both the centralized and distributed systems. This is due to the better performance of BB in minimizing the consecutive delays compared to the local rules.

The last three columns of Tables 4.4, 4.5 and 4.6 give other relevant information about the distributed framework. The number of coordination constraints and of scheduling iterations are two indicators on the information exchanged between the two local ROMAs and the coordinator. Comparing the results for the different infrastructure scenarios, when the level of disturbances increases we conclude that more coordination constraints have to be imposed and more scheduling iterations are required to obtain globally feasible solutions.

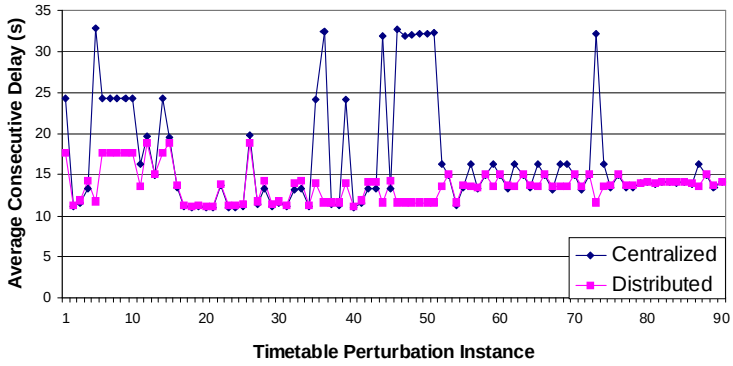
During the iterative scheduling procedure, the percentage of global infeasibilities at intermediate iterations (see the last column of the tables) is also increasing when dealing with more disrupted scenarios. However, the final iteration of the schedule coordination procedure is always able to return a globally feasible solution. It is interesting to note that this indicator does not depend on the scheduling procedure used by the local ROMAs.



Infrastructure Scenario (i)

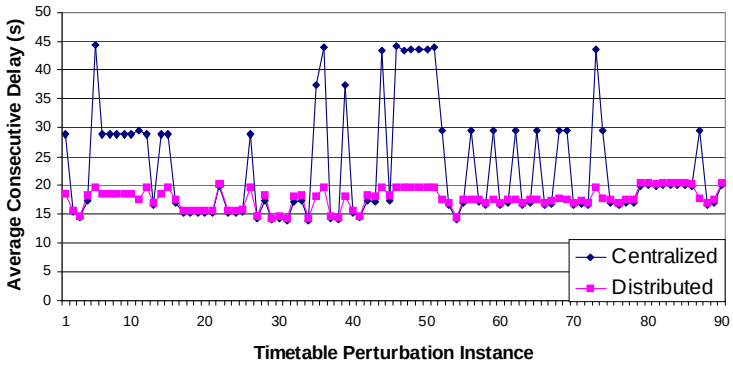


Infrastructure Scenario (ii)

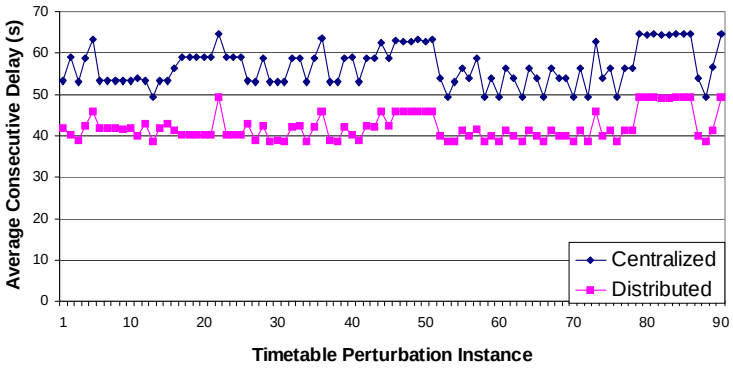


Infrastructure Scenario (iii)

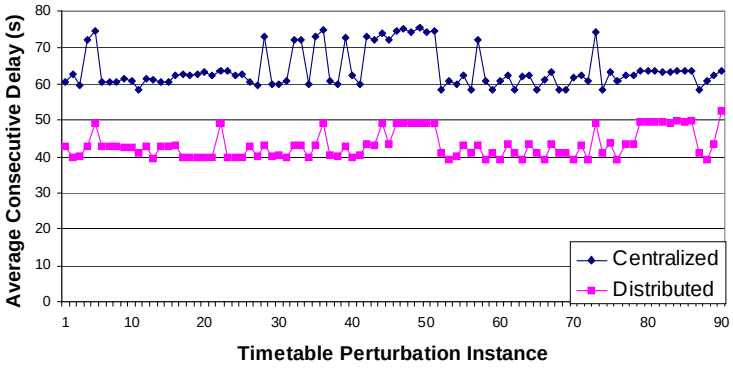
Figure 4.9: Comparative diagrams of average consecutive (output) delays against timetable perturbation instances sorted by average entrance delay, scenarios i-iii, BB algorithm



Infrastructure Scenario (iv)



Infrastructure Scenario (v)



Infrastructure Scenario (vi)

Figure 4.10: Comparative diagrams of average consecutive (output) delays against timetable perturbation instances sorted by average entrance delay, scenarios iv-vi, BB algorithm

Figures 4.9 and 4.10 give a detailed view on the performance of the centralized and distributed systems when using the BB algorithm. Each plot is done for the 90 timetable perturbation instances of Section 4.6.1 and one infrastructure scenario. The x-axis shows the instances ordered by increasing average entrance delay, while the y-axis reports the average consecutive delay for each instance. The plots give an idea of the low correlation between entrance and consecutive delays. We found similar features when comparing other input/output factors such as the number of delayed trains or other punctuality measures. Differently, the average consecutive delay increases appreciably with the variation of the infrastructure scenario, since when reducing the available infrastructure more traffic is scheduled on the same platforms and there are more potential conflicts causing a larger delay propagation.

In the scenarios from (iii) to (vi), the distributed system outperforms significantly the centralized system. The distributed system is more robust to changes in the entrance delays in the scenarios (iii) and (iv), and it is systematically better in the scenarios (v) and (vi). The main reasons for this behavior are seemingly the following. On the one hand, it is difficult for the centralized system to compute optimal solutions in a short computation time when trains interact heavily and there is a limited capacity available in the station interlocking areas. On the other hand, the network decomposition into subproblems keeps the complexity within an acceptable level for the local ROMAs, and the coordination procedure is still able to guide the local solvers to find globally feasible solutions.

4.7 Conclusions and Future Research

This paper presents models, procedures and experiments on centralized and distributed rescheduling systems for the management of a complex and densely occupied railway area of the Dutch network. Heuristic and exact train scheduling algorithms plus novel coordination procedures have been investigated on several timetable perturbation instances and infrastructure scenarios. The distributed system with the BB algorithm for the local ROMAs is faster than the centralized ROMA solved by BB, and shows significantly better performance in terms of the consecutive delays, specially for serious traffic disturbances. Differently, when using the FCFS procedure the centralized system is faster than the distributed one, but the average consecutive delay may increase up to 75% compared to the results of BB, and an even larger gap is obtained when comparing BB with the ARI procedure.

Future research should be dedicated to develop more sophisticated techniques for traffic control of large and busy networks. Following this research line, a number of issues should be addressed. Since Theorem 4.5.1 holds for any number of areas, effective coor-

dination techniques could be developed for managing larger networks with more than two dispatching areas. To this aim, the coordinator should have more decisional capacity in order to solve possible cycles in the border graph of multiple areas (i.e., deadlocks involving trains running in different areas). The impact of coordination actions on the quality of the global solutions is also a problem that requires further research. The impact of various subdivisions of the network into local areas should be investigated more extensively. Finally, other features, such as rerouting and speed regulation, should be integrated in the decision support systems in order to face practical needs of railway operators.

4.8 Acknowledgements

We thank the Dutch infrastructure manager ProRail (specially D. Middelkoop and L. Lodder) for providing the instances. This work is partially supported by the Dutch programs "Towards Reliable Mobility" of the Transport Research Centre Delft and by the Italian Ministry of Research, Grant number RBIP06BZW8, project FIRB "Advanced tracking system in intermodal freight transportation".

Bibliography

[1] B. Adenso-Díaz, M. Oliva González, and P. González-Torre. On-line timetable rescheduling in regional train services. *Transportation Research, Part B*, 33:378–398, 1999.

[2] N. Berends and N. Ouburg. Beschrijving ARI-functionaliteit. Technical Report 20, ProRail Internal Specification (in Dutch), Utrecht, The Netherlands, 2005.

[3] G. Caimi, D. Burkolter, T. Herrmann, F. Chudak, and M. Laumanns. Design of a railway scheduling model for dense services. *Networks and Spatial Economics*, 9 (1)(1):25–46, 2009.

[4] Y. H. Chou, P. F. Weston, and C. Roberts. Dynamic Distributed Control for Real-time Rescheduling of Railway Networks. In I. A. Hansen, A. Radtke, J. Pachl, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

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

[6] Pacciarelli D. Deliverable D3: Traffic Regulation and Co-operation Methodologies code wp4ur_dv_7001_d. In *project COMBINE 2 "enhanced COntrol centres for fixed and Moving Block sIgNalling systEms" Number: IST-2001-34705*. 2003.

[7] A. D'Ariano. *Improving Real-Time Train Dispatching: Models, Algorithms and Applications*. PhD Thesis, TRAIL Thesis Series T2008/6, The Netherlands, 2008.

[8] A. D'Ariano, F. Corman, D. Pacciarelli, and M. Pranzo. Reordering and local rerouting strategies to manage train traffic in real-time. *Transportation Science*, 42 (4):405–419, 2008.

[9] A. D'Ariano, D. Pacciarelli, and M. Pranzo. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183(2):643–657, 2007.

[10] A. D'Ariano and M. Pranzo. An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances. *Networks and Spatial Economics*, 9 (1):63–84, 2009.

[11] A. D'Ariano, M. Pranzo, and I. A. Hansen. Conflict resolution and train speed co-ordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 8(2):208–222, 2007.

[12] E. Goddard. Overview of signalling and train control systems. In *The 9th Institution of Engineering and Technology Professional Development Course on Electric Traction Systems*, pages 336–350. Manchester, UK, 2006.

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

[14] C. Hirai, N. Tomii, Y. Tashiro, S. Kondou, and A. Fujimori. An algorithm for train rescheduling using rescheduling pattern description language R. In J. Allan, C. A. Brebbia, A. F. Rumsey, G. Sciutto, S. Sone, and C. J. Goodman, editors, *Computers in Railways X*, pages 551–561. WIT Press, Southampton, UK, 2006.

[15] R. V. Iyer and S Gosh. DARYN – A distributed decision-making algorithm for railway networks: Modeling and simulation. *IEEE Transactions on Vehicular Technology*, 44 (1):180–191, 1995.

[16] J. Jacobs. Reducing delays by means of computer-aided 'on-the-spot' rescheduling. In J. Allan, C. A. Brebbia, R. J. Hill, G. Sciutto, and S. Sone, editors, *Computers in Railways IX*, pages 603–612. WIT Press, Southampton, UK, 2004.

[17] L.-M. Jia and X.-D. Zhang. Distributed intelligent railway traffic control: A fuzzy-decision making-based approach. *Engineering Applications of Artificial Intelligence*, 7(3):311–319, 1994.

[18] E. Lamma, P. Mello, and M. Milano. A distributed constraint-based scheduler. *Artificial Intelligence in Engineering*, 11:91–105, 1997.

[19] T. S. Lee and S Gosh. Stability of RYNSORD – A decentralized algorithm for railway networks under perturbations. *IEEE Transactions on Vehicular Technology*, 50 (1):287–301, 2001.

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

[21] M. Mazzarello and E. Ottaviani. A traffic management system for real-time traffic optimisation in railways. *Transportation Research, Part B*, 41 (2):246–274, 2007.

[22] Message Passing Interface Forum. MPI: A message passing interface standard. *The International Journal of Supercomputer Applications and High Performance Computing*, 8, 1994.

[23] M. Missikoff. 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*, pages 633–641. 1997.

[24] L. Nie and I. A. Hansen. System analysis of train operations and track occupancy at railway stations. *European Journal of Transport and Infrastructure Research*, 5 (1):31–54, 2005.

[25] J. Pachl. *Railway Operation and Control*. VTD Rail Publishing, Mountlake Terrace, 2002.

[26] G. Parodi, G. Vernazza, and F. Zunino. Stability and deadlock avoidance in distributed system for traffic control. *IEEE Transactions on Vehicular Technology*, 45 (4):732–743, 1996.

[27] L. Ping, N. Axin, J. Limin, and W. Fuzhang. Study on intelligent train dispatching. In *Proceedings of the 4th IEEE International Conference on Intelligent Transportation Systems*, pages 949–953. 2001.

[28] J. Rodriguez. A constraint programming model for real-time train scheduling at junctions. *Transportation Research, Part B*, 41 (2):231–245, 2007.

[29] İ. Şahin. Railway traffic control and train scheduling based on inter-train conflict management. *Transportation Research, Part B*, 33:511–534, 1999.

[30] M. A. Salido, M. Abril, F. Barber, L. Ingolotti, P. Tormos, and A. Lova. Domain-dependent distributed models for railway scheduling. *Knowledge-Based Systems*, 20:186–194, 2007.

[31] A. Schrijver and A. Steenbeek. Dienstregelingontwikkeling voor Railned: Rapport CADANS 1.0. Technical report, Centrum voor Wiskunde en Informatica, Amsterdam, the Netherlands, 1994. In Dutch.

[32] C. Strotmann. *Railway Scheduling Problems and their decomposition*. PhD thesis, Universität Osnabrück, 2007.

[33] R. Takagi, P. F. Weston, C. J. Goodman, C. Bouch, J. Armstrong, J. Preston, and S. Sone. 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, and C. J. Goodman, editors, *Computers in Railways X*, pages 479–488. WIT Press, Southampton, UK, 2006.

[34] J. Törnquist. *Railway traffic disturbance management*. PhD thesis, Blekinge Institute of Technology, 2006.

[35] S. Tsuruta, T. Eguchi, S. Yanai, and T. Ooshima. A coordination technique in a highly automated train rescheduling system. In *IEEE International Conference on Systems, Man, and Cybernetics*. 1999.

[36] J. H. A. van den Berg and M. A. Odijk. DONS: Computer aided design of regular service time-tables. In T.K.S. Murthy, C.A. Brebbia, B. Mellitt, G. Sciutto, and S. Sone, editors, *Computers in Railways IV*, pages 109–16. WIT Press, 1994.

[37] S. Wegele, R. Slovák, and E. Schnieder. Real-time decision support for optimal dispatching of train operation. In I. A. Hansen, A. Radtke, J. Pachl, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

[38] J. Yuan. *Stochastic Modelling of Train Delays and Delay Propagation in Stations*. PhD Thesis, TRAIL Thesis Series T2006/6, The Netherlands, 2006.

[39] P. J. Zwaneveld, L. G. Kroon, and S. P. M. Van Hoesel. Routing trains through a railway station based on a node packing model. *European Journal of Operational Research*, 128:14–33, 2001.

Chapter 5

Dispatching and coordination in multi-area railway traffic management

Abstract *This paper deals with the development of decision support systems for traffic management of large and busy railway networks. Railway operators typically structure the control of complicated networks into the coordinated control of several local dispatching areas. A dispatcher takes rescheduling decisions on the trains running on its local area while a coordinator addresses global issues that may arise between areas. While several advanced train dispatching models and algorithms have been proposed to support the dispatchers' task, the coordination problem did not receive much attention in the literature on train scheduling. As far as decision support systems are concerned, the control of a railway network can be achieved either by a centralized approach in which the whole scheduling problem is solved by a single scheduler, or by a distributed approach in which a coordinator sets constraints between areas and delegates scheduling decisions to local schedulers. This paper compares centralized and distributed procedures to support the task of dispatchers and coordinators. We adopt dispatching procedures driven by optimization algorithms and based on local or global information and decisions. Computational experiments on a Dutch railway network, actually controlled by ten dispatchers, assess the performance of the centralized and distributed procedures. Different types of traffic disturbances, including entrance delays and blocked tracks, are analyzed on various time horizons of traffic prediction. Results show that the centralized procedure faces increasing difficulty in finding feasible solutions in a short computation time. For seriously disturbed instances with a permanent track blockage, the distributed approach is more suitable than the centralized one in order to manage traffic in the overall area.*

This chapter is an edited version of the article

F. Corman, A. D'Ariano, D. Pacciarelli, M. Pranzo. Dispatching and coordination in multi-area railway traffic management, submitted for publication.

5.1 Introduction

Railway traffic management of large and busy networks is typically based on detailed train timetables, which are usually defined off-line even in presence of blocked tracks. Careful scheduling is necessary since safety rules between trains impose that at most one train at a time can occupy a *block section*, i.e., a portion of railway delimited by signals. In accordance to such rules, the sequence of trains traversing each block section of the railway network has to be accurately defined so that feasible timetables can be developed, without any deadlock. A set of trains cause a *deadlock* when each train in the set claims a block section ahead which is not available, due either to a disruption or to the occupation/reservation for another train in the set.

During operations, a disturbance handling phase is needed in order to limit propagation of train delays and to avoid any deadlock in the network. In the railway practice this real-time task is often hierarchically organized into two decision levels. At the lower level, dispatchers control local areas with a visibility of the traffic flow limited to their respective areas. At the higher level, coordinators are responsible for the traffic management over a railway network of n areas with a global overview of the traffic flow and based on the rescheduling decisions taken by dispatchers. Typically, coordinators are mainly interested in controlling the trains traversing multiple areas and in taking decisions at the border between areas, while the traffic control in the local areas is left to the dispatchers. Figure 5.1(a) gives an example situation for a small single track line with 10 block sections and 6 trains, divided into 2 dispatching areas, named X and Y. In the left-right traffic direction, trains A and B pass through block sections 1, 2, 4, 5, 7 and 8; train D through block sections 5, 7 and 8. In the other direction, train C passes through block sections 4, 3 and 1; train E through block sections 6, 4, 3 and 1; F through block sections 9, 7, 6, 4 and 10. Figures 5.1(b) and 5.1(c) show the view of the local dispatchers on their respective area at a specific time instant of the traffic prediction.

Decision support tools are needed to help dispatchers and coordinators to detect and avoid deadlocks under the two-level hierarchy. On the one hand, tools have been recently developed by academic and professional railway researchers to support the dispatching process in a single dispatching area. On the other hand, the coordination problem did not receive much attention in the literature on multi-area train scheduling, although poor coordination between areas may result in poor overall performance. In this context, distributed optimization tools offer an effective way to deliver viable solutions in a short computation time. However, such tools evaluate the solution quality and feasibility only locally. The solutions obtained through distributed control may be of bad quality from a global perspective and even the global feasibility cannot be always guaranteed. Coordination is therefore needed in order to provide effective solutions for the overall network. Clearly, dividing the network in a large number of dispatching areas would increase the workload of the coordinator and simplify that of the local dispatchers. So, a compromise has to be

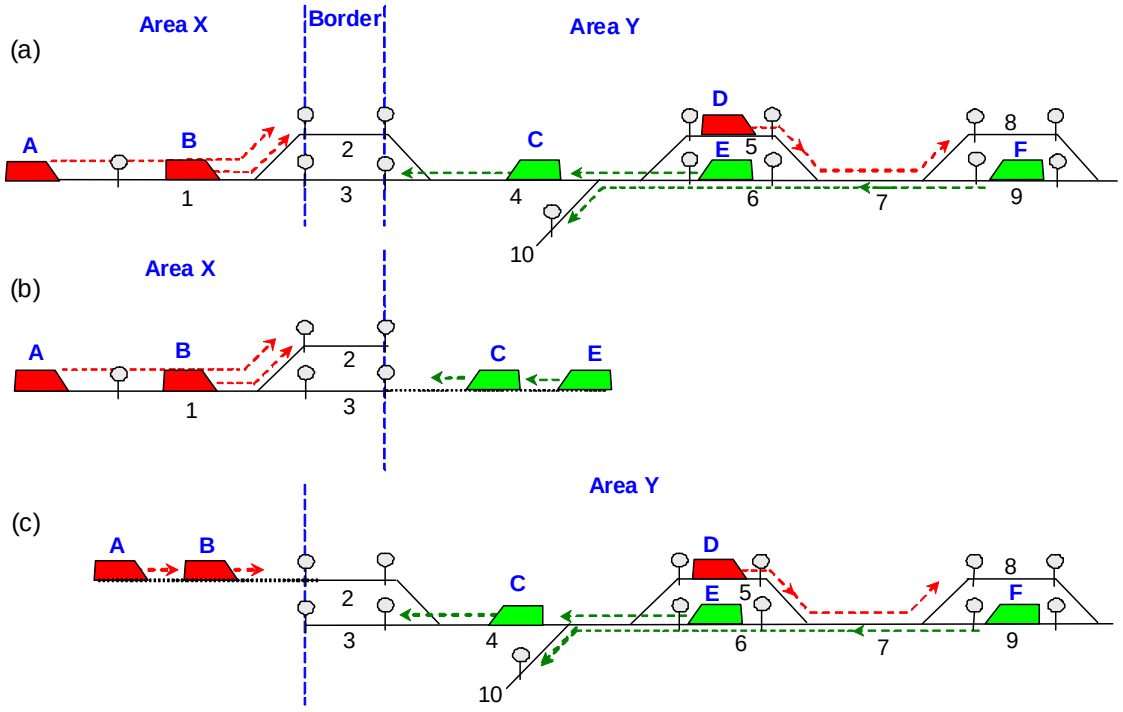


Figure 5.1: Example of a centralized view and local views for area X and area Y.

found between the size of the dispatching areas, the computation time left to the local dispatchers to produce local solutions, and the type of algorithms used to reschedule trains in each area. Centralized scheduling algorithms can be used in small dispatching areas while distributed approaches with coordination are more suitable in large areas in order to obtain feasible solutions in a short time.

This paper investigates the effects of decomposing a multi-area network in local areas, each controlled by a local dispatcher. In our model, a network coordinator sets constraints at the border between areas and delegates scheduling decisions to local schedulers, driving the search of the local dispatchers toward globally feasible schedules. In this context, the disturbance handling phase can be particularly difficult, specially when managing railway networks with dense traffic. In the example situation of Figure 5.1 there is a high risk of deadlock, which can only be avoided by giving precedence to train D on block section 7. This situation must be detected well in advance, before F enters block section 7, and solved by imposing the precedence between D and F. With the centralized view of Figure 5.1(a) is relatively simple to detect this risk of deadlock. In the two-level hierarchy deadlock avoidance can be a harder problem, since this requires to control two trains that do not cross the XY border (i.e., trains F and D), and with our model the coordinator does not take decisions on these trains. Furthermore, the local dispatcher of each area has a limited view of the network and may not recognize the risk of deadlock. In Figure 5.1(c),

if train F is heavily delayed there is a high chance that the dispatcher of area Y gives precedence to train F over D on block section 7, with the (wrong) idea that all the trains moving from right to left can have priority over the others without causing any deadlock. In the next sections, we will show that the coordinator forces the dispatcher of area Y to avoid the deadlock by setting suitable constraints among the trains crossing the border (i.e., trains A, B, C and E).

The main contribution of this paper is the assessment of advanced rescheduling approaches for managing the rail traffic in a large network with dense traffic. The underlying model used to reschedule trains in each local dispatching area is a microscopic formulation based on the alternative graph of [18] and on the blocking time theory [13]. The coordination model is based on aggregated information coming from each local area. The coordination procedure generalizes to n dispatching areas the one studied in [4] (included in this thesis as Chapter 4) for two dispatching areas of the Dutch railway network. The new coordination procedure is tested on a practical railway network for different network decompositions, including the case in which the control of the whole area is centralized under the control of a single scheduler. The tested approaches range from simple and fast dispatching rules to more sophisticated scheduling heuristics and a state-of-the-art exact algorithm [11]. The advanced train scheduling algorithms take decisions based on global information about the traffic flow and the impact of disturbances. Such algorithms provide better performance of the rail system, with respect to local dispatching rules, but suffer from a rapid increase of execution time and memory requirements as the system grows in size with more trains, track segments and stations. In fact, even the problem of deciding whether a feasible schedule exists or not is NP-hard [18] and thus the complexity of known exact solution algorithms is exponentially growing with the problem size.

An extensive campaign of experiments compares the centralized and distributed procedures on real-world instances. Specifically, four heuristics and two variants of the exact algorithm are evaluated for the dispatching problem. The best algorithms are then incorporated within the distributed framework and tested for different network divisions. The assessment is based on different types of traffic disturbance and increasing time horizons of traffic prediction. We consider delayed trains at their entrance in the network and a serious and permanent track blockage situation. The proposed algorithms are compared in terms of number of feasible solutions, computation time and delay minimization.

The paper is organized as follows. Section 2 discusses the literature on railway traffic management on networks spanning over several dispatching areas. Section 3 presents the centralized and distributed approaches, and describes the solution procedures for dispatching and coordination. Section 4 reports on the computational experiments on a Dutch railway network and various disturbed traffic situations. Section 5 concludes the paper and outlines directions for further research.

5.2 Review of the related literature

This section reviews mathematical models and algorithms to support the task of dispatchers and coordinators. We limit the analysis to the real-time control of railway traffic and do not review papers on timetable development, which address a significantly different problem. A major difference between timetable planning and rescheduling is that in the latter case models and algorithms must be compliant with the actual state of the network and with the current position of each train. Moreover, rescheduling algorithms must deliver good solutions within short computation time. We classify the problems addressed in the literature in distributed rescheduling, centralized rescheduling and coordinated rescheduling. The complexity of the railway network considered ranges from a simple junction to a set of dispatching areas.

Among distributed approaches based on negotiations at the level of junctions, Vernazza and Zunino [27] propose an approach to solve train conflicts locally by enabling a negotiation between the 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 controller. The system simulates a realistic network and train conflicts at simple railway junctions are solved by local decision rules depending on the traffic intensity. Parodi et al. [20] study an advanced resource-allocation task for train scheduling based on local decision rules, and study how to detect and solve in advance possible deadlocks for different network configurations.

In Iyer and Gosh [14] every train is equipped with an onboard 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 track segments and participates in the negotiation with the trains. Their experiments, carried on an artificial test case with up to 12 stations, 17 track segments and 48 trains, show that the computation time increases rapidly.

Lee and Gosh [17] also report on a decentralized train scheduling algorithm to control large networks within a short computation time. They study the system performance on a simulated railroad network with 50 stations, 84 track segments and several freight trains. Their results indicate that the proposed approach is stable compared 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.

Other distributed approaches use Petri Nets to model the railway traffic flow. Fay [12] describes an expert system and suggests a fuzzy rule-base, Fuzzy Petri Net, for train traffic control during disturbances. Experiments are performed on fictional data with some

trains and one station. Zhu [29] introduces a simulation model based on stochastic Petri nets in order to assess the impact of incidents on the quality of operations. The latter approach focuses on how to determine train traffic delays caused by primary stochastic disturbances, especially technical failures. Cheng and Yang [1] include further factors, such as train connections and passenger trip types, in a similar fuzzy Petri Net approach for managing the dispatching process. The dispatching local decision rules are collected via interviews to experts. However, Petri Net approaches still seem to be far from producing near-optimal solutions to practical problem instances.

Several distributed approaches to railway scheduling problems are introduced by Salido et al. [24]. Semi-independent subproblems are generated by graph partitioning and by grouping together subsets of trains or contiguous stations. The subproblems are solved by constraint programming techniques. Experiments based on small fictitious instances with increasing trains and stations show promising results for the distributed approaches.

Among the centralized approaches for managing a dispatching area, Şahin [23] formulates a meet and pass problem as a job shop scheduling problem. Conflicts between up to 20 trains are solved in the order they appear for 19 meet points. An algorithm based on look-ahead measures detects potential delays and takes ordering decisions at merging or crossing points in order to minimize the average delays.

Wegele et al. [28] use 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, adaptation of train speeds on corridors and local rerouting in-side stations. Examples of application on a large part of the German railway network are reported for a single delayed train.

Rodriguez [22] focuses on a real-time train conflict resolution problem and proposes a train routing and scheduling system based on constraint programming. The experiments show that a truncated branch and bound algorithm can find satisfactory solutions within a short time for a railway junction of a few kilometers traversed by up to 24 trains.

Törnquist and Persson [26] introduce a model for dispatching trains in a railway network with several merging and crossing points. A mixed integer linear programming problem is formulated and solved with commercial software packages. Heuristic scheduling strategies are proposed to reduce the search space by restrictions of reordering and local rerouting actions. Experiments are presented for various disturbance settings on a Swedish railway network with 253 track segments traversed by up to 80 trains for a 90-minute horizon of traffic prediction.

An exact method has been proposed by D'Ariano et al. [11] for a train scheduling problem with fixed routing. Their computational experiments, carried on the Dutch railway bottleneck around Schiphol International airport and for multiple delayed trains, show

that optimal or near-optimal solutions can be found within a short computation time. In two follow-up papers [10, 5], this algorithm is incorporated in more sophisticated meta-heuristic frameworks to manage multiple train rerouting options.

The problem of coordinating the tasks of different dispatchers is still a very under-researched topic. Among the few papers on this subject, Jia and Zhang [15] present a first approach based on fuzzy decision-making for distributed railway traffic control. A multi-level decisional process is described that consists of several regional decision centers to be coordinated. The test case is a main Chinese network with 12 stations and 12 trains.

Lamma et al. [16] propose a distributed advisory system which helps traffic controllers in traffic management and control within railway stations and along railway branches. The scheduling of trains along a railway line with up to 31 trains is performed by traffic control modules, each one controlling a limited number of block sections and solving train conflicts based on local decision rules. These modules need to frequently exchange information about the local traffic and to verify the solution feasibility.

Chou et al. [2] propose a distributed control system and study a number of railway areas that are mutually influenced. A novel time-shift coordination strategy between neighboring traffic control areas is proposed for collaborative train rescheduling. Distributed control techniques in neighboring regions are applied in a fictitious network and evaluated in terms of delay cost. Recently, Chou et al. [3] demonstrate a significantly lower delay cost compared to a first come first served rule for a realistic railway junction with around 12 trains per peak hour traveling in a single direction.

We next discuss recent approaches based on the alternative graph formulation introduced in [18]. Methodologies for railway traffic regulation and coordination of local areas were developed within the European project COMBINE 2 [8]. Mazzarello and Ottaviani [19] report on the implementation of these methodologies for two test cases of the Dutch railway network. They also report on a practical pilot carried out for one of the two test cases. Based on the COMBINE 2 architecture, Strotmann [25] presents a two-level approach for rescheduling trains between multiple areas. A first level solver computes traffic control measures in each area, while a second level solver is adopted to check whether neighboring areas have consistent solutions. A coordinator graph is used to find out infeasibilities between neighboring areas which will result in additional constraints to the first level solver. An iterative approach of imposing train ordering constraints is adopted until a feasible solution to the global problem is found or it is proven that no globally feasible solution exists. Fictitious examples are studied with up to 16 trains and 73 block sections. Corman et al. [4, 6] (included in this thesis as Chapter 4) develop ideas from the COMBINE 2 project to solve the coordination problem for a complex and busy railway station divided into two dispatching areas, and compare distributed and centralized systems.

As a general remark about the existing literature, we observe that most of the existing approaches lack of a thorough computational assessment and limit the analysis to simple networks or simple perturbation patterns. In fact, the analyzed delay patterns are often quite specific, e.g. only one train is delayed or the problem is limited to a single junction or to a straight line. Moreover, the models used in the literature for the assessment are often simplified and do not capture entirely the consequences of delays and other disturbances. In order to solve practical problems, detailed models are necessary that capture the real problem complexity. For instance, when dealing with large networks and dense traffic, possibly with disruption, the risk of deadlock is relevant and should not be ignored by real-time models. The alternative graph model is among the few models that incorporate the level of detail necessary to generate deadlock-free schedules within an optimization framework.

In practice, the real-time railway traffic management is based on the decomposition of large and busy networks in coordinated local areas. However, there is a lack of research on this topic, besides few seminal papers with limited empirical assessment. Published papers do not often analyze in depth the trade-off between solution quality, time horizon of traffic prediction and computational effort of the proposed algorithms.

In this paper we explore the limits for practical applicability of known versus new scheduling and coordination algorithms based on the alternative graph to support railway operators in the management of disturbed traffic situations on an increasing number of trains and levels of disturbance. To this aim, we analyze a Dutch railway network, spanning over ten dispatching areas, with various time horizons of traffic predictions and different network decompositions. The disturbances include multiple delayed trains and a serious and permanent disruption in the network, which requires the rerouting of several trains and the management of complex traffic situations with the risk of deadlocks.

5.3 Solution methods and procedures

This section presents the disturbance handling process addressed in this paper. Centralized and distributed traffic management architectures are first introduced and then described in terms of scheduling and coordination models and algorithms.

5.3.1 Disturbance handling process

Disturbance handling is the management of railway traffic during operations in order to react to severe disturbances that make the timetable infeasible. The real-time timetable adjustment consists of rescheduling the trains running in the overall railway network, in

terms of their routes, orders and times. The railway traffic is predicted over a given time horizon and the delay propagation is studied over multiple dispatching areas.

We consider a *microscopic timetable* which describes the movement of all trains running in the network during a given hour, specifying, for each train, 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 after its scheduled arrival time. A *timetable perturbation* is given by a set of train delays at the entrance of the network or at scheduled stops and an *infrastructure disruption* is the presence of blocked tracks, requiring modifications of scheduled speeds and/or routes.

The microscopic formulation of the disturbance handling process requires modeling of railway networks at the level of block sections and signals. A block section is a track segment between two main signals and may host at most one train at a time. 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 performed in a dispatching area during a *service*. The *release time* of a train in an area is the minimum time at which the train can enter its first block section of the area, i.e., this is the earliest starting time of the first operation in the area associated to the train route. Each operation requires a given *running time* which depends on the actual speed profile followed by the train while traversing the block section. The minimum time separations among the running trains translate into a minimum *setup time* between the exit of a train from a block section and the entrance of the subsequent train into the same block section.

In our terminology, a *conflict* occurs when two or more trains claim the same block section simultaneously. In this situation a decision on the train ordering has to be taken and one of the trains involved has to change its running time according to the speed constraints of the signaling system. We compute the resulting train delays as follows. The *total delay* is the difference between the calculated train arrival time and the scheduled time at a relevant point in the network, and is divided into two parts. The *initial delay* is caused by disturbances (e.g., failures, entrance delays, blocked tracks) and cannot be recovered by rescheduling train movements, except by running trains at their maximum speed. The *consecutive delay* is caused by the interaction between trains running in the network during a given time horizon of traffic prediction.

5.3.2 Traffic management architectures

Figure 5.2 shows the system architectures in terms of actors involved and information flow. We consider a network divided into n local areas with a dispatcher for each area.

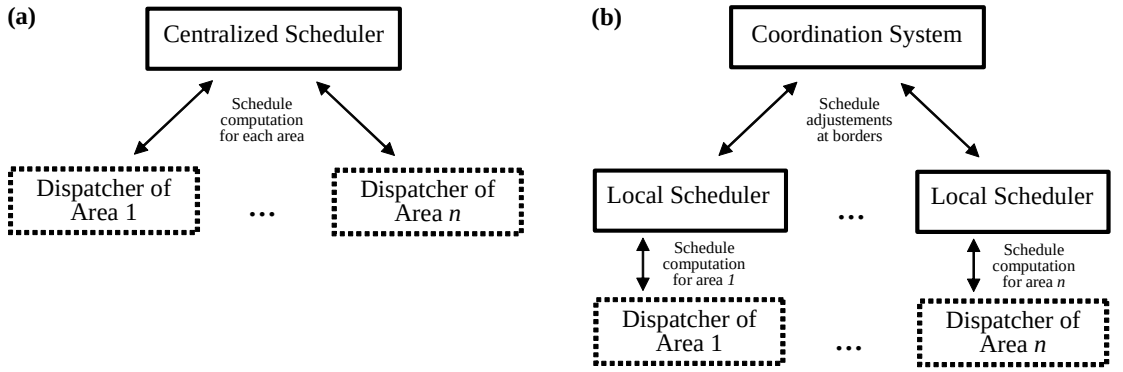


Figure 5.2: Centralized and distributed architectures.

With the centralized architecture of Figure 5.2(a), there is no need of coordination since the disturbance handling process is modeled and solved over the overall railway area as a single scheduling problem solved by a single centralized scheduler. When a globally feasible solution is found, each dispatcher receives the solution of its local area by the centralized scheduler. If the centralized scheduler does not find a feasible solution within a given time limit an infeasibility is reported and the human dispatchers are asked to take control actions in their respective areas.

With the distributed architecture of Figure 5.2(b), each area is controlled by a local scheduler while the overall area is supervised by a coordination system at a higher level. In this case there are two levels of infeasibility. If a local scheduler does not compute a feasible solution within the given time limit a *local infeasibility* is reported and the human dispatcher of that area is asked to take control actions. In this case there is not a globally feasible solution. If all local schedulers find a locally feasible solution in their areas but these solutions are not globally feasible, then a *global infeasibility* is found. In this case, the coordinator is asked to repair the situation by setting constraints at the borders between areas and the local schedulers are asked to find a new local solution compliant with coordination constraints. The procedure continues until a globally feasible solution is found or some local scheduler does not find a locally feasible solution compliant with the coordinator constraints. In the latter case, there is a global infeasibility.

5.3.3 Dispatching model and procedures

The disturbance handling process is formulated as a job shop scheduling problem with additional constraints and formulated as an alternative graph [18] using the blocking time theory [13] to compute time separations between operations.

The alternative graph is a triple $\mathcal{G} = (N, F, A)$, where N is the set of nodes, F is the set of

fixed arcs and A is the set of pairs of alternative arcs. A *selection* S of arcs from A is obtained by choosing at most one arc from each pair in the corresponding set. The selection is *complete* if exactly one arc is chosen from each alternative pair. A problem *solution* is represented by an alternative graph solution $(N, F \cup S)$ in which S is a complete selection. The solution is feasible if the graph contains no positive length cycles. A partial selection S *implies* the alternative arc (i, j), belonging to the unselected pair $((i, j), (h, k)) \in A$, if the selection $S' = S \cup \{(h, k)\}$ is infeasible, i.e., if the graph $(N, F \cup S')$ contains a positive length cycle.

The alternative graph has been used to model and solve train scheduling problems in several papers [5, 4, 7, 9, 10, 11, 19, 8, 25]. The main value of this formulation is the detailed and flexible representation of network topology and signaling system. In case of fixed block signaling, each block signal corresponds to a node in the alternative graph and the arcs between nodes are used to model the blocking times. The alternative graph represents the routes of all trains in a given control area along with their precedence constraints (minimum headways) and release times. Since a train must traverse the block sections in its route sequentially, a train route is modeled in the alternative graph with a job that is a chain of operations (modeled by nodes) and associated precedence constraints (modeled by fixed arcs). This formulation requires that a feasible route 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 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. At each block section, a passing order between trains must be therefore defined. This is modeled in the alternative graph by introducing a suitable pair of alternative arcs for each pair of trains traversing the 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 is no positive length cycle in the graph (i.e., the schedule is deadlock-free). In order to evaluate a schedule, we use the maximum consecutive delay as performance indicator of a solution. The consecutive delay is the delay introduced when solving conflicts in the dispatching area under study, and is caused by the propagation of the initial delays to the other trains.

Figure 5.3 shows the alternative graph solution for the area X of the illustrative example of Figure 5.1. The dummy nodes 0 and $*$ are used to model the release times and the objective function, respectively. The other nodes indicate pairs <train, block section>, e.g. node $B1$ represents train B running on block section 1. We observe that there are no pairs of unselected alternative arcs in the graph since the train sequencing on each block section of this area is forced by the initial position of trains A and B.

We next describe the six procedures we use to solve the dispatching problem. Four are published procedures, three heuristics and an exact algorithm described in [11]. The fifth

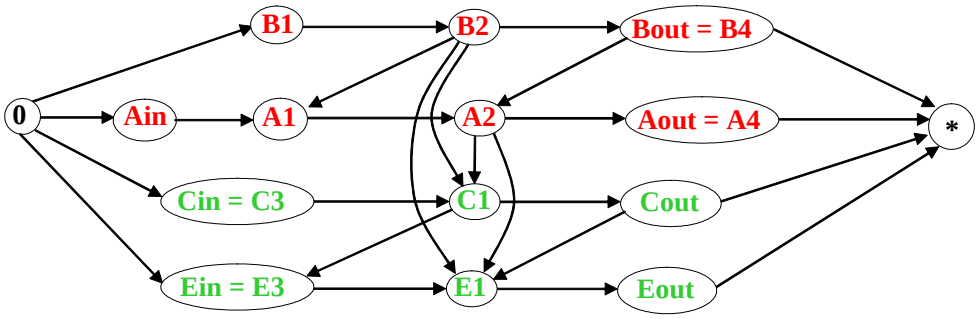


Figure 5.3: Alternative graph solution for the local scheduler of area X.

is a new heuristic used as a stand alone procedure or incorporated in the branch and bound algorithm [11]. In total we have four heuristics and two exact algorithms.

The First Come First Served (FCFS) dispatching rule, often used in the railway practice, solves train conflicts by assigning each block section to the first train that requires it. In other words, FCFS gives precedence to the train arriving first at each 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 scheduled 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: (i) give priority to the fast trains over the slow ones and (ii) FCFS.

The third heuristic is the AMCC (Avoid Most Critical Completion time) algorithm [11]. This heuristic belongs to the family of the Arc Greedy Heuristics (AGH) [21]. All the heuristics in the AGH family have the same algorithmic structure and are based on the idea of repeatedly extending a feasible selection. At each step, one unselected pair from the set A is chosen and one of the two alternative arcs is added to the current selection, until a complete selection is built or a positive length cycle is detected. The algorithms in the family differ only for the evaluation criterion applied to select the next alternative arc at each step. AMCC evaluates the consecutive delay that would be achieved by selecting each of the unselected alternative arcs, chooses the one which would cause the largest increase and avoids its selection by selecting the other arc of the pair.

The exact method that we consider in this paper, referred to in the following as BB, is the branch and bound algorithm described in D'Ariano et al. [11]. Also this algorithm is based on the alternative graph and computes optimal train schedules by using static and dynamic implications, including speed-ups based on the infrastructure topology. BB uses as initial solution the best one computed by FCFS, FLFS and AMCC.

The new heuristic is called Job Greedy Heuristic (JGH). This is also based on the idea of repeatedly extending a feasible selection. However, differently from AGH, the decision taken at each step consists in the insertion of a complete job in the current partial schedule, until a solution is found or an infeasibility is detected. As shown in Figure 5.4, at each iteration the insertion procedure tries to extend a current selection by selecting all the alternative pairs involving arcs between one of the previously inserted jobs and a new candidate job. The new selection is obtained by applying one or more arc greedy heuristics to the restricted problem and by retaining the best solution found. In order to choose the next job to insert, several candidate jobs are evaluated for insertion by building the new selection. The candidate job achieving the best evaluation is permanently inserted in the partial schedule and a new iteration begins. Possibly all the uninserted jobs are evaluated for insertion as candidate. If no feasible insertion is found the algorithm reports an infeasibility. To speed up the computation, JGH evaluates the insertion of at most K jobs at each step, where K is the minimum between the number of remaining uninserted jobs, a fraction Y of the total job number or a maximum number of insertions W. The K jobs chosen among the uninserted jobs are those with minimum release times. If no feasible insertion is found among the first K jobs, the algorithm tries to insert also the remaining uninserted jobs, one at a time ordered for increasing release time, until a feasible insertion is found, if any.

In a preliminary test phase, we tuned the JGH algorithm on a set of five timetable perturbations. The best configuration uses five AGH criteria (ALCP, AMBP, SLCP, SMBP, SMSP [21]) and parameters $Y = 0.2$ and $W = 10$.

The sixth algorithm, referred to as BB+, is simply obtained from BB by adding JGH to the set of heuristics used to compute the initial solution. We test BB and BB+ to evaluate the contribution of JGH to the performance of the branch and bound procedure.

5.3.4 Coordination model and procedures

A well known approach to face large and hard scheduling problems consists of decomposing them into smaller affordable sub-problems, which can be solved independently from each other. The local solutions are then composed to obtain a solution to the original problem. In this section we apply this kind of approach to reschedule train movements in a large area with dense traffic. The computational complexity of the train scheduling problem is limited within an acceptable level by network decomposition. The scheduling decisions in each local area are taken in a fully parallel fashion by the local schedulers. Since the composed solution must be globally feasible, the main issue is to coordinate the local schedulers in such a way that the union of all local solutions is globally feasible.

Given a division of the network, we adopt a compact representation of variables and con-

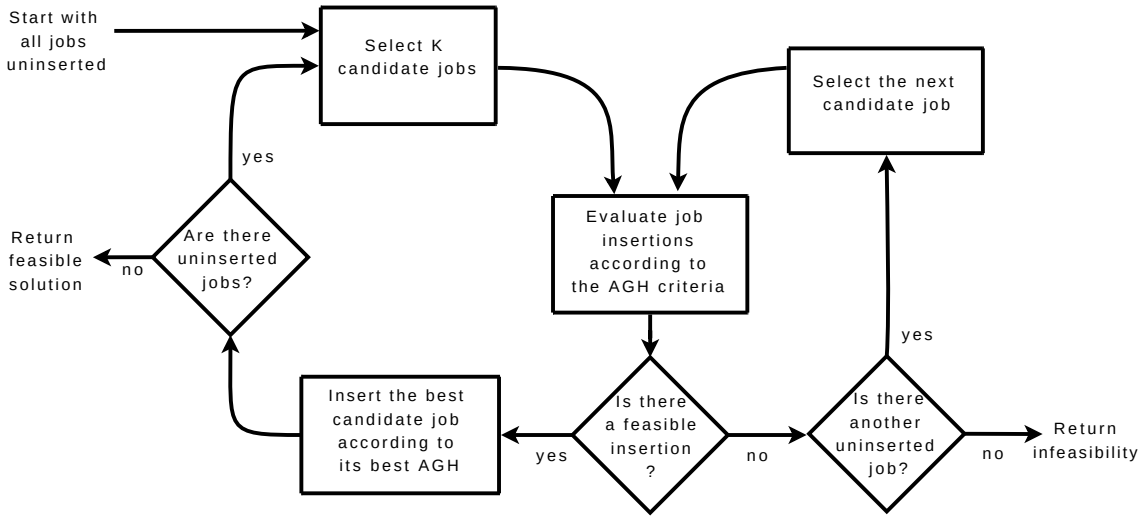


Figure 5.4: General scheme of the JGH procedure.

straints of each local area in order to limit the size of the set of data to be managed by the coordinator, avoiding the coordination task to become the bottleneck of the procedure. Each local dispatcher sends to the coordinator the entrance/exit time of each train traversing its borders and, for each pair of trains entering/leaving the local area, the minimum temporal distance between the two events. This information is used to build the *border graph* $G_B = (V_B, A_B)$, which consists of the set V_B of all border nodes and the set A_B of all border arcs. Border nodes are associated to the operations representing trains crossing the borders between areas. Let b_i, b_j be two border nodes, the set of border arcs is defined as follows. If a directed path from b_i to b_j exists in the graph associated to a solutions of a local area, the arc (b_i, b_j) is added to A_B with weight equal to the length of a longest path from b_i to b_j in the local graph. As proved in [4], the local solutions produced by the local schedulers are globally feasible if and only if there are no positive length cycles in the resulting border graph. Figure 5.5 shows a possible positive length cycle in the border graph of the illustrative example of Figure 5.1.

The coordination procedure is based on the border graph and makes use of the information provided by the local dispatcher and the heuristic rule described in [4]. With respect to the former procedure, we added a pre-processing phase in which each local scheduler sends to the coordinator the precedences among trains that must be kept in order to find a feasible solution. These are all the longest paths between border nodes in the graph associated to the initial selection $(N, A \cup S^0)$ of its area, i.e., the selection S^0 is obtained by the constraints on the initial position of each train. These constraints are then propagated on the alternative graph by using the static and dynamic implications described in [11]. With reference to the example in Figure 5.1(b), the local scheduler of area X finds a positive length path from the exit of B to the entrance of E (see the path with arcs

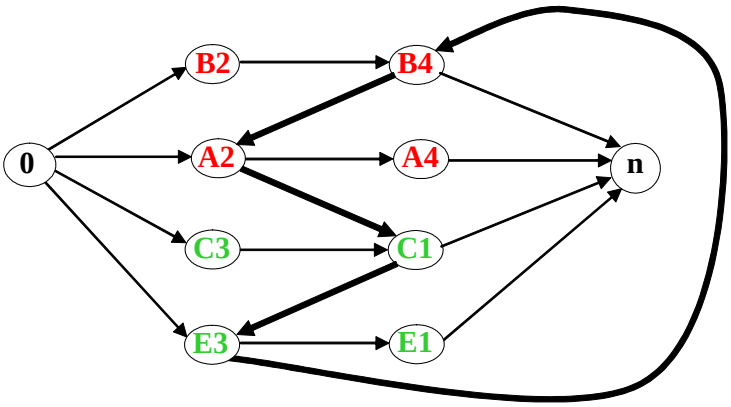


Figure 5.5: Positive length cycle in the border graph (in bold).

(Bout,A2),(A2,C1) and (C1,Ein) in Figure 5.3). This information becomes a precedence constraint $(B4, E3)$ with positive weight in area Y. Figure 5.6 shows part of the alternative graph of area Y.

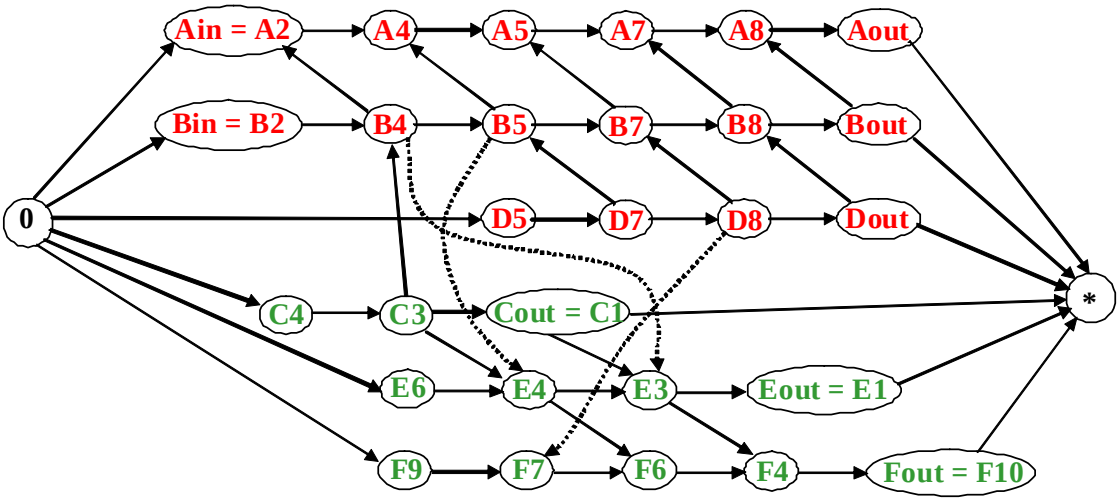


Figure 5.6: Initial selection for the alternative graph of area Y.

The solid arcs are the arcs implied by the initial position of the trains (redundant arcs are not depicted). The 7 pairs of unselected alternative arcs are not depicted in Figure 5.6, i.e., the alternative arcs: $((A5, E4), (E3, A4))$, $((B5, E4), (E3, B4))$, $((A5, F4), (F10, A4))$, $((B5, F4), (F10, B4))$, $((A8, F7), (F6, A7))$, $((B8, F7), (F6, B7))$ and $((D8, F7), (F6, D7))$. The 3 dotted arcs are not included in the initial selection of area Y. However, if the coordinator imposes the precedence constraint (arc) $(B4, E3)$ to the local scheduler of area Y, the initial selection with $(B4, E3)$ implies $(B5, E4)$, and the latter constraint implies $(D8, F7)$. Clearly, the alternative of the dotted arcs would

introduce a positive length cycle in the graph of area Y.

Figure 5.7 presents the general coordination procedure for n local areas, after the pre-processing phase. Firstly, each local scheduler computes the train scheduling solution of the corresponding local area. The local solutions, if locally feasible, are used to build the border graph. The border graph is used to check the global feasibility of the local solutions. A number of border constraints between each pair of adjacent dispatching areas have to be satisfied: (i) the exit time of a train from one area must be equal to the entrance time in the adjacent area, (ii) trains traversing a border on the same block section must keep the same order at the exit from an area and at the entrance in the adjacent area, (iii) locally feasible solutions should not cause deadlock situations from a global perspective. Such constraints cannot be checked directly by the single local dispatcher, since each sub-problem has a myopic view of the entire process. In [4], we show that simple exchange of information between neighboring areas is sufficient when there are only two areas. In case of three or more areas, a global overview of all areas is necessary, i.e., the coordinator must take decisions in order to avoid any possible infeasibility. In case of a global infeasibility, the coordinator may either impose precedence constraints among some trains at border nodes or may impose the same value for the exit time from an area and the entrance time in the subsequent area for some train. This is done by an update strategy that imposes new constraints at the borders. Each local scheduler then computes a new schedule in which these constraints are satisfied. The iterative procedure terminates when a globally feasible solution is found or when a local infeasibility is found or when the time limit of computation is reached.

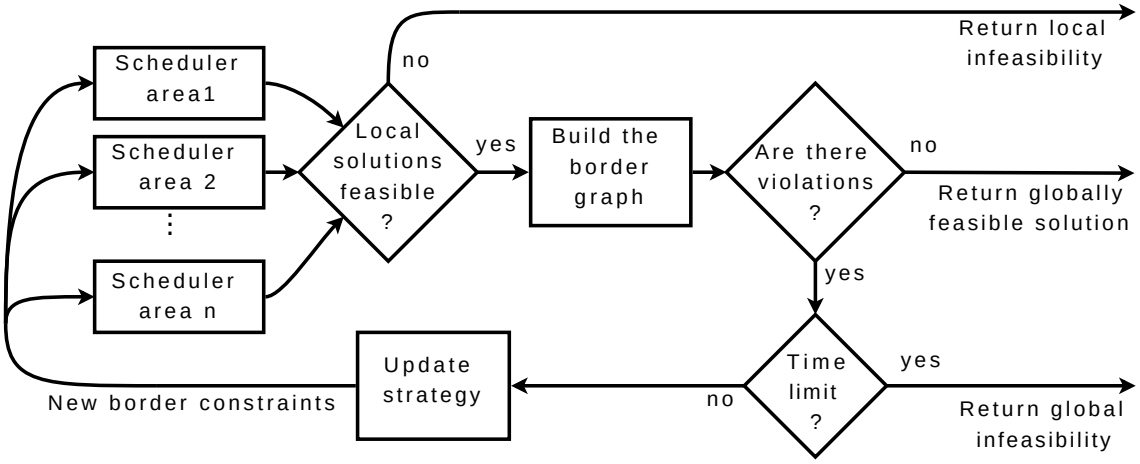


Figure 5.7: Scheme of the general coordination procedure.

The update strategy adopted in this paper works as follows. For each pair of adjacent local dispatchers, the strategy checks all border constraints for possible violation and takes the following actions in order to remove any violation. For each violated constraint (i) the coordinator sets the release time of the train entering the next area equal to the exit time of

the same train from the previous area. For each violation (ii), if the infeasibility involves two trains running in the same direction, the train order of the previous area is imposed to be the same in the following area. On the other hand, if the infeasibility involves two trains running in opposite directions, the train order is obtained by giving precedence to the first train reaching the border in the two schedules of both local areas. For a violation (iii), the coordinator detects a positive length cycle in the border graph. In this case, there must be at least one train, among those involved in the cycle, for which the exit time from an area is larger than the entrance time in the next area computed by the two associated local schedulers. The coordinator selects among these trains the earliest exit time τ scheduled by the local dispatchers, and then sets the release time of the associated train in the next area equal to τ. In other words, the coordinator sets only one additional constraint for one train and an area with this violation. Then, the local scheduler of the subsequent area is asked to reschedule trains with the new release time. Chances are that in the next iteration this train is scheduled in a different way so there will be no more positive length cycle in the border graph.

5.3.5 Overview of the different techniques

Table 5.1 shows differences and similarities of the scheduling algorithms from the perspective of local or global decisions and local or global information used to take decisions.

Table 5.1: Comparison of train scheduling algorithms.

Algorithm	Decisional level	Information Level
FCFS	Local	Local
FLFS	Local	Local
AMCC	Local	Global
JGH	Global	Global
BB	Global	Global
Coordination	Local	Global

To summarize, FCFS and FLFS are two representative local rules that take train ordering decisions at each railway junction by means of simple and local decision criteria. The information used is the time at which each train reaches the specific junction. Differently, AMCC and JGH use global information by means of the alternative graph formulation. AMCC takes local decisions, i.e., at each step the heuristic procedure decides the precedence between two trains at a specific block section. JGH is the only considered heuristic that applies global decisions by scheduling the whole path of a train at each step. Clearly, the exact methods implemented for the alternative graph formulation (i.e., BB and BB+) are based on global information and decisions. Finally, the coordinator uses local schedulers with global information on their area (i.e., BB, BB+ or JGH), while the coordination

procedure is based on global information on the overall network and on local decisions at the borders between pairs of dispatching areas.

5.4 Computational experiments

This section presents our computational results on a large part of the Dutch railway network, and evaluates the dispatching and coordination approaches of Section 5.3. The algorithms are implemented in C++ and run on a PC equipped with a dual-core processor Intel Pentium D (3 GHz), 1 GB Ram and Linux operating system. We adopted a time limit of 10 seconds for each local scheduler of the distributed architecture and a time limit of 300 seconds for the overall centralized/distributed procedures.

5.4.1 Test case description

We study a railway network in the South-East of the Netherlands that spans over ten dispatching areas of the Dutch railway network. The network layout comprises a combination of single and double-tracks of different length, with a diameter of 300 km. In total, there are more than 1200 block sections and stopping platforms at stations. The network studied includes the major stations of Utrecht Central, Arnhem and Den Bosch, plus other 40 minor stations. The two main traffic directions are served by the line between Utrecht and Arnhem (towards Germany) and the line between Utrecht and Den Bosch (from Amsterdam towards Eindhoven and the southern part of the country).

For the distributed architecture, we consider the two network decompositions of Figure 5.8. For the 3-area division the border stations are Bunnik, Utrecht Lunetten and Nijmegen Dukenburg, while for the 6-area division the border stations are also Den Bosch Oost, Arnhem Zuid, Utrecht Terwijde and Utrecht Zuilen.

The reference timetable is periodic with a periodicity of one hour. Train traffic includes local and intercity services. We consider three time horizons of traffic prediction in order to investigate delay propagation on graphs of increasing size. Table 5.2 reports information on the alternative graph for the centralized scheduler, the 3-area local schedulers and the 6-area local schedulers. Column 1 reports the time horizon of reference (in minutes), Columns 2-5 the average number of scheduled trains, fixed arcs, nodes and alternative pairs of the associated alternative graph of each local area. When enlarging the time horizon, we obtain a linear growth of the number of trains and nodes while the number of alternative pairs increases quadratically.

Table 5.3 shows information on the border graphs for the two divisions. Column 1 reports

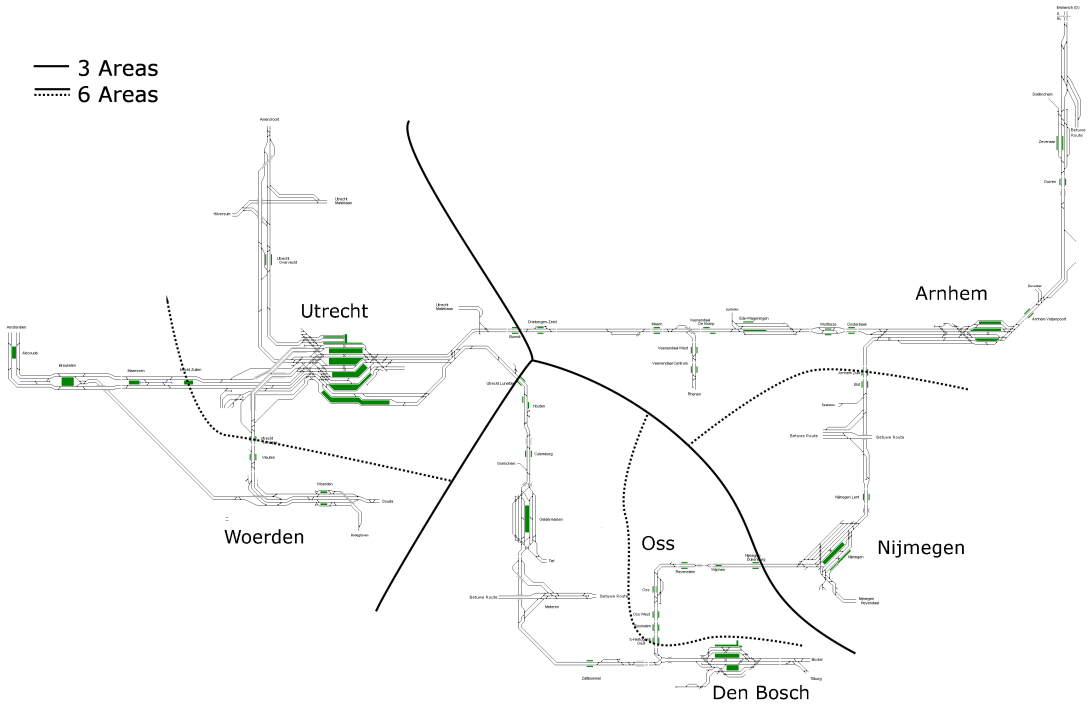


Figure 5.8: The studied railway network and two divisions in local areas.

Table 5.2: Alternative graph properties for each scheduler.

Time Horiz	Trains	Fixed Arcs	Nodes	Alternative Pairs
Centralized Scheduler				
30 min	99	3508	3081	3019
60 min	154	6968	6136	14528
90 min	205	10398	9171	34660
3-Area Local Schedulers				
30 min	40	1207	1055	1026
60 min	65	2398	2102	4935
90 min	89	3577	3141	11773
6-Area Local Schedulers				
30 min	25	628	550	529
60 min	44	1257	1096	2544
90 min	62	1877	1638	6065

the time horizon of reference (in minutes), Column 2-3 (4-5) show the number of arcs and nodes for the 3-area (6-area) division. When enlarging the time horizon, we obtain a rather limited number of arcs and nodes in the border graph.

Table 5.3: Border graph properties for each division.

Time Horiz	3-Area division		6-Area division	
	Arcs	Nodes	Arcs	Nodes
30 min	124	44	299	102
60 min	407	86	1044	208
90 min	852	128	2189	310

We consider random variations of the entrance times of all trains running in the network. The real-life stochasticity of train operations is modeled by the statistical fitting procedure of the Weibull distribution described in [7] (included in this thesis as Chapter 3). We analyze a total amount of more than 33000 train events (arrivals, departures, dwell processes and passing times) that have been recorded at Utrecht Central Station in April 2008 by ProRail. Trains experience a deviation on their entrance time in the overall network that we call "small perturbations" in this section. We also analyze "large perturbations" that are obtained by doubling the scale parameter of the Weibull distribution. The latter perturbations are artificially generated in order to study the effects of the dispatching and coordination procedures in the presence of more disturbed traffic situations.

Table 5.4: Description of the entrance delays.

Time Horiz	Small Perturbations		Large Perturbations	
	Max Delay (s)	Avg Delay (s)	Max Delay (s)	Avg Delay (s)
30 min	523	21.7	1351	301.5
60 min	558	22.7	1377	295.9
90 min	567	17.7	1485	277.2

The entrance delays are reported in Table 5.4 for three time horizons of traffic prediction. Each row of Table 5.4 reports the average value (in seconds) of the maximum and average entrance delays over 5 different delay instances. Since there are 2 sets of timetable perturbations (small and large) and 3 time horizons of traffic prediction (30, 60 and 90 minutes), we generated 30 problem instances in total.

5.4.2 Timetable perturbations

This subsection presents the performance of 12 combinations of the six dispatching algorithms described in Section 5.3 with different network divisions. Figure 5.9 reports the percentage of globally feasible solutions found by the 12 combinations for the management of the overall network. Each algorithm is evaluated over all the instances of Table

5.4. The algorithms are defined by the two-field code $\alpha\ \beta$, where α is the algorithm and β is the number of local areas. Clearly, $\beta = 1$ requires only the centralized scheduler, otherwise we use the coordination system plus the local schedulers.

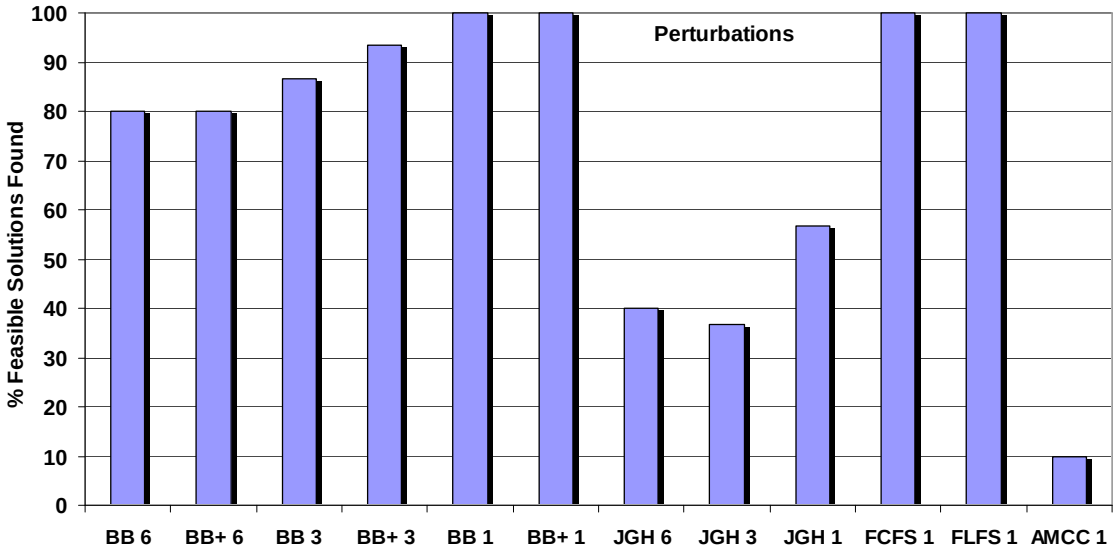


Figure 5.9: Feasibility of the algorithms for the timetable perturbations.

From the results of Figure 5.9 we decide to restrict the detailed assessment of our computational results to a subset of scheduling algorithms as follows. Concerning the algorithms for the centralized scheduler, the two practical dispatching rules (FCFS 1 and FLFS 1) are always able to compute a feasible solution for this set of instances, but we only select FCFS 1 since the two rules present very similar results in terms of solution quality. The other two heuristics (AMCC 1 and JGH 1) find less feasible solutions and we decided to skip AMCC 1. Both the versions of our branch and bound algorithm (BB 1 and BB+ 1) are taken since they offer good solutions in acceptable computation time. In the distributed architecture, for further analysis we select only BB+ that offers more globally feasible solutions compared to both BB and JGH.

Tables 5.5 and 5.6 report more information on the selected algorithms for the small and large perturbations, respectively. Each row of these tables presents the average result over 5 delay instances for a given algorithm and time horizon of traffic prediction. The best value of each row is emphasized in bold. For each algorithm we show the percentage of globally feasible solutions, the percentage of globally optimal solutions, the maximum and average consecutive delays (in seconds) and the computation time (in seconds). For the branch and bound algorithms of the centralized scheduler (i.e., BB 1 and BB+ 1), the latter indicator is the time to compute the best solution.

From the results of Table 5.5, all the algorithms but JGH 1 are able to compute a globally

Table 5.5: Performance of the procedures: Small perturbations.

Time Horiz	Indicator	FCFS 1	JGH 1	BB 1	BB+ 1	BB+ 3	BB+ 6
30 min	Feasible Solut (%)	**100**	**100**	**100**	**100**	**100**	**100**
	Optimal Solut (%)	0	**60**	**60**	**60**	**60**	**60**
	Max Cons Delay (s)	512	**189**	253	**189**	**189**	**189**
	Avg Cons Delay (s)	12.2	5.8	7.3	5.7	4.9	**4.1**
	Comp Time (s)	**0.2**	2	1	3	16	8
60 min	Feasible Solut (%)	**100**	**100**	**100**	**100**	**100**	**100**
	Optimal Solut (%)	0	**40**	**40**	**40**	**40**	**40**
	Max Cons Delay (s)	770	286	541	286	**283**	301
	Avg Cons Delay (s)	16.4	9.7	12.3	9.7	6.5	**5.8**
	Comp Time (s)	**0.2**	62	37	67	30	44
90 min	Feasible Solut (%)	**100**	20	**100**	**100**	**100**	**100**
	Optimal Solut (%)	0	0	0	0	0	0
	Max Cons Delay (s)	833	783	769	718	606	**480**
	Avg Cons Delay (s)	19.3	18.4	17.7	17.2	10.6	**7.9**
	Comp Time (s)	**1**	279	34	288	102	168

feasible solution. When JGH 1 returns no solution, we impose a failure penalty for the maximum and average consecutive delays equal to the worst solution value found by the other algorithms. In order to compare the different results, we consider the same type of failure penalty for the results of Table 5.6.

Table 5.6: Performance of the procedures: Large perturbations.

Time Horiz	Indicator	FCFS 1	JGH 1	BB 1	BB+ 1	BB+ 3	BB+ 6
30 min	Feasible Solut (%)	**100**	60	**100**	**100**	**100**	**100**
	Optimal Solut (%)	0	0	**60**	**60**	**60**	**60**
	Max Cons Delay (s)	492	299	286	249	**231**	243
	Avg Cons Delay (s)	23.5	20.0	17.9	18.9	16.0	**12.8**
	Comp Time (s)	**0.2**	5	3	6	10	11
60 min	Feasible Solut (%)	**100**	60	**100**	**100**	60	40
	Optimal Solut (%)	0	0	0	0	0	0
	Max Cons Delay (s)	714	604	714	**586**	731	665
	Avg Cons Delay (s)	35.1	39.4	35.1	39.3	34.7	**33.6**
	Comp Time (s)	**0.4**	165	**0.4**	172	63	115
90 min	Feasible Solut (%)	**100**	0	**100**	**100**	**100**	40
	Optimal Solut (%)	0	0	0	0	0	0
	Max Cons Delay (s)	960	-	960	960	**770**	968
	Avg Cons Delay (s)	40.3	-	40.3	40.3	**29.3**	35.3
	Comp Time (s)	**1**	-	**1**	**1**	111	167

For both tables, the distributed approaches (i.e., BB+ 3 and BB+ 6) find the same percentage of optimal solutions compared to the branch and bound algorithms of the centralized approach (i.e., BB 1 and BB+ 1). Moreover, the average consecutive delay of the distributed approaches is always less than the other algorithms. The latter result shows that

the distributed management of the whole network can generate better quality solutions. However, FCFS 1, BB 1 and BB+ 1 are able to find more feasible solutions than the distributed approaches, since the latter approaches require to be coordinated by imposing additional constraints at the area borders. For this reason, BB+ 6 is able to compute less globally feasible solutions than BB+ 3. We also observe that the maximum consecutive delay of the FCFS 1 solutions is 50% larger than the best one obtained by the other algorithms. Differently, JGH 1 is the worst algorithm in terms of the percentage of feasible solutions but improves considerably the performance of the branch and bound algorithm, see for example BB+ 1 versus BB 1.

Concerning the computation time, FCFS 1 is clearly the fastest algorithm in both tables, even for large time horizons. The other algorithms require a few seconds in order to compute/search for a solution for the 30-minute time horizon. In case of larger time horizons, the distributed approaches find a globally feasible solution in less than 3 minutes while the centralized approaches need more time for the 90-minute time horizon. In particular, JGH 1 is quite time consuming for large time horizons and finds no feasible solutions within the given time limit of computation for the 90-minute time horizon.

5.4.3 Disrupted traffic situation

We next study how to reschedule trains in case of an infrastructure disruption that requires heavy modifications of the train routes in the timetable. A blockage of a single track is simulated on the double track line that connects Utrecht to Den Bosch, between the stations of Zaltbommel and Den Bosch. Trains of both traffic directions have to run on the other track or have to be globally rerouted (see Figure 5.10).

In the original timetable, 12 trains per hour (6 per direction) are scheduled on the disrupted line, four of which are local services between Utrecht and Den Bosch while the other eight are intercity services connecting the northern region of the country with the southern region. We apply the following modification of the train routes in order to recover the disrupted scenario: Four intercity trains (2 trains per hour per direction) and four local trains (2 trains per hour per direction) are still scheduled on the line Utrecht - Den Bosch. In the vicinity of the disruption, the trains are locally rerouted for a stretch of 6 km, along the only available track that now serves a bidirectional flow. This results in a shortage of capacity that leads to delays propagating throughout the network. The other four intercity trains (2 train per hour per direction) are globally rerouted, i.e., they change their routes but keep the same origin and destination stations. The latter trains are rerouted via the line going to Nijmegen and Arnhem, or vice versa. The running time required for the alternative trip between Utrecht and Den Bosch is around 40 minutes longer than the original trip time, which is 30-minute long.

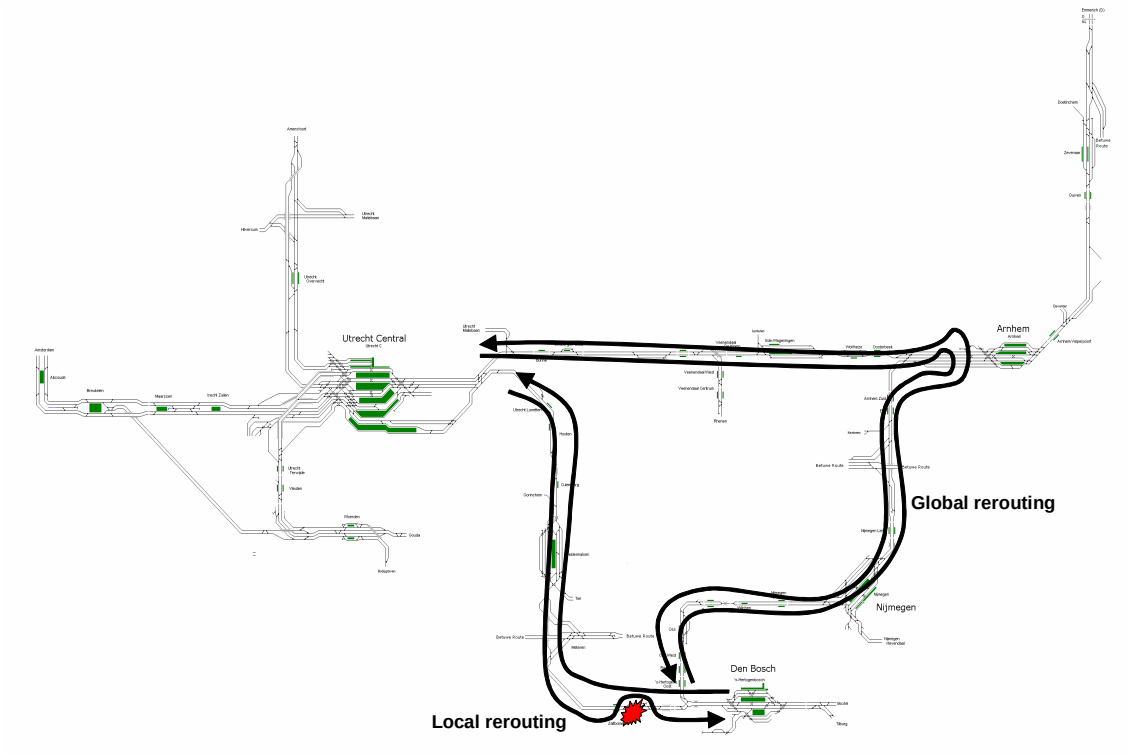


Figure 5.10: Disrupted timetable with local and global rerouting.

A graphical representation of the two timetables, original and disrupted, is reported for comparison in Figure 5.11. Every solid line indicates that there are two trains running per hour per direction on a specific line. Light green lines are local services and dark blue lines are intercity services. The dotted line represents one international service scheduled per hour. The track blockage is represented by the cross between the stations of Geldermalsen and Den Bosch.

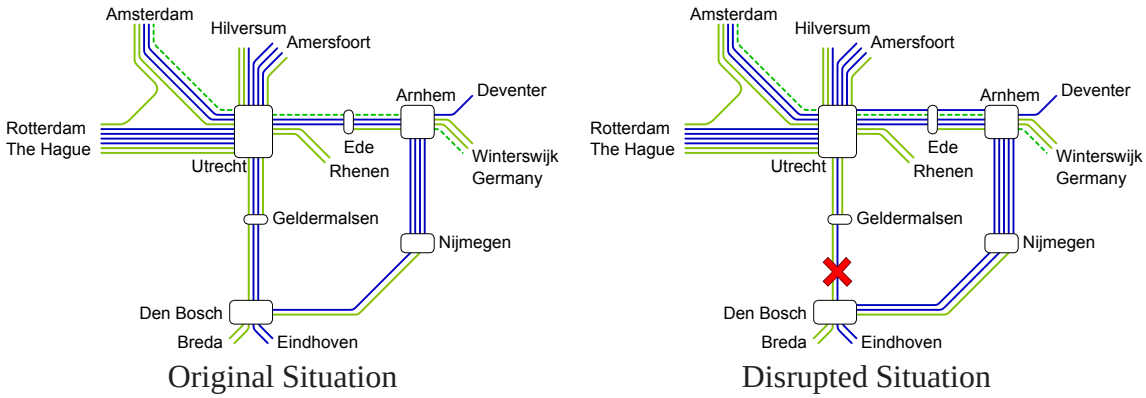


Figure 5.11: Available train services for the two traffic situations.

Figure 5.12 reports the performance of all the algorithms in delivering globally feasible solutions for the disrupted situation of Figure 5.11 and under the timetable perturbations described in Section 5.4.1. We limit our analysis of each algorithm to the 20 instances of Table 5.4 with 30-minute and 60-minute time horizons. The instances of the 90-minute time horizon are not considered because in this case the algorithms find a few feasible solutions only. In fact, the traffic management of the disrupted situation is more difficult compared to the original situation, since there is a serious risk of deadlock due to the larger number of trains that share the single track section in both traffic directions.

We conclude from Figure 5.12 that in the disrupted situation the distributed approaches perform better than the centralized ones. Specifically, BB+ 3 is always able to compute a globally feasible solution for all the instances. Among the centralized approaches, BB+ 1 and JGH 1 obtain more feasible solutions than the others.

Tables 5.7 and 5.8 give more information on the same subset of algorithms reported in Tables 5.5 and 5.6. Each row of the new tables presents the average result over 5 delay instances for a given algorithm and time horizon of traffic prediction. The best value of each row is again emphasized in bold. For each algorithm we show the percentage of globally feasible solutions, the maximum and average consecutive delays (in seconds) and the computation time (in seconds). For the branch and bound algorithms of the centralized scheduler (i.e., BB 1 and BB+ 1), the latter indicator is the time to compute the best solution, if any, or the given time to search for a feasible solution. We do not report the percentage of globally optimal solutions since in this case BB 1 and BB+ 1 were not able

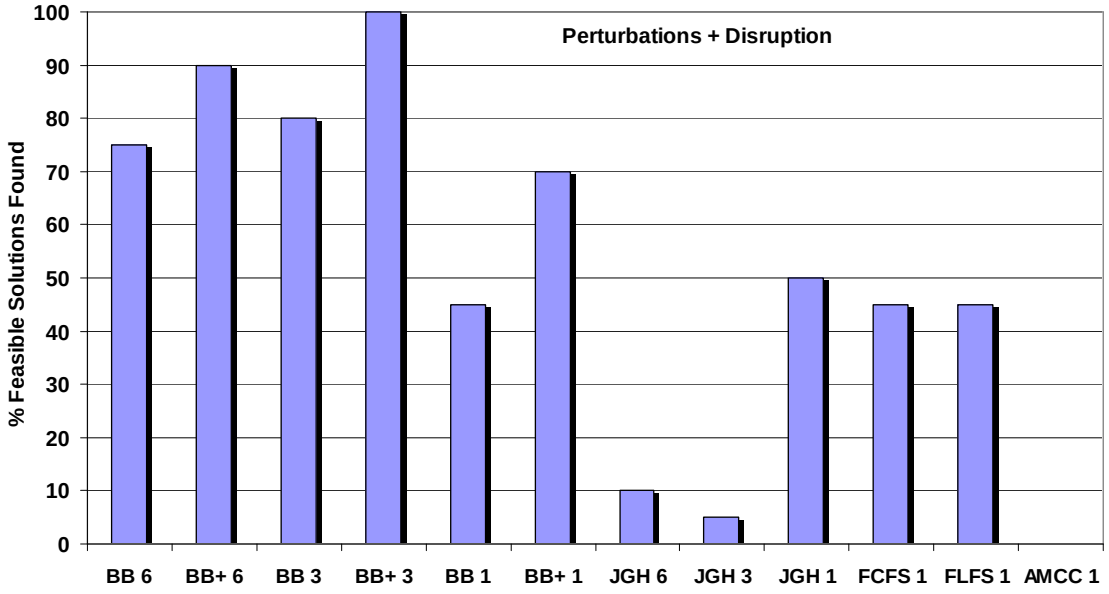


Figure 5.12: Feasibility of the algorithms for the disrupted situation.

to prove optimality within the time limit of computation. If one algorithm does not find a globally feasible solution for one instance, we consider the same type of failure penalty used in Tables 5.5 and 5.6.

Table 5.7: Performance of the procedures: Small perturbations and disruption.

Time Horiz	Indicator	FCFS 1	JGH 1	BB 1	BB+ 1	BB+ 3	BB+ 6
30 min	Feasible Solut (%)	**100**	80	**100**	**100**	**100**	**100**
	Max Cons Delay (s)	700	577	688	577	688	**390**
	Avg Cons Delay (s)	19.7	24.9	18.8	24.9	12.3	**9.0**
	Comp Time (s)	**0.1**	8	0.3	10	30	16
60 min	Feasible Solut (%)	20	20	20	40	**100**	**100**
	Max Cons Delay (s)	1225	1133	1225	1133	1225	**812**
	Avg Cons Delay (s)	24.5	24.6	24.5	24.5	17.8	**12.6**
	Comp Time (s)	**0.4**	253	108	280	78	119

From the results of Table 5.7, the distributed approaches are always able to compute a globally feasible solution for small perturbations. The centralized approaches generate feasible solutions for all the instances with 30-minute time horizon but often fail with larger instances. Regarding to the solution quality, for this set of instances the best algorithm is BB+ 6 and requires, on average, less than two minutes of computation.

The algorithms show a slightly similar performance in Table 5.8 compared to Table 5.7. Again, the distributed approaches yield a larger number of globally feasible solutions compared to the centralized approaches, even if this number is lower than in Table 5.7

Table 5.8: Performance of the procedures: Large perturbations and disruption.

Time Horiz	Indicator	FCFS 1	JGH 1	BB 1	BB+ 1	BB+ 3	BB+ 6
30 min	Feasible Solut (%)	40	60	40	80	**100**	**100**
	Max Cons Delay (s)	555	467	544	450	393	**346**
	Avg Cons Delay (s)	28.8	31.8	28.7	28.7	23.4	**17.7**
	Comp Time (s)	**0.2**	9	180	67	45	17
60 min	Feasible Solut (%)	20	40	20	60	**100**	60
	Max Cons Delay (s)	1137	**952**	1137	**952**	1137	1009
	Avg Cons Delay (s)	59.9	60.1	59.9	59.9	**34.7**	43.5
	Comp Time (s)	**0.2**	266	105	293	80	142

due to the larger entrance delays. The average consecutive delay is also better minimized by the distributed approaches with respect to the centralized ones. On the other hand, the maximum consecutive delay obtained by JGH 1 (and BB+ 1) for the instances of the 60-minute time horizon is smaller than the one of BB+ 3 and BB+ 6. Finally, since the computation time of JGH 1 and BB + is more than three minutes for the 60-minute time horizon, these algorithms should only be applied for shorter time horizons during disrupted operations.

5.4.4 Discussion

The computational experiments report the performance of the centralized and distributed approaches for various traffic disturbances. The limits of the algorithms are evaluated in terms of their ability to find globally feasible solutions for time horizons of traffic prediction of increasing length. On the whole, for the original traffic situation and for any type of timetable perturbation the branch and bound algorithm of the centralized approach and the distributed approaches can easily compute a globally feasible solution in a few seconds for small instances, with 30-minute time horizon. The new scheduling algorithm (JGH) is quite effective in finding good solutions even if this often fails in finding a feasible solution. The combination of JGH with BB (i.e., BB+) clearly outperforms both JGH and BB in terms of number of feasible solutions and solution quality. In this practical case, the solutions are also often globally optimal, and the distributed approach outperforms the centralized one in terms of average consecutive delay minimization. The problem becomes considerably hard to tackle as the magnitude of the time horizon increases.

When dealing with the disrupted traffic situation, only the distributed approach is able to compute a feasible solution for all the timetable perturbation instances. This is consequence of the lack of spare capacity in the network that causes a major propagation of the entrance delays. Specifically, the local rerouting of trains generates bidirectional traffic on the single track adjacent to the disrupted track (nearby Zaltbommel), while the global rerouting requires more trains sharing the block sections of the alternative routes. Central-

ized algorithms experience increasing difficulty in managing the increasing complexity of the train scheduling problem and the high probability of conflicts and deadlocks. The distributed approach is the most suitable to reschedule trains after the disruption and this is quite effective in dealing with local and global reroutings. However, the problem of coordinating multiple local scheduling solutions is still a complicated issue and the probability of finding a feasible solution decreases for an increasing number of local areas. However, we observe that when distributed algorithms do not find a globally feasible solution this is always due to one or more local infeasibilities, i.e., the time limit of computation is never reached for the coordination procedure.

The study on large time horizons is useful in order to quantify the propagation of train delays in the network. When enlarging the time horizon up to 60 minutes, there is a serious increase of the consecutive delays, specially in the blocked track case. In the traffic situation without disruption, the 90-minute time horizon seems to be less perturbed than the 60-minute time horizon since the entrance delays are progressively absorbed by the time reserves of the timetable in the second hour of traffic prediction.

5.5 Conclusions and future research

This paper compares centralized and distributed approaches for managing the disturbance handling process accurately. We propose an extensive study of dispatching and coordination algorithms in terms of feasibility, quality and computation time. A new dispatching algorithm, JGH, is proposed to schedule one train per time in the centralized approach. We also extend a distributed approach for the coordination of two areas to the general case of k areas. From the computational results we conclude that when dealing with short time horizons of traffic prediction, the proposed algorithms are able to compute good quality solutions in a very short time of computation. In case of larger time horizons of traffic prediction, severe traffic disturbances and blocked tracks, the distributed approach presents a better feasibility performance than the centralized one.

Further research should be dedicated to a number of issues: (i) development of more robust coordination algorithms for a large number of areas; (ii) development of coordination algorithms able to drive local schedulers towards globally optimal other than feasible solutions; (iii) analysis of different railway network decompositions, disrupted traffic situations and alternative timetables; (iv) use of the proposed approaches in order to validate draft country-wide timetables at the detailed level of signals and block sections; (v) multi-objective approaches for taking multi-criteria train scheduling decisions, considering e.g. the objectives of infrastructure managers and train operating companies.

5.6 Acknowledgements

This work is partially supported by the Dutch program “Towards Reliable Mobility” of the Transport Research Centre Delft and by the Italian Ministry of Research, Grant number RBIP06BZW8, project FIRB “Advanced tracking system in intermodal freight transportation”.

Bibliography

[1] Y.-H. Cheng and L.-A. Yang. A fuzzy petri nets approach for railway traffic control in case of abnormality: Evidence from taiwan railway system. *Expert Systems with Applications*, 36(4):8040–8048, 2009.

[2] Y. H. Chou, P. F. Weston, and C. Roberts. Dynamic Distributed Control for Real-time Rescheduling of Railway Networks. In I. A. Hansen, A. Radtke, J. Pachl, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

[3] Y. H. Chou, P. F. Weston, and C. Roberts. Collaborative rescheduling in a distributed railway control system. In Hansen, Wendler, Weidmann, Luethi, Rodriguez, Ricci, and Kroon, editors, *Proceedings of the 3rd International Seminar on Railway Operations Modelling and Analysis*. Zürich, Switzerland, 2009.

[4] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Centralized versus distributed railway traffic control. In *Proceedings of the 11th International Conference on Advanced Systems for Public Transport (CASPT09)*. Hong Kong, China, 2009.

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

[6] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo. Centralized versus distributed systems to reschedule trains in two dispatching areas. *Public Transport: Planning and Operations*, 2010 (accepted for publication).

[7] F. Corman, A. D'Ariano, M. Pranzo, and I.A. Hansen. Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area. In Hansen, Wendler, Weidmann, Luethi, Rodriguez, Ricci, and Kroon, editors, *Proceedings of the 3rd International Seminar on Railway Operations Modelling and Analysis*. Zürich, Switzerland, 2009.

[8] Pacciarelli D. Deliverable D3: Traffic Regulation and Co-operation Methodologies code wp4ur_dv_7001_d. In *project COMBINE 2 "enhanced COntrol centres for fixed and Moving Block sIgNalling systEms" Number: IST-2001-34705*. 2003.

[9] A. D'Ariano. *Improving Real-Time Train Dispatching: Models, Algorithms and Applications*. PhD Thesis, TRAIL Thesis Series T2008/6, The Netherlands, 2008.

[10] A. D'Ariano, F. Corman, D. Pacciarelli, and M. Pranzo. Reordering and local rerouting strategies to manage train traffic in real-time. *Transportation Science*, 42 (4):405–419, 2008.

[11] A. D'Ariano, D. Pacciarelli, and M. Pranzo. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183(2):643–657, 2007.

[12] A. Fay. A fuzzy knowledge-based system for railway traffic control. *Engineering Application of Artificial Intelligence*, 13:719–729, 2000.

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

[14] R. V. Iyer and S Gosh. DARYN – A distributed decision-making algorithm for railway networks: Modeling and simulation. *IEEE Transactions on Vehicular Technology*, 44 (1):180–191, 1995.

[15] L.-M. Jia and X.-D. Zhang. Distributed intelligent railway traffic control: A fuzzy-decision making-based approach. *Engineering Applications of Artificial Intelligence*, 7(3):311–319, 1994.

[16] E. Lamma, P. Mello, and M. Milano. A distributed constraint-based scheduler. *Artificial Intelligence in Engineering*, 11:91–105, 1997.

[17] T. S. Lee and S Gosh. Stability of RYNSORD – A decentralized algorithm for railway networks under perturbations. *IEEE Transactions on Vehicular Technology*, 50 (1):287–301, 2001.

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

[19] M. Mazzarello and E. Ottaviani. A traffic management system for real-time traffic optimisation in railways. *Transportation Research, Part B*, 41 (2):246–274, 2007.

[20] G. Parodi, G. Vernazza, and F. Zunino. Stability and deadlock avoidance in distributed system for traffic control. *IEEE Transactions on Vehicular Technology*, 45 (4):732–743, 1996.

[21] M. Pranzo, C. Meloni, and D. Pacciarelli. A new class of greedy heuristics for job shop scheduling problems. In *Lecture Notes in Computer Science*, volume 2647, pages 223–236. 2003.

[22] J. Rodriguez. A constraint programming model for real-time train scheduling at junctions. *Transportation Research, Part B*, 41 (2):231–245, 2007.

[23] İ. Şahin. Railway traffic control and train scheduling based on inter-train conflict management. *Transportation Research, Part B*, 33:511–534, 1999.

[24] M. A. Salido, M. Abril, F. Barber, L. Ingolotti, P. Tormos, and A. Lova. Domain-dependent distributed models for railway scheduling. *Knowledge-Based Systems*, 20:186–194, 2007.

[25] C. Strotmann. *Railway Scheduling Problems and their decomposition*. PhD thesis, Universität Osnabrück, 2007.

[26] J. Törnquist and J. A. Persson. N-tracked railway traffic re-scheduling during disturbances. *Transportation Research, Part B*, 41 (3):342–362, 2007.

[27] G. Vernazza and R. Zunino. A distributed intelligence methodology for railway traffic control. *IEEE Transactions on Vehicular Technology*, 39 (3):263–270, 1990.

[28] S. Wegele, R. Slovák, and E. Schnieder. Real-time decision support for optimal dispatching of train operation. In I. A. Hansen, A. Radtke, J. Pachl, and E. Wendler, editors, *Proceedings of the 2nd International Seminar on Railway Operations Modelling and Analysis*. Hannover, Germany, 2007.

[29] P. Zhu. *Betriebliche Leistung von Bahnsystemen unter Störungsbedingungen*. PhD thesis, Berichte der Institute für Automatisierungstechnik, TU Braunschweig, 2001. In German.

Chapter 6

Evaluation of green wave policy in real-time railway traffic management

Abstract *In order to face the expected growth of transport demand in the next years, several new traffic control policies have been proposed and analyzed both to generate timetables and to effectively manage the traffic in real-time. In this paper, a detailed optimization model is used to analyze one such policy, called green wave, which consists in letting trains waiting at the stations to avoid speed profile modifications in open corridors. Such policy is expected to be especially effective when the corridors are the bottleneck of the network. However, there is a lack of quantitative studies on the real-time effects of using this policy. To this end, this work shows a comparison of the delays obtained when trains are allowed or not to change their speed profile in open corridors. An extensive computational study is described for two practical dispatching areas of the Dutch railway network.*

This chapter is an edited version of the article

F. Corman, A. D'Ariano, D. Pacciarelli, M. Pranzo, Evaluation of green wave policy in real-time railway traffic management, presented at *10th International Conference on Applications of Advanced Technologies in Transportation*. Athens, Greece, 2008, and published in *Transportation Research Part C*, Vol. 17, No. 6, December 2009, pages 607-616.

Reprinted with permission from Elsevier.

6.1 Introduction

The problem faced by railway managers when rescheduling trains in real-time, is called conflict detection and resolution (CDR) problem. Given a timetable and the real-time position of the trains, the CDR problem consists of rescheduling train movements such that the deviation from the timetable is minimized, the new schedule is compatible with the actual train positions and there exists a feasible speed profile for each train achieving the schedule and compatible with signaling constraints.

Signaling systems vary quite a lot from country to country. The standard feature of a railway signaling system is characterized by the three-aspect fixed block signaling. This paper refers to the Dutch signaling system, though the concepts apply as well to most signalling systems. A signal aspect may be either red, yellow or green. A red signal aspect at the entrance of a block section means that the block section is either out of service or occupied by another train, a yellow signal aspect means that the block section is empty while the successive one is occupied, a green signal aspect means that the two subsequent block sections are empty.

A train is allowed to enter a block section at its *scheduled speed* SS only if the signal aspect is green. It must switch to a prescribed *approaching speed* AS (40 km/h in the Netherlands) if the signal aspect is yellow and it must stop before a red signal aspect.

Figure 6.1 shows a typical speed profile of a train facing a red signal aspect while traversing a line with five signals. The first signal aspect is green, which enables the train to traverse the subsequent block section at its scheduled speed SS. The aspect of the second signal is yellow, therefore the train decreases its speed to AS until the next signal aspect, which happens to be red. The train must stop until the signal aspect turns to yellow, and when this happens it increases its speed up to AS. When arriving at the sight distance from the next signal, it can accelerate up to SS if the fourth signal aspect is green.

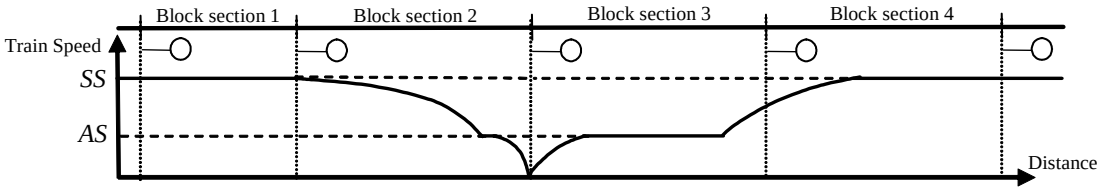


Figure 6.1: Three aspects Dutch signaling system.

Signals ensure safety of railway traffic, since the track segment between two signals can host at most one train at a time. However, signals are also used to regulate railway traffic, and different policies are currently being studied by railway managers to improve traffic

performance in terms of train punctuality and energy consumption. This paper analyzes two such policies. With a WIC (*Wait In Corridors*) policy, trains are allowed to wait both in stations and along the lines, i.e., they can meet yellow or red signal aspects along open corridors. With a GW (*Green Wave*) policy, trains are allowed to wait only at their scheduled stops. Each train can leave a station only if it can reach the following station traveling at its scheduled speed profile, therefore facing only green signal aspects. In other words, while the GW policy permits only dwell time extensions at scheduled stops, the WIC policy allows modifying both running times and dwell times. The former policy is expected to be effective when station capacity is larger than open corridor capacity. The latter one is more flexible but requires the implementation of a speed regulation policy to adjust train speed profiles according to the actual signal aspects.

In the literature on conflict resolution the CDR problem is approached with various techniques and adopting several policies. Cordeau *et al.* [4] classify approaches to the CDR problem into fixed and variable speed models. In view of their extensive survey we limit the literature review to the most recent contributions. With a fixed speed model, the CDR problem is solved by adopting fixed train speed profiles. The suggested train orders and routes are then used to compute a feasible trajectory for each running train that satisfies the signaling constraints. This common approach has been recently used, among the others, within the European Project COMBINE [12, 14, 13] In D'Ariano *et al.* [7], the following iterative scheduling procedure is described to manage railway traffic in presence of disturbances. First, a CDR problem with fixed speed profiles is solved by a branch and bound algorithm. Then, a train speed coordination procedure detects whether the minimum required headway distances are respected and adjusts the speed profile of the first train facing yellow and/or red signal aspects. After this speed profile adjustment, a new CDR problem is formulated and solved by the branch and bound algorithm. The iterative scheduling procedure terminates in a finite number of iterations and returns a conflict-free timetable with feasible speed profiles. The solution proposed by this algorithm can be improved by computing locally new trajectories that further reduce delays and energy consumption, as in D'Ariano and Albrecht [5]and Albrecht [1].

The discussed models are designed according to a WIC policy in which train speed profiles are modified in open corridors with respect to the scheduled ones. An alternative approach is to solve the fixed speed CDR problem in such a way that scheduled speed profiles are always feasible. This is obtained by adding the constraints that a minimum headway distance between consecutive trains must be always satisfied, reducing the need for trains to brake and accelerate in open corridors, resulting in smoother and more efficient operation. This is also the aim of the GW policy, which imposes trains to follow exactly their scheduled speed profile, thus avoiding the need for speed profile adjustments. Asuka and Komaya [2] propose a similar approach for the management of a metropolitan line.

This paper raises the issue of defining the operating conditions under which one policy is preferable with respect to the other. A detailed model is given for the resolution of the CDR problem subject to either the WIC or GW policies, based on the alternative graph formulation of Mascis and Pacciarelli [11] and on the blocking time theory introduced by Happel [9]. Effective solution algorithms allow to solve the CDR problem based on this formulation. Extensive computational experiments are reported to assess the performance that can be achieved with the two policies in terms of train delays and energy consumption for different perturbations scenarios. The proposed computational study is based on two practical and complicated dispatching areas of the Dutch railway network, namely the Schiphol bottleneck area [10] and the Utrecht Den Bosch dispatching area. The former is a small and complex area subject to high frequency traffic, while the latter consists of longer open corridors, heterogeneous traffic and lighter traffic conditions. Finally, the results obtained are discussed and possible criteria are defined to select an appropriate policy for a given railway network. However, this paper focuses more on the development of tools for assessing a policy rather than on defining general recommendations for the selection of the best policy. In fact, the quantitative benefit of a policy depends on a number of factors like the infrastructure characteristics, rolling stock and traffic conditions, and besides, on the tools available for traffic management. Therefore, the final choice should be taken after careful evaluation of the specific operating conditions of the area under study.

The paper is organized as follows. The next section presents the alternative graph formulation of the CDR problem with the WIC and GW policies, and discusses the differences between the two models with an illustrative example. Then, the three scheduling algorithms used in the computational experiments are briefly described, namely the branch and bound algorithm of [6]. and two dispatching rules similar to those frequently adopted by Dutch railway traffic controllers [3]. The third section reports the computational results. Some conclusions on the proposed traffic management policies and suggestions for further research follow.

6.2 Models and Algorithms

Since the pioneering work of Szpigel [16], the job shop scheduling model has been used by many authors to formulate the problem of scheduling train movements in a railway network. In the last years the basic model has experienced an increase in the level of details incorporated in the formulation, the most relevant being the introduction of no-store and no-wait constraints, as in Mascis *et al.* [12]. With this model the sequence of block sections traversed by a train are viewed as a set of machines in a job shop scheduling problem, where trains correspond to jobs. The traversing of a block section by a train is called an operation o_i, and requires a running time p_i. We refer to $o_{\sigma(i)}$ and $o_{\mu(i)}$ to denote the operations following and preceding o_i, respectively.

We use the alternative graph of Mascis and Pacciarelli (2002) to model the railway scheduling problem, which associates a node i to the starting time t_i of each operation o_i and introduces an arc (i, j) to model a precedence relation between operations o_i and o_j, the weight on the arc representing a minimum time interval between t_i and t_j. Arcs are *fixed* when the precedence has to be satisfied in each feasible solution. For example, arcs $(\mu(i), i)$ and $(i, \sigma(i))$ are fixed and represent the fact that $o_{\mu(i)}$, i and $o_{\sigma(i)}$ are processed in sequence. The two arcs are weighted with the running times $p_{\mu(i)}$ and p_i of the associated train on the block sections associated to $o_{\mu(i)}$ and o_i, respectively.

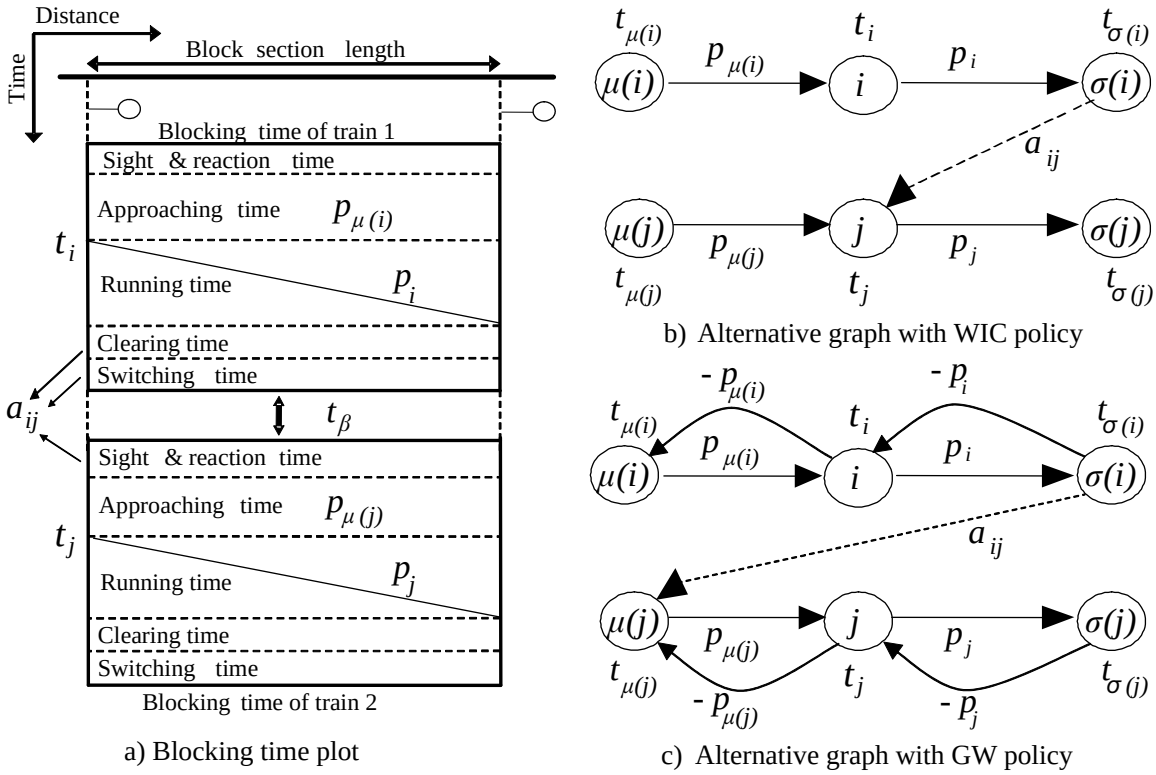


Figure 6.2: Alternative graph formulations of the blocking time theory.

Inter-train conflicts arise when two trains require the same block section at the same time. Conflicts are solved giving precedence to one train over the other. In the alternative graph model, conflicts are modeled with pairs of arcs, called *alternative arcs*. Alternative arcs take into account railway safety rules that prevent a train to enter a block section which is occupied by another train and to enter at high speed a block section if the two block sections that follow are not empty. We adopt the blocking time theory described in Pachl [15] to model the minimum time interval between the exit of a train from a block section and the entrance of the following train on the same or on the preceding block section (see Figure 2(a)). These values are called *clearing time*, *switching time* and *sight & reaction*

time, the sum of these three values, that we call *setup time*, is the weight of the alternative arc.

Different traffic control policies are translated into different alternative graph models of the railway scheduling problem. Figure 2(b) shows the alternative graph model for the WIC policy. In this case a train can enter a block section not earlier than the exit of the preceding train plus the setup time. Figure 2(c) shows the alternative graph model for the GW policy, which requires all trains to strictly follow their scheduled speed profiles. This constraint is modeled with no-wait constraints between the starting times of consecutive operations for each train, which forbid variations in the running times. In turn, the no-wait constraints are modeled through fixed arcs with negative weight, as in Mascis and Pacciarelli [11]. Moreover, since trains must always meet green signals along open corridors, there must be a distance of at least two empty block sections between each pair of consecutive trains. Therefore, the minimum approaching time of a train equals the running time on the previous block section of its path. This is represented with the alternative arc a_{ij} in Figure 2(c).

The alternative graph model is completed by inserting two additional dummy nodes and several fixed arcs to represent arrival/departure constraints. Specifically, dummy node 0 represents the starting time of traffic prediction and dummy node n takes into account the objective function. Since a train is not allowed to depart before its scheduled departure time, an arc is added from node 0 to each node associated to the exit of a train from a platform, weighted with its scheduled departure time. Similarly, an arc is added from node 0 to each node associated to the entrance of a train in the network. Finally, since a train is considered late if arriving at a station after its scheduled arrival time, an arc is added from the node i, associated to the entrance on a platform, to the dummy node n. The weight on arc (i, n) is equal to the scheduled arrival time, but negative, so that the length of the path from 0 to n, passing through i equals the delay of a train at a station, i.e., the difference between its actual arrival time and its scheduled arrival time. Similarly, an arc is added from the node i, associated to the exit of a train from the network, to the dummy node n.

The conflict resolution problem consists in choosing one arc from each alternative pair (i.e., a solution for each potential conflict), such that the resulting graph has no positive length cycles (i.e., the schedule is deadlock-free) and such that the length of the longest path from 0 to n is minimum (i.e., the maximum delay is minimum).

6.2.1 Illustrative example

Figure 6.3 shows a railway network with 8 block sections and a station. The two stops before the block sections 1 and 2 are not included in the network, so that trains entering

the network from these block sections are allowed to wait before entering the network. On the other hand, there is no stop before block section 5, so that incoming trains enter the network at their scheduled speed which should not be modified with the GW policy. Let us consider three trains traversing the network. Trains A and B share block sections 3, 4 and 8 and stop both at block section 4 and before entering the network. Train C traverses the network from block section 5 to block section 8 without stopping, and shares only the block section 8 with trains A and B. Only the most relevant signals are depicted in Figure 6.3.

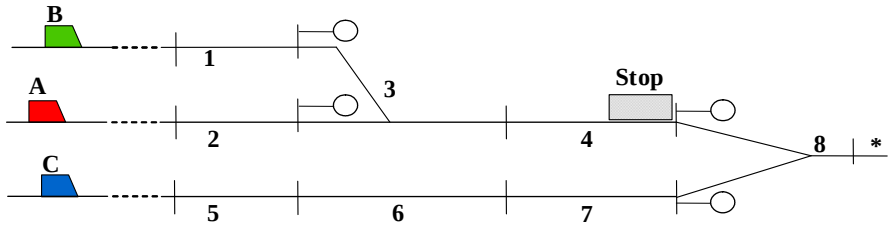


Figure 6.3: A small railway network with three running trains.

Figure 6.4 reports the alternative graph formulation of the WIC policy. Each node of the graph is denoted by the pair (train, block section), except for the dummy nodes. The number on each node corresponds to the associated block section, while the train is shown on the left of each route. The weight of the horizontal (fixed) arcs (not depicted in the figures) equals the running time of the trains on the different block sections or, alternatively, the dwell time of the trains on the stop platform. Arcs $(0, A2)$, $(0, B1)$ and $(0, C5)$ represent the entrance of each train in the network and are weighted with the minimum entrance time. Arcs $(0, A8)$ and $(0, B8)$ are weighted with the departure time of trains A and B from the scheduled stop. Arcs $(AStop, n)$, $(A*, n)$, $(BStop, n)$, $(B*, n)$ and $(C*, n)$ represent the arrival time of the trains at the scheduled stop or at the exit of the network, i.e., at the end of block section 8. For each pair of trains sharing a block section a sequencing decision must be taken. This is modeled with a pair of alternative arcs representing the two possible sequencing decisions. Alternative arcs are depicted by dotted lines.

Figure 6.5 illustrates the alternative graph formulation with the GW policy. In this case, trains A and B are allowed to wait before their entrance in the network and the weight on arcs $(0, A2)$ and $(0, B1)$ defines their minimum entrance time. The entrance time of train C cannot be changed since the speed profile of this train must be as scheduled. To force equality, arc $(C5, 0)$ is added with the same weight of arc $(0, C5)$ but with negative value. The other backward arcs added between two consecutive operations of the same train represent the no-wait constraints and force the trains to strictly follow their scheduled speed profiles in open corridors. The weight on each backward arc, say $(A4, A3)$, is the same of the forward arc $(A3, A4)$ but with negative value. To better describe the role of backward arcs, let us suppose that train A precedes train C on block section 8. Train C

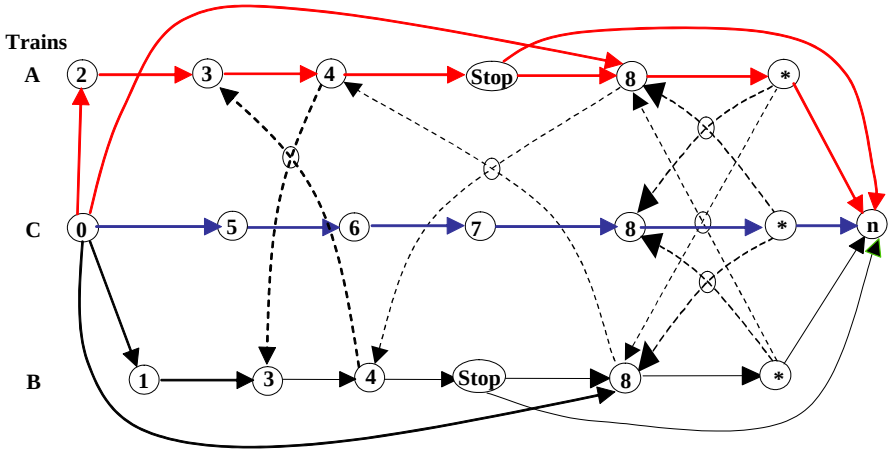


Figure 6.4: Alternative graph formulation with the WIC policy.

is delayed by A at block section 8 if and only if the path from 0 to $C7$ passing through $A2, A3, A4, AStop, A8, A*$ is longer than the path from 0 to $C7$ passing through $C5, C6$, and therefore if and only if the cycle highlighted in bold in Figure 6.6 has a positive length. This implies that train C does not follow its scheduled speed profile if and only if the solution is infeasible. Hence, if there are no positive length cycles in the graph associated to a solution, no train is delayed other than at its scheduled stops.

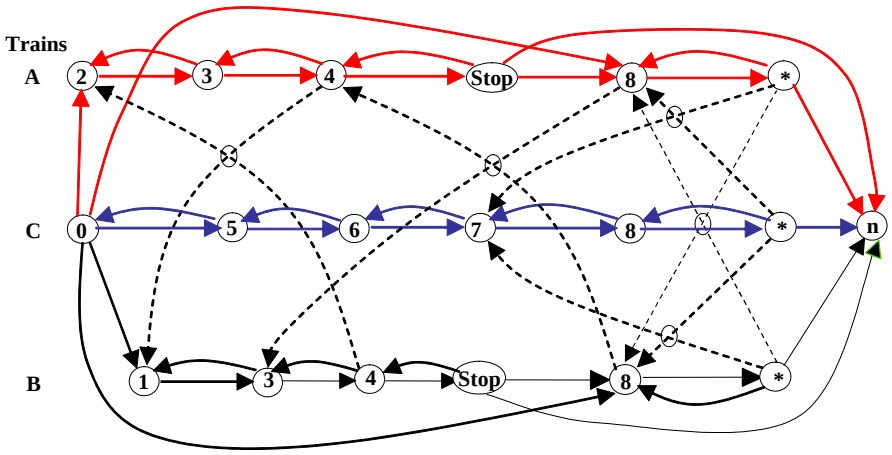


Figure 6.5: Alternative graph formulation with the GW policy.

Notice that the alternative pairs under the GW policy are different with respect to the WIC policy, since each train must meet green signal aspects along the lines, while it can meet also red signal aspects in stations. Hence, the alternative arcs associated to the train departure after a scheduled stop are the same as with the WIC policy, while trains running

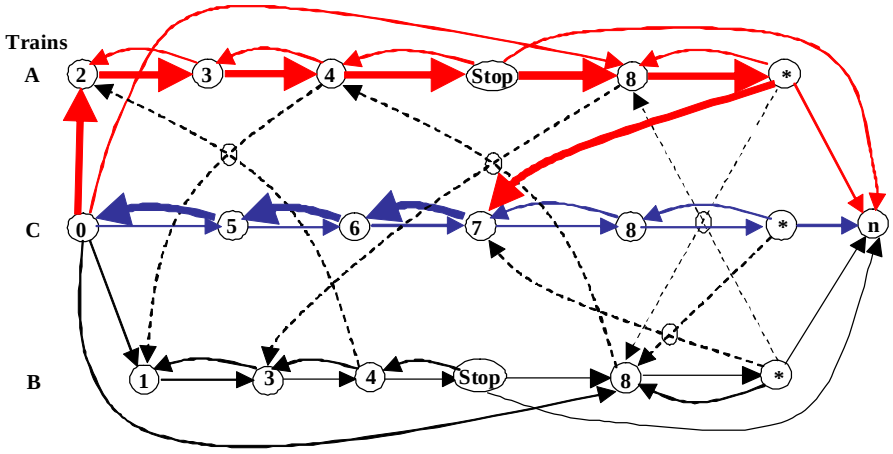


Figure 6.6: A cycle with the GW policy.

in open corridors need two empty block sections ahead. The latter constraints are modeled by the alternative arcs imposing a distance of two block sections for each pair of trains sharing a block section.

6.2.2 Scheduling algorithms

The three scheduling algorithms used in this paper are next described.

The first algorithm is the branch and bound (BB) described in D'Ariano *et al.* [6]. This algorithm can solve to optimality practical size instances of the fixed speed CDR problem within short computation time. For the purposes of this paper, we limit the algorithm execution to 120 seconds of computation. This limit is compatible with real-time operations and allows to compute proven optimal solutions in most instances.

The second algorithm simulates the practice of traffic management adopted in the Netherlands, which is based on the ARI system described by Berends and Ouburg (2005). This semi-automated system detects and solves train conflicts automatically when delays are contained in a predefined time-window specified by the dispatcher. In order to evaluate the effectiveness of a completely automated system, we implemented a version of the ARI system that simulates the behavior of the dispatchers with priority rules when delays exceed a time-window of four minutes. Precedence between trains is defined on the basis of the train type (first intercity, then regional and then freight) and, in case of tie, priority is given to the train with the smallest number of remaining scheduled stops.

The third algorithm is the well-known First In First Out (FIFO) dispatching rule, usually adopted in the railway practice, which is simply to give precedence to the train arriving

first at the current block section.

6.2.3 Speed adjustment

As discussed in the first section, the CDR solutions obtained with the WIC policy need the implementation of a speed adjustment procedure to satisfy the Dutch signaling system constraints. The speed adjustment procedure considered in this paper implements the basic drivers' behavior described in Figure 1 (proceed at speed SS in case of green signal aspect, decrease the speed to AS in case of yellow signal aspect, stop in case of red signal aspect). A more detailed discussion on speed adjustment and driver behavior can be found, e.g. in D'Ariano *et al.* [6]. We retain the order of trains defined in the solutions to the CDR problem, and adjust the traversing times for all trains facing yellow and red signal aspects.

This simple procedure enables the computation of feasible train trajectories and the prediction of the practical effects of the proposed schedules, in terms of energy consumption and delay minimization. Moreover, the speed profiles computed by the procedure can be implemented with typical railway infrastructure and do not require the implementation of modern driver support systems like those described, e.g., in Hansen and Pachl [8].

6.3 Computational Experiments

This section describes the computational experiments on a large sample of practical size instances based on two different dispatching areas of the Dutch railway network. For each dispatching area, we generate 72 instances simulating different configurations of late trains and entrance delays.

6.3.1 Description of instances

The first dispatching area is the line from Utrecht to Den Bosch depicted in Figure 6.7. It includes two double track corridors around 50 km long, 191 block sections, 7 passenger stations (Utrecht Lunetten, Houten, Houten Castellum, Culemborg, Geldermalsen, Zaltbommel and Den Bosch), a dedicated stop for freight trains in Zaltbommel and 21 platforms. The timetable for this network is hourly cyclic, and plans 26 trains per hour in the station of Geldermalsen, and up to 40 trains per hour in the station of Den Bosch.

Each instance for this network is generated by delaying a variable number of trains (from

1 to 8 trains), randomly chosen among those entering the network in the first half hour. The entrance delay of the late trains is also randomly chosen on the basis of a uniform distribution in the interval $[50, 4800]$ seconds. The 72 instances are grouped in 12 sets of 6 instances, each with a similar value of average entrance delay ranging approximately from 10 to 120 seconds. The alternative graph for each instance includes more than 3400 pairs of alternative arcs with the GW policy and more than 3500 with the WIC policy.

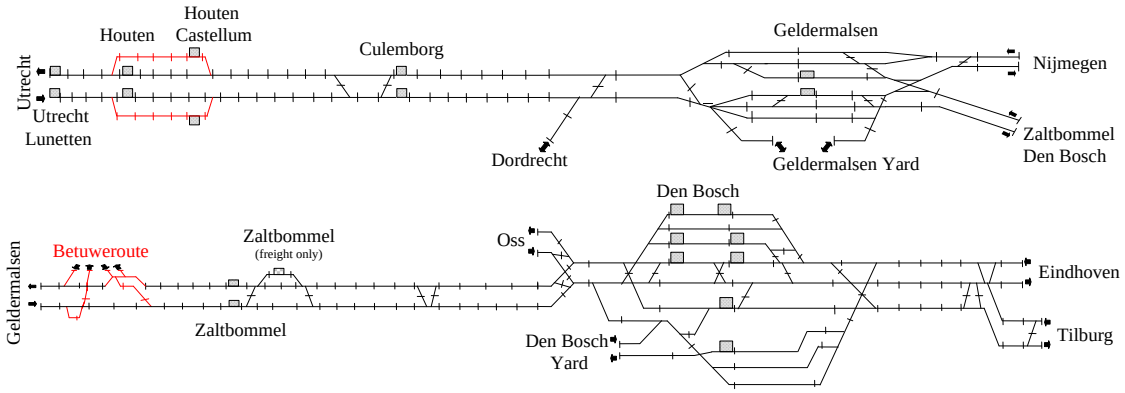


Figure 6.7: Utrecht Den Bosch dispatching area.

The second dispatching area shown in Figure 6.8 is a bottleneck of the Dutch railway network including the underground station of Schiphol, beneath the international airport of Amsterdam. This is a small but complex network around 20 km long, with 86 block sections and 4 platforms in the Hoofddorp station and 6 in the Schiphol station, plus other 8 at the borders of the network. The timetable considered for the experiments on this network is hourly cyclic, and includes 54 trains per hour. This is a variant of the timetable for year 2007 described by Hemelrijk *et al.* [10], with a larger number of trains than those actually operated for year 2007 (40 trains per hour). This timetable is considered very close to the capacity of the network, thus making it particularly challenging for CDR algorithms.

Each instance of this network is generated by delaying a variable number of trains (from 1 to 6 trains), randomly chosen among those entering the network in the first half hour. The entrance delay of the late trains is also randomly chosen on the basis of a uniform distribution in the interval $[100, 4500]$ seconds. The 72 instances are grouped in 8 sets of 9 instances, each with a similar value of average entrance delay ranging approximately from 10 to 110 seconds. The alternative graph for each instance includes more than 7600 pairs of alternative arcs with the GW policy and more than 8000 with the WIC policy.

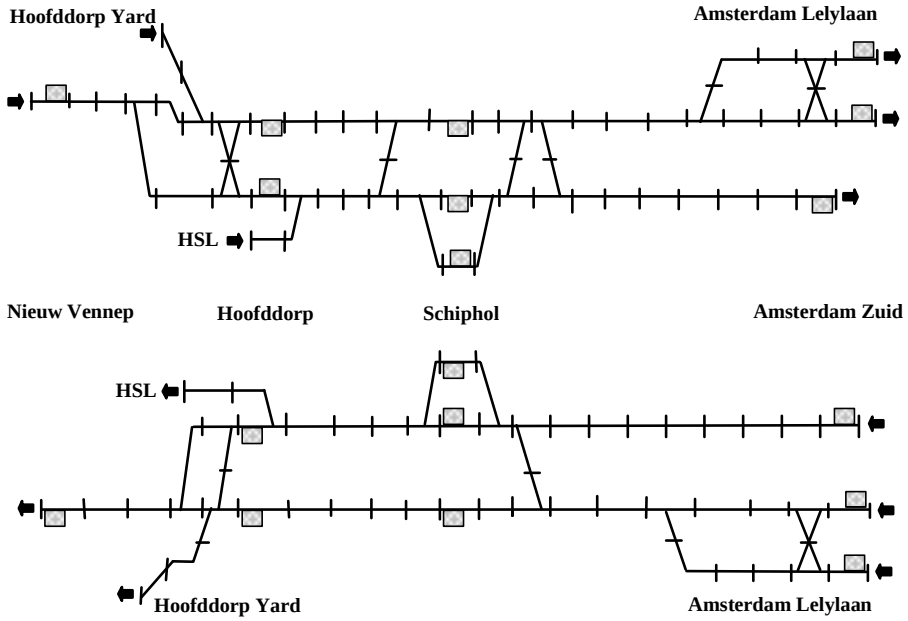


Figure 6.8: Schiphol dispatching area.

6.3.2 Performance of algorithms and policies

For each of the 144 instances, four different feasible schedules are computed by using the three CDR algorithms described in the previous section and the two alternative graph models based on the WIC and GW policies. The first two solutions (GW+BB and WIC+BB) are obtained with the Branch and Bound of D'Ariano *et al.* [6] based on the GW or WIC policies. The third and fourth solutions (WIC+ARI and WIC+FIFO) are obtained with the ARI or FIFO procedures based on the WIC policy. We do not report on the solutions obtained with the combinations GW+ARI and GW+FIFO since these procedures are not able to find feasible solutions for all instances. When using the WIC policy, feasible speed profiles are computed in a post-processing step by using the speed adjustment procedure described in the previous section.

All the algorithms are implemented in C++ language and executed on a PC equipped with a 3 GHz Pentium D processor, 1 GB Ram and Linux operating system. Each run of the BB procedure is terminated after 120 seconds of computation.

Figure 6.9 shows the performance of the four combinations policy+algorithm achieved for the Utrecht - Den Bosch and Schiphol dispatching areas. We also added a fifth curve, FS+BB, showing the delays obtained by the BB algorithm with fixed speed and without the speed adjustment procedure. The FS+BB values are not feasible in practice, but are interesting as they represent a lower bound on the performance achievable by advanced

speed optimization algorithms when the train orders are those computed by FS+BB. In all plots, the average input delays on the horizontal axis refer to the average delays of all trains, computed at the entrance of the network. The average output delays on the vertical axis are the average delays of all trains, computed at all scheduled stops and at the exit of the network. Each point in Figure 6.9 reports the average results over the six instances for the Utrecht - Den Bosch area and nine instances for the Schiphol area with similar average entrance delay.

A few comments are in order. For both areas the branch and bound algorithm clearly outperforms the ARI algorithm, while WIC+FIFO achieves similar results to WIC+BB for the Utrecht - Den Bosch area. Since the timetable for this area contains a larger amount of time reserve, simple CDR algorithms behave quite reasonably. On the other hand, the more challenging instances for the Schiphol network clearly show that advanced scheduling algorithms are needed when the scheduled traffic becomes closer to the network capacity. When comparing the WIC+BB and GW+BB solutions, it can be observed that the GW policy is slightly worse than the WIC policy for the Utrecht - Den Bosch area, while it is clearly better for the Schiphol area. These results are due to the large number of speed adjustments required by the BB solution with the WIC policy. In fact, the gap between the WIC+BB and FS+BB plots shows that the average increase in the delay caused by the speed adjustment procedure is more than 100 seconds for the Schiphol network. This gap leaves open the question on the extent at which combining the WIC policy with advanced speed optimization algorithms might improve the results of WIC+BB. For the Utrecht - Den Bosch area, the gap between the WIC+BB and FS+BB plots is much smaller, and therefore more sophisticated speed optimization algorithms would be less relevant in this area due to the larger amount of time reserve in the timetable.

Table 6.1 summarizes a number of results for the two dispatching areas. Each row refers to a given policy and to a given CDR algorithm. Each entry is the average over the 72 instances previously described. For each row and dispatching area, we report the average computation time, the average energy consumption and the average output delays expressed in percentage with respect to the reference case GW+BB.

It is interesting to observe that the computation time needed to solve the CDR problem and to adjust the speed profile of each train running in the Schiphol network is more than one order of magnitude greater than for the Utrecht - Den Bosch network. Specifically, more than 30 seconds of computation are required by the speed adjustment procedure to solve the instances of the Schiphol area, versus around 2 seconds for the Utrecht - Den Bosch area. With the GW policy, instances are significantly harder to solve than with the WIC policy, mainly due to the addition of the no-wait constraints in the alternative graph formulation. For the Schiphol area, the average time required by the BB algorithm increases from 33.3 seconds with the WIC policy to 99.5 seconds with the GW policy. On the other hand, when considering the overall time needed to compute a feasible solution, it

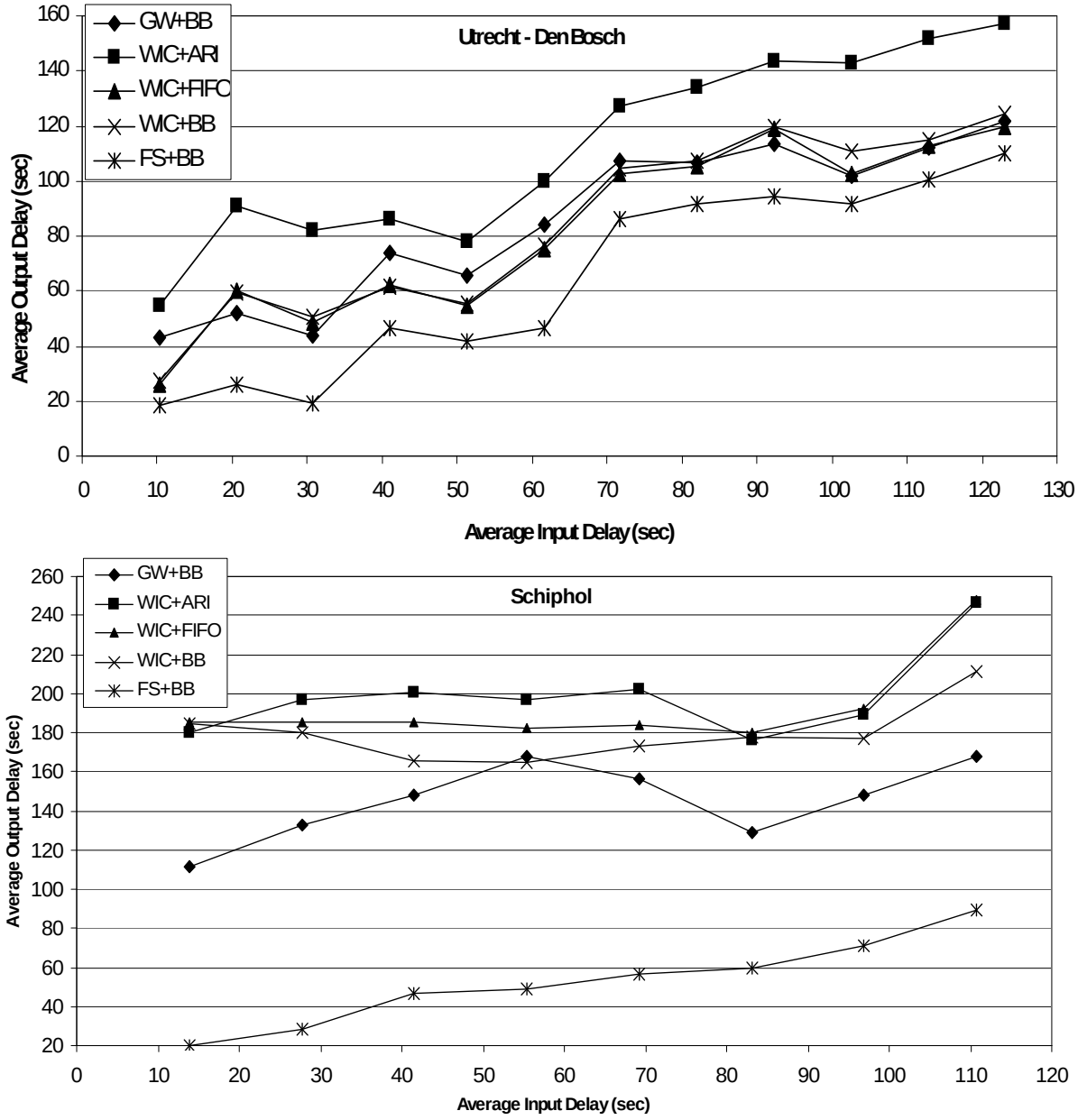


Figure 6.9: Average results in terms of input versus output delays.

Table 6.1: Comparison of combined scheduling algorithms and policies.

Policy + Scheduling Algorithm	Utrecht - Den Bosch			Schiphol		
	Comput. Time	% Energy Cons.	% Average Output Delay	Comput. Time	% Energy Cons.	% Average Output Delay
GW+BB	1.2	100.0	100.0	99.5	100.0	100.0
WIC+ARI	2.8	108.1	131.5	39.9	115.6	136.8
WIC+FIFO	2.2	107.5	96.4	34.6	116.2	132.9
WIC+BB	2.4	107.4	98.7	33.3	116.2	123.5
FS+BB	0.1	-	76.3	0.4	-	29.3

turns out that the instances of the Utrecht - Den Bosch area and the GW policy require the smallest computation time, since there is no need for speed adjustment and these instances are easy to solve by the BB procedure.

As far as the energy consumption is concerned, the GW policy exhibits a remarkable saving with respect to the WIC policy. For the Utrecht - Den Bosch area, there is a difference of about 7% between the two policies. This gap increases to more than 15% for the Schiphol area, due to the larger number of brakings and accelerations required by the WIC policy.

From an overall point of view, Table 6.1 clearly shows that the GW policy is a very promising alternative to the WIC policy, at least when no advanced speed optimization procedure is available to control traffic. This is evident in terms of both energy consumption and delay minimization.

6.3.3 Discussion

Several considerations on the GW and WIC policies follow from an interpretation of the results obtained for the two dispatching areas.

The Schiphol dispatching area is short and densely occupied but traffic is homogeneous, with passenger trains and low speed differences. The longer Utrecht - Den Bosch dispatching area contains a larger amount of time reserve but traffic is heterogeneous, with mixed passenger and freight trains and significant speed differences. The ratio between the capacity available at the platforms and that of the corridors is quite larger for the Den Bosch station with respect to the Schiphol station.

For both areas, the output delay is frequently larger than the input delay, the difference being larger for the Schiphol network. This behavior is likely due to quite different factors

for the two networks and for the two policies. In the Schiphol network, the performance of WIC+BB is mainly limited by the traffic in the corridors, close to capacity saturation, while the performance of GW+BB are mainly limited by the reduced capacity available at the platforms. In the former case every braking penalizes the arrival of subsequent trains, in the latter case train arrivals are penalized by late departures of previous trains. For the Utrecht - Den Bosch network, the output delay of WIC+BB is similar to the input delay. Since in this network there is little opportunity to recover input delays, this performance of WIC+BB is quite good. The performance of GW+BB is limited by the speed differences among trains and by the lack of routing flexibility, so that trains are forced to wait at their platforms until the whole corridor becomes available, and subsequent trains cannot enter the station. It is likely that significant improvements might be achieved by allowing trains to be rerouted dynamically.

From this discussion, the green wave policy is expected to be most effective when dealing with short corridors, small differences in train speeds and spare capacity available at the stations.

6.4 Conclusions and Future Research

This paper discusses the concept of green wave policy in the management of complex railway networks. Two detailed models are introduced to formulate the CDR problem with the GW policy and with an alternative policy of letting trains change their running times along open corridors. An assessment of the performance of the two policies is made by using the three CDR algorithms and for two practical dispatching areas of the Dutch railway network under different traffic conditions.

From the computational experiments, it turns out that the GW policy is an effective policy to reduce both train delays and energy consumption, without requiring sophisticated speed optimization methods. However, in order to use this policy in real operations, advanced decision support systems for CDR are necessary. Further algorithmic development is necessary to develop such systems since the algorithms described in this paper still have some limits for real-time purposes. Simple heuristics may fail in finding feasible solutions while the branch and bound algorithm is still quite time consuming. A number of other issues remain that need further development.

- As mentioned in the introduction of the paper, the quantitative benefit of a policy depends on the combined effects of a number of factors. More extensive research is therefore necessary to define general criteria to choose an appropriate policy for a given railway network.
- The GW policy makes extensive use of dwell time extensions. Therefore, rerouting

policies and flexible departure times at scheduled stops in station areas are potentially promising for delay reduction purposes in combination with the GW policy and for congested railway networks.

- Advanced speed optimization algorithms with the WIC policy might generate solutions of comparable or even better quality, even though such methods would require the addition of new technical equipments to the existing infrastructure.
- It remains to be solved the challenging problem of coordinating adjacent dispatching areas, which must take into account the definition of the entrance time of each train in each area.
- It would be interesting to include other railway constraints in the model, such as different signaling technologies, inter-train connections and others. However, additional constraints increase the propagation of train delays, thus making the effect of the constraints imposed by the signaling system potentially more disruptive. In general, adding new constraints to the alternative graph formulation is not difficult, but the effects of such constraints still have to be evaluated in terms of solution quality and computational effort.
- The potential of the alternative graph formulation has not been fully explored. It may also be useful to solve other problems of traffic control and capacity research.

6.5 Acknowledgements

This work has been partially funded by the Dutch government, program TRANSUMO "Reliability of transport chains", by the Dutch foundation "Next Generation Infrastructures", and by the Italian Ministry of Research, Grant number RBIP06BZW8, project FIRB "Advanced tracking system in intermodal freight transportation". The authors are also grateful to ProRail, the Dutch rail infrastructure manager, that provided the instances and to the anonymous referees for their helpful comments.

Bibliography

[1] T. Albrecht. The influence of anticipating train driving on the dispatching process in railway conflict situations. *Networks and Spatial Economics*, 9(1):85–101, 2009.

[2] M. Asuka and K. Komaya. Automatic train operation using braking pattern transitional time. In *Proceedings of the World Congress on Railway Research*, pages 628–636. Edinburgh, UK, 2003.

[3] N. Berends and N. Ouburg. Beschrijving ARI-functionaliteit. Technical Report 20, ProRail Internal Specification (in Dutch), Utrecht, The Netherlands, 2005.

[4] J. F. Cordeau, P. Toth, and D. Vigo. A Survey of Optimization Models for Train Routing and Scheduling. *Transportation Science*, 32 (4):380–404, 1998.

[5] A. D'Ariano and T. Albrecht. Running time re-optimization during real-time timetable perturbations. In J. Allan, C. A. Brebbia, A. F. Rumsey, G. Sciutto, S. Sone, and C. J. Goodman, editors, *Computers in Railways X*, pages 531–540. WIT Press, Southampton, UK, 2006.

[6] A. D'Ariano, D. Pacciarelli, and M. Pranzo. A branch and bound algorithm for scheduling trains in a railway network. *European Journal of Operational Research*, 183(2):643–657, 2007.

[7] A. D'Ariano, M. Pranzo, and I. A. Hansen. Conflict resolution and train speed co-ordination for solving real-time timetable perturbations. *IEEE Transactions on Intelligent Transportation Systems*, 8(2):208–222, 2007.

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

[9] O. Happel. Sperrzeiten als grundlage fr die fahrplankonstruktion. *Eisenbahntechnische Rundschau (ETR)*, 8:79–90, 1959.

[10] R. Hemelrijk, J. Kruijer, and D. K. De Vries. Schiphol tunnel 2007. Description of the situation. Technical Report 22, Holland Railconsult, Utrecht, 2003.

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

[12] A. Mascis, D. Pacciarelli, and M. Pranzo. Scheduling models for short-term railway traffic optimisation. In Hickman, Mirchandani, and Voss, editors, *Computer-aided Systems in Public Transport, Lecture Notes in Economics and Mathematical Systems 600*, pages 71–90. Springer, 2008.

[13] M. Mazzarello and E. Ottaviani. A traffic management system for real-time traffic optimisation in railways. *Transportation Research, Part B*, 41 (2):246–274, 2007.

[14] D. Pacciarelli and M. Pranzo. Speed regulation in rail networks. In H. J. Van Zuylen and F. Middelham, editors, *Proceedings of the 11th IFAC Symposium on Control in Transportation Systems*. Delft, the Netherlands, 2006.

[15] J. Pachl. *Railway Operation and Control*. VTD Rail Publishing, Mountlake Terrace, 2002.

[16] B. Szpigel. Optimal Train Scheduling on a Single Track Railway. In M. Ross, editor, *Operational Research '72*, pages 343–352, Amsterdam, the Netherlands, 1973.

Summary

Railway is a transportation mode of central importance, especially for commuting in urban areas, for direct, fast and reliable connections between cities, and in general for the very reliable, energy efficient and safe services delivered. Despite those good potentials, the modal split is still much in favour of road traffic, concerning passengers and freight transportation. This is mainly due to limited attractiveness of railway, where the perceived consequences of unreliability play a major role. In fact, due to limited possibility for trains to overtake along open tracks and capacity limitations at stations where multiple lines merge and cross each other, and when dense timetables schedule frequent and heterogeneous services, the railway system is highly sensitive to delay propagation. Primary delays (due e.g. to mechanical failures, extended dwell times, accidents) spread easily in the network as consecutive delays, affecting a large share of passengers.

Advanced methods for careful off-line planning are used, to deliver robust timetables that exploit buffer and recovery times to reduce the impact of small delays during operations. Anyway, when train traffic results in infeasibilities, train conflicts and delay propagation, dispatching actions are required in real-time, in order to solve those problems and return timely to planned operations.

Current dispatching shows a reactive approach, i.e. dispatchers limit their actions to strongly disturbed situations, and receive only limited information on the actual status of traffic; current dispatching actions are mostly based on experience or simple rules, no quantitative assessment in their outcome is provided, and they often result in suboptimal decisions. Much better performance could be achieved if decisions were taken that are proactive and informed, based on detailed information of the traffic status at the current moment and extended for a reasonable period of traffic prediction (e.g. 30 minutes), while advanced mathematical methods could be used to deliver dispatching actions that are optimal at local and global scale.

A significant improvement would be possible when considering enhanced possibilities to manage traffic in real-time, according to the principles of dynamic railway traffic management. In fact, the quality and reliability of train operations can be improved by shifting some operational decisions on train orders, routes, departure times from an off-line plan-

ning point of view - months before actual operations - to real-time control, i.e. taking the actual unpredictable situation of train traffic into account to improve operations' quality. Advanced and comprehensive support for dispatchers is needed to let them understand in full the consequences of particular dispatching actions from a network perspective, and manage optimally the enlarged degree of freedom of operations. Clearly, this support is most required for problems that are difficult to be solved, i.e. those in which the experience of the dispatcher, automatic route setting systems or simple dispatching rule as the First-Come First-Served may result in suboptimal solutions or even in infeasible deadlock situations.

This thesis aims at filling the gaps in real-time railway traffic management, with particular focus on difficult and challenging instances, such as those that have an extended degree of complexity, due to geographical size, traffic frequency and density, amount of traffic, operational policies adopted. Flexible and detailed mathematical methods based on the alternative graph are used to model precisely railway operations at the very detailed level of block sections, signals and switches, in order to deliver a plan that is microscopically feasible according to the proven blocking time theory. Powerful algorithms solve potential conflicts that are expected based on the current situation of train positions, speeds, traffic and infrastructure status, and additional constraints due to the published timetable or connections between services.

The innovative work of D'Ariano has been extended from the point of view of the modelling and algorithms and introducing furthermore innovative approaches to manage large instances, or with additional requirements. More precisely, the main features and contributions of the thesis include: (i) a detailed microscopic model that combines the blocking time theory with the alternative graph model, in order to provide feasible train movements at the level of block sections; and (ii) a comprehensive set of detailed models and advanced algorithms that are able to compute dispatching solutions taking into account the extended degrees of freedom in railway operations proposed for dynamic railway traffic management.

With particular regard to the previous work of D'Ariano, and the current state of the art of research on railway rescheduling, the main contributions comprise:

- enhanced models to describe properly the complexity of train movements in bottleneck interlocking areas of large stations, where many trains are planned to run after each other, crossing and hindering each other over many train routes composed of individual track sections. Aggregation of track sections into blocks and introduction of virtual resources have been used to take into account the precise operational rules and to solve optimally the many potential conflicts in the short time required for real operations.
- an innovative framework to address the exponential growth in instance size brought

about by considering large instances while keeping microscopic detail of train movements. Large dispatching problems are decomposed into a number of mutually interconnected local dispatching problems, each focusing on a smaller area. The latter problems can be solved independently, while a coordination procedure harmonizes their solutions and guides the solvers toward locally feasible and globally optimal solutions.

- the study of the coordination problem between multiple dispatching areas making up a large network. Advanced algorithms are designed and evaluated which find feasible solutions for multiple cases of time horizon, traffic perturbation, infrastructure availability in a short computation time.

- extensions to the available models of the dispatching problem to deliver more attractive solutions, that take into account other interesting objectives or operating policies. In this way particular interest of the stakeholders, like energy-efficient operations, simpler dispatching actions, satisfaction of passenger connections could be considered while optimizing dispatching actions.

Overall, the investigation on comprehensive mathematical models and algorithms underlines the great impact that such advanced procedures may have in handling the difficult dispatching problem in order to deliver a better and more attractive railway service to passengers, and increasing the attractiveness of the railway system with benefits for the whole society. We briefly recall the individual content of the articles here collected.

Article I presents a general introduction to the dispatching process, including requirements, models and procedures. The recent concept of dynamic railway traffic management is discussed that prescribes less details about train orders, routes, passing times fixed during the timetabling and planning stage, and leaves more freedom for the dispatcher to adapt the plan to the actual situation, reducing the impact of delays. Advanced simulation and optimization models resulting in the dispatching system ROMA are used that take into account the complexity and the detail of railway operations at a microscopic level. By doing so, the main strategies of dynamic traffic management are evaluated extensively, leading to the conclusion that the extended degree of freedom available during real-time dispatching leads to significant benefits in operations' quality when complemented with advanced algorithms able to tackle the inherent complexity of the dispatching problem to deliver optimal solutions.

Article II reports on the extensions to the model and algorithms needed to control dense traffic in complex and large stations, that are too difficult for dispatchers to be solved effectively. Moreover, station areas represent the bottlenecks of railway operations, due to the many incompatible train routes crossing each other that lead to many potential conflicts between trains; therefore, microscopic detail and advanced optimization is required to properly manage traffic. Station routes defined by a set of track sections are aggregated

into blocks to model precisely real operations, simplify the model when possible, and speed up the computation process while retaining the microscopic data needed to detect conflicts between train paths. A practical focus is kept in the choice of a large real-life test case and simulating perturbation instances derived from realization data. Applicability implications are considered by studying the setup of a decision support system for operations, including relevant features of an interface to the dispatcher. The advanced algorithms are assessed against simple dispatching rules and less performing automated route setting systems, to prove the potential of advanced algorithms.

Article III reports a significant step towards handling large and complex instances, where much improvement can be achieved when taking fully into account the global effects of local choices. To do so, a geographical decomposition of large railway networks into several adjacent and mutually interconnected areas is considered in the optimization model. A framework is designed and implemented that is able to manage concurrent resolution of multiple dispatching areas, supporting exchange of relevant information between the various processes, and guiding the solution towards locally feasible and globally optimal dispatching actions. The distributed approach proves feasible and comparable in solution quality with an advanced centralized approach, and moreover delivers better solutions when heavy disruptions to operations are considered.

Article IV compares extensively centralized and distributed approaches by extending the distributed approach to larger areas and multiple decompositions. Those instances require to address the interesting problem of coordinating multiple dispatching areas to deliver a global solution at network scale, in a similar way to real-life interaction between different dispatchers and network traffic controllers. The influence of multiple factors that have a strong influence on the complexity and the performance of the dispatching process is analysed considering the time horizon of traffic prediction, multiple solution algorithms, different infrastructure availability scenarios and perturbed operations with multiple trains delayed.

Article V reports on a different extension of the dispatching support system, in order to consider multiple operating policies and different objectives into the optimization process, to include characteristic goals of the multiple stakeholders of railway operations. In this way, more attractive solutions could be computed that optimize dispatching actions while at the same time improving performance according to other indicators. As an example, the benefits of a green wave speed management policy when dispatching trains are assessed, considering the impact of network size, traffic density, scheduling complexity and station capacity in the outcome of the dispatching system. With a green wave policy, trains run unhindered between two stations, thus resulting in energy efficient operations, while the suggested dispatching actions are simplified and consider only dwell time extensions. On the other hand, capacity at stations becomes critical, and the overall problem is computationally more difficult to solve. The results prove that flexible and general models as

those included in ROMA can be extended to deliver solutions that take into account more comprehensive objective functions, as far as advanced algorithms are available.

To summarize, this thesis provides innovative contributions on the application of advanced mathematical methods, algorithmic enhancements, novel approaches to solve the dispatching problem, resulting in a dispatching decision support system, working in a laboratory environment. The ideas proposed are evaluated based on real-life instances to assess quantitatively the benefits of real-time railway traffic management.

Samenvatting

Spoorvervoer is van essentieel belang voor met name forensen in stedelijke gebieden, voor snelle en betrouwbare verbindingen tussen steden en in het algemeen voor zeer betrouwbare, energiezuinige en veilige vervoersdiensten. Ondanks deze goede potenties is het aandeel wegverkeer nog altijd groter, zowel in personen- als vrachtvervoer. Dit is vooral te wijten aan de beperkte aantrekkelijkheid van spoorvervoer, waarin de ogenschijnlijke consequenties van onbetrouwbaarheid een aanzienlijke rol spelen. Het railverkeerssysteem is zeer gevoelig voor de voortplanting van vertragingen vanwege de beperkte mogelijkheden om treinen te laten inhalen op de vrije baan, de capaciteitsbeperkingen rondom stations waar meerdere lijnen samenkomen en elkaar kruisen, en de intensieve dienstregeling met frequente en heterogene diensten. Primaire vertragingen (bijvoorbeeld door storingen, langere verblijftijden en ongevallen) verspreiden zich gemakkelijk door het netwerk als secundaire vertragingen en hebben zodanig invloed op een grote hoeveelheid reizigers.

Geavanceerde methoden worden gebruikt voor een nauwkeurige off-line planning, resulterend in robuuste dienstregelingen die gebruik maken van buffertijden en spelingen om de invloed van kleine vertragingen tijdens operatie te reduceren. Desondanks, als het spoorverkeer leidt tot trein- en dienstregelingsconflicten en het verspreiden van vertragingen, zijn in real-time bijstuurmaatregelen vereist om deze problemen op te lossen en snel terug te keren naar de geplande situatie.

De huidige verkeersleiding gebeurt reactief, d.w.z. railverkeersleiders grijpen enkel in wanneer de situatie ernstig is verstoord en ontvangen slechts beperkte informatie over de actuele status van het spoorverkeer. De huidige verkeersleiding is vooral gebaseerd op ervaring of simpele richtlijnen en een kwantitatieve beoordeling van het resultaat ontbreekt, waardoor het vaak leidt tot suboptimale beslissingen. Een veel betere prestatie kan geleverd worden wanneer besluiten proactief en onderbouwd genomen worden, gebaseerd op gedetailleerde informatie over de actuele verkeersstatus en aangevuld met een verkeersvoorspelling over een redelijke periode (bijvoorbeeld 30 minuten), terwijl geavanceerde mathematische methodieken benut kunnen worden om regelacties te bepalen die zowel op lokaal als globaal niveau optimaal zijn.

Een significante verbetering is te verwachten indien meer mogelijkheden worden benut

om het verkeer in real-time te managen uitgaande van de principes van dynamisch railverkeersmanagement. De kwaliteit en betrouwbaarheid van de dienstuitvoering kan worden verbeterd door bepaalde operationele besluiten aangaande treinvolgordes, routes en vertrektijden te verschuiven van off-line planning maanden voorafgaand aan de werkelijke uitvoering naar real-time control, daarbij rekening houdend met de onvoorspelbaarheid van de verkeerssituatie. Geavanceerde en uitgebreide ondersteuning van verkeersleiders is nodig om enerzijds volledig te begrijpen wat de netwerkwijde consequenties zijn van specifieke maatregelen, en anderzijds om te gaan met de toename aan vrijheidsgraden op operationeel niveau. Het moge duidelijk zijn dat deze ondersteuning het meest nodig is bij problemen die moeilijk op te lossen zijn, d.w.z. bij situaties waarin afgaand op ervaring van de verkeersleider systemen voor automatische rijweginstelling en simpele maatregelen zoals First-Come First-Served kunnen resulteren in suboptimale oplossingen of zelfs onmogelijke deadlock situaties.

Dit proefschrift richt zich op het verhelpen van de lacunes in real-time railverkeersmanagement, waarbij de focus ligt op ingewikkelde en moeilijk oplosbare situaties door een grote mate van complexiteit, geografische omvang, verkeersdichtheid en frequentie, en van toepassing zijnd operationeel beleid. Flexibele en gedetailleerde mathematische methodieken gebaseerd op *alternative graphs* zijn gebruikt om op nauwkeurige wijze de railverkeersafwikkeling op gedetailleerd niveau van bloksecties, seinen en wissels te modelleren en hiermee een plan op te stellen dat op microscopisch niveau haalbaar is volgens de bewezen bloktijd theorie. Krachtige algoritmen lossen potentiële conflicten op die verwacht worden op basis van de actuele situatie aangaande treinposities en snelheden, status van verkeer en infrastructuur en additionele beperkingen komend van de gepubliceerde dienstregeling of aansluitingen tussen diensten.

Het innovatieve werk van D'Ariano is aangevuld betreffende modellering en algoritmes en de introductie van innovatieve aanpakken om grootschalige situaties en situaties met additionele eisen te managen. Om precies te zijn, omvatten de voornaamste karakteristieken en bijdragen van dit proefschrift: (i) een gedetailleerd microscopisch model dat bloktijdtheorie combineert met het *alternative graph* model om mogelijke treinbewegingen op het niveau van bloksecties te bepalen en (ii) een uitgebreide verzameling aan gedetailleerde modellen en geavanceerde algoritmes die het mogelijk maken om regeloplossingen te bepalen rekening houdend met het grote aantal vrijheidsgraden in de verkeersafwikkeling binnen dynamisch railverkeersmanagement.

Met betrekking tot het voorgaande werk van D'Ariano en de huidige stand van de techniek op het gebied van dynamisch railverkeersmanagement omvatten de grootste bijdragen:

- verbeterde modellen om de complexiteit van treinverplaatsingen op de emplacementen van grote stations te beschrijven waar vele treinen achtereenvolgens gepland zijn en vele rijwegen elkaar kruisen en conflicteren op individuele spoorsec-

ties. Aggregatie van spoorsecties in blokken en introductie van virtuele elementen zijn toegepast om specifieke operationele regels te waarborgen en tegelijkertijd potentiële conflicten op te lossen binnen de korte tijd die beschikbaar is voor de werkelijke dienstuitvoering.

- een innovatief raamwerk om te kunnen omgaan met de exponentiële groei van de probleemomvang die ontstaat door het in beschouwing nemen van grootschalige netwerken en handhaving van een microscopisch detail van treinverplaatsingen. Grotere verkeersleidingsproblemen worden ontleed in een aantal onderling samenhangende lokale regelproblemen, elk met een focus op een kleiner gebied. Deze laatste kunnen onafhankelijk van elkaar opgelost worden terwijl een coördinerende procedure de afzonderlijke oplossingen harmoniseert en aanstuurt op een lokaal realiseerbare en globaal optimale oplossing.

- studie naar de coördinatie van meerdere treindienstleidingsgebieden in een groter netwerk. Geavanceerde algoritmes zijn ontworpen en geëvalueerd die in staat zijn om binnen een korte rekentijd mogelijke oplossingen te vinden voor diverse situaties betreffende tijdshorizon, vertragingen en beschikbaarheid van infrastructuur.

- uitbreidingen op bestaande modellen voor railverkeersmanagement voor het bepalen van betere oplossingen die rekening houden met andere doelen of operationeel beleid. Op deze wijze kan tijdens het optimalisatie proces tegemoet worden gekomen aan specifieke wensen van belanghebbenden, zoals energiezuinig rijden, eenvoudige regelacties en tevredenheid over reizigersaansluitingen.

Het onderzoek in zijn geheel naar omvangrijke mathematische modellen en algoritmes onderschrijft de grote potentie van dergelijke geavanceerde procedures bij het omgaan met moeilijk oplosbare verkeersleidingsproblemen om te komen tot betere en aantrekkelijkere spoordienstregelingen voor reizigers en om de aantrekkelijkheid van het spoorwegsysteem te verbeteren ten goede van de gehele gemeenschap. We recapituleren kort de inhoud en bijdragen van de afzonderlijke artikelen die hier gebundeld zijn.

Artikel I presenteert een algemene introductie in het verkeersleidingsproces, waaronder eisen, modellen en methodieken. Het recente concept van dynamisch railsverkeermanagement wordt behandeld waarbij minder gedetaileerd treinvolgordes, rijwegen en passeertijden worden vastgesteld tijdens het plannen en en in de dienstregeling, en meer vrijheid wordt gelaten aan de verkeersleider om het plan aan te passen aan de actuele situatie, daarbij het effect van vertragingen reducerend. Geavanceerde simulatie- en optimaliseringsmodellen in het railverkeersmanagementsysteem ROMA worden gebruikt die rekening houden met de complexiteit en detail van de railverkeersafwikkeling op microscopisch niveau. Hierbij worden de voornaamste strategieën in dynamisch verkeersmanagement uitgebreid geëvalueerd met als conclusie dat de toename in vrijheidsgraden in real-time

verkeersmanagement leidt tot significante voordelen in de prestaties van de dienstuitvoering wanneer deze aangevuld wordt met geavanceerde algoritmes die in staat zijn om optimale oplossingen aan te dragen binnen de inherente complexiteit van het verkeersleidingsprobleem.

Artikel II rapporteert over de uitbreidingen van de modellen en algoritmes die nodig zijn voor het managen van druk spoorverkeer bij grote en complexe stations, dat voor treindeinstleiders te moeilijk is om effectief op te lossen. Bovendien representeren stationsgebieden de knelpunten van de dienstuitvoering gegeven de vele conflicterende rijwegen die elkaar kruisen en leiden tot vele potentiële conflicten tussen treinen. Daarom is microscopisch detail en geavanceerde optimalisatie nodig bij het adequaat managen van het spoorverkeer. Rijwegen over emplacementen, gedefinieerd als een aaneenschakeling van spoorsecties, worden gebundeld tot blokken om op nauwkeurige wijze het daadwerkelijke infrastructuurgebruik te modelleren, het model waar nodig te simplificeren, en het rekenproces te versnellen, terwijl de microscopische data wordt behouden om conflicterende rijwegen te detecteren. Een praktische focus wordt aangehouden in de keuze voor een grootschalige real-life test casus en het simuleren van verstoringsscenarios genaseerd op realisatiegegevens. Implicaties voor de toepasbaarheid worden bekeken door het bestuderen van de structuur van een beslissingsondersteunend systeem voor de dienstuitvoering, inclusief relevante karakteristieken van een interface voor de verkeersleider. De geavanceerde algoritmes worden geëvalueerd in relatie tot simpele afhandelingsstrategieën en minder presterende geautomatiseerde systemen voor rijweginstellingen om daarmee de potentie van geavanceerde algoritmes aan te tonen.

Artikel III rapporteert een significante stap naar het omgaan met grootschalige en complexe situaties waar veel winst behaald kan worden wanneer de globale effecten van lokale beslissingen volledig in overweging worden genomen. Hiervoor wordt een geografische decompositie van grootschalige spoornetwerken in meerdere aangrenzende en onderling samenhangende deelgebieden opgenomen in het optimaliseringsmodel. Een raamwerk is ontwikkeld en gemplementeerd waarin het mogelijk is om de oplossingen van meerdere treindienstleidingsgebieden gelijktijdig te managen, onderlinge uitwisseling van relevante informatie te ondersteunen, en de oplossing te leiden naar lokaal realiseerbare en globaal optimale maatregelen. De gedistribueerde aanpak blijkt haalbaar en levert resultaten die vergelijkbaar zijn met een geavanceerde centrale aanpak. Daarnaast worden betere resultaten verkregen bij zware verstoringen.

Artikel IV maakt een uitgebreide vergelijking tussen gecentraliseerde en gedistribueerde aanpakken door de gedistribueerde methode toe te passen op grotere gebieden en meerdere decomposities. Deze situaties vereisen de aanpak van het interessante probleem van het coördineren van meerdere treindienstleidingsgebieden om te komen tot een globale oplossing op netwerkniveau, vergelijkbaar met de interactie in werkelijkheid tussen verschillende treindienstleiders en netwerkverkeersleiders. De invloed van meerdere fac-

toren die een sterke invloed uitoefenen op de complexiteit en de prestatie van het verkeersleidingsproces wordt geanalyseerd, waarbij gelet wordt op de tijdshorizon voor verkeersvoorspelling, meerdere oplossingsalgoritmen, verschillende scenarios voor infrastructuurbeschikbaarheid en verstoorde dienstuitvoeringen met meerdere vertraagde treinen.

Artikel V presenteert een andere uitbreiding op het treindienstleidingsondersteunend systeem die rekening houdt met verschillende vormen van operationeel beleid en verschillende doelen in het optimaliseringsproces, zodat karakteristieke doelen van de verschillende belanghebbenden in het spoorvervoer worden meegenomen. Op deze wijze kunnen betere oplossingen worden bepaald die bijsturingsmaatregelen optimaliseren en die tegelijkertijd een verbetering opleveren ten opzichte van andere indicatoren. Als voorbeeld wordt het voordeel van een groene golf snelheidsmanagementsstrategie op de resultaten van een railverkeersmanagementsysteem geëvalueerdmet betrekking op het effect van netwerkgrootte, verkeersdichtheid, planningscomplexiteit en stationscapaciteit. Met een groene golf strategie rijden treinen ongehinderd tussen stations wat resulteert in energiezuinig rijden terwijl de voorgestelde regelacties eenvoudiger worden en alleen verlengde stoptijden op stations in overweging worden genomen. Anderzijds wordt de capaciteit bij stations kritiek en daarmee het gehele vraagstuk moeilijker om op te lossen. De resultaten tonen aan dat flexibele en generieke modellen zoals opgenomen in ROMA kunnen worden uitgebreid om oplossingen te vinden voor meer omvattende doelfuncties, voor zover geavanceerde algoritmen beschikbaar zijn.

Samenvattend voorziet dit proefschrift in innovatieve bijdragen voor het toepassen van geavanceerde mathematische methodieken, algoritmische uitbreidingen en nieuwe aanpakken om verkeersleidingsproblemen op te lossen, resulterend in een prototype beslissingsondersteunend systeem voor verkeersleiding. De voorgestelde ideeën zijn gevalueerd op basis van real-life situaties om de voordelen van real-time railverkeersmanagement te kwantificeren.

About the Author

Francesco Corman was born 1982 in Rieti, 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 Real-time railway rescheduling and rerouting, was conducted under the supervision of prof. D. Pacciarelli (Roma Tre), supervised by Andrea D'Ariano and Ingo Hansen (Delft University of Technology), and supported by a University Scholarship and ProRail. in 2006, he was awarded the Galluzzi Prize for best Engineering thesis, the Accenture Thesis Prize and the AIRO prize for the best thesis in operational research. In 2007, after two months spent as researcher at the operations research group of prof. D. Pacciarelli, he joined as a Ph.D the Department of Transport and Planning, Faculty of Civil Engineering and Geosciences, Delft University of Technology. At the beginning of 2007, he was awarded the Professional Italian Title of Engineer. He is a member of TRAIL Research School.

His research publications were acknowledged by a first place at the INFORMS Railway Applications Section Annual Meeting, and a selection among the best papers at International Association of Railway Operations Research Seminar. In addition to the works presented in 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, among which Transportation Research Part C. His general research interests include operations research and its application to real-life problems, like logistics and transportation.

TRAIL Thesis Series

The following list contains the most recent dissertations in the TRAIL series. See for a complete overview of more than 100 titles the TRAIL website: www.rsTRAIL.nl.
A series of The Netherlands TRAIL Research School for transport, infrastructure and logistics.

Corman, F., *Real-Time Railway Traffic Management: dispatching in complex, large and busy railway networks*, T2010/14, December 2010, TRAIL Thesis Series, the Netherlands

Kwakkel, J., *The Treatment of Uncertainty in Airport Strategic Planning*, T2010/13, December 2010, TRAIL Thesis Series, the Netherlands

Pang, Y., *Intelligent Belt Conveyor Monitoring and Control*, T2010/12, December 2010, TRAIL Thesis Series, the Netherlands

Kim, N.S., *Intermodal Freight Transport on the Right Track? Environmental and economic performances and their trade-off*, T2010/11, December 2010, TRAIL Thesis Series, the Netherlands

Snelder, M., *Designing Robust Road Networks: a general design method applied to the Netherlands*, T2010/10, December 2010, TRAIL Thesis Series, the Netherlands

Hinsbergen, C.P.IJ. van, *Bayesian Data Assimilation for Improved Modeling of Road Traffic*, T2010/9, November 2010, TRAIL Thesis Series, the Netherlands

Zuurbier, F.S., *Intelligent Route Guidance*, T2010/8, November 2010, TRAIL Thesis Series, the Netherlands

Larco Martinelli, J.A., *Incorporating Worker-Specific Factors in Operations Management Models*, T2010/7, November 2010, TRAIL Thesis Series, the Netherlands

Ham, J.C. van, *Zeehavenontwikkeling in Nederland: naar een beter beleidsvormingsproces*, T2010/6, August 2010, TRAIL Thesis Series, the Netherlands

Boer, E. de, *School Concentration and School Travel*, T2010/5, June 2010, TRAIL Thesis Series, the Netherlands

Berg, M. van den, *Integrated Control of Mixed Traffic Networks using Model Predictive Control*, T2010/4, April 2010, TRAIL Thesis Series, the Netherlands

Top, J. van den, *Modelling Risk Control Measures in Railways*, T2010/3, April 2010, TRAIL Thesis Series, the Netherlands

Craen, S. de, *The X-factor: A longitudinal study of calibration in young novice drivers*, T2010/2, March 2010, TRAIL Thesis Series, the Netherlands

Tarau, A.N., *Model-based Control for Postal Automation and Baggage Handling*, T2010/1, January 2010, TRAIL Thesis Series, the Netherlands

Knoop, V.L., *Road Incidents and Network Dynamics: Effects on driving behaviour and traffic congestion*, T2009/13, December 2009, TRAIL Thesis Series, the Netherlands

Baskar, L.D., *Traffic Control and Management with Intelligent Vehicle Highway Systems*, T2009/12, November 2009, TRAIL Thesis Series, the Netherlands

Konings, J.W., *Intermodal Barge Transport: Network Design, Nodes and Competitiveness*, T2009/11, November 2009, TRAIL Thesis Series, the Netherlands

Kusumaningtyas, I., *Mind Your Step: Exploring aspects in the application of long accelerating moving walkways*, T2009/10, October 2009, TRAIL Thesis Series, the Netherlands

Gong, Y., *Stochastic Modelling and Analysis of Warehouse Operations, T2009/9*, September 2009, TRAIL Thesis Series, the Netherlands

Eddia, S., *Transport Policy Implementation and Outcomes: the Case of Yaounde in the 1990s*, T2009/8, September 2009, TRAIL Thesis Series, the Netherlands

Platz, T.E., *The Efficient Integration of Inland Shipping into Continental Intermodal Transport Chains: Measures and decisive factors*, T2009/7, August 2009, TRAIL Thesis Series, the Netherlands

Tahmasseby, S., *Reliability in Urban Public Transport Network Assessment and Design*, T2009/6, June 2009, TRAIL Thesis Series, the Netherlands

Bogers, E.A.I., *Traffic Information and Learning in Day-to-day Route Choice*, T2009/5, June 2009, TRAIL Thesis Series, the Netherlands

Amelsfort, D.H. van, *Behavioural Responses and Network Effects of Time-varying Road Pricing*, T2009/4, May 2009, TRAIL Thesis Series, the Netherlands

Li, H., *Reliability-based Dynamic Network Design with Stochastic Networks*, T2009/3, May 2009, TRAIL Thesis Series, the Netherlands

Stankova, K., *On Stackelberg and Inverse Stackelberg Games & their Applications in the Optimal Toll Design Problem, the Energy Markets Liberalization Problem, and in the Theory of Incentives*, T2009/2, February 2009, TRAIL Thesis Series, the Netherlands

Li, T., *Informedness and Customer-Centric Revenue*, T2009/1, January 2009, TRAIL Thesis Series, the Netherlands

Agusdinata, D.B., *Exploratory Modeling and Analysis: A promising method to deal with deep uncertainty*, T2008/17, December 2008, TRAIL Thesis Series, the Netherlands

Kreutzberger, E., *The Innovation of Intermodal Rail Freight Bundling Networks in Europe: Concepts, Developments, Performances*, T2008/16, December 2008, TRAIL Thesis Series, the Netherlands

Taale, H., *Integrated Anticipatory Control of Road Networks: A game theoretical approach*, T2008/15, December 2008, TRAIL Thesis Series, the Netherlands

Li, M., *Robustness Analysis for Road Networks: A framework with combined DTA models*, T2008/14, December 2008, TRAIL Thesis Series, the Netherlands

Yu, M., *Enhancing Warehouse Performance by Efficient Order Picking*, T2008/13, October 2008, TRAIL Thesis Series, the Netherlands

Liu, H., *Travel Time Prediction for Urban Networks*, T2008/12, October 2008, TRAIL Thesis Series, the Netherlands

Kaa, E.J. van de, *Extended Prospect Theory: Findings on Choice Behaviour from Economics and the Behavioural Sciences and their Relevance for Travel Behaviour*, T2008/11, October 2008, TRAIL Thesis Series, the Netherlands

Nijland, H., *Theory and Practice of the Assessment and Valuation of Noise from Roads and Railroads in Europe*, T2008/10, September 2008, TRAIL Thesis Series, the Netherlands

Annema, J.A., *The Practice of Forward-Looking Transport Policy Assessment Studies*, T2008/9, September 2008, TRAIL Thesis Series, the Netherlands

Ossen, S.J.L., *Theory and Empirics of Longitudinal Driving Behavior*, T2008/8, September 2008, TRAIL Thesis Series, the Netherlands

Tu, H., *Monitoring Travel Time Reliability on Freeways*, T2008/7, April 2008, TRAIL Thesis Series, the Netherlands