DISS. ETH NO. 25111

AUTOMATED RAILWAY TRAFFIC RESCHEDULING AND CUSTOMER INFORMATION

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH

(Dr. sc. ETH Zurich)

presented by

AMBRA TOLETTI

MSc. ETH Appl. Math

born on 07.06.1990

citizen of
Monteggio (TI), Switzerland

accepted on the recommendation of

Prof. Dr. Ulrich A. Weidmann, examiner
Prof. Dr. Anita Schöbel, co-examiner
Dr. Marco Laumanns, co-examiner
Dr. Andreas Dietzsch, co-examiner

2018

Abstract

This thesis focuses on algorithms to support real-time railway traffic management. The thesis is situated in the research context of *dynamic capacity improvement*, which aims at increasing railway capacity using the currently existing reserves in the system to schedule new services without construction of new infrastructures. The key idea of dynamic capacity improvement is to increase planning and operating precision through automated systems, such that time reserves at capacity bottlenecks can be reduced or even eliminated. The algorithms proposed in this work are integrable into the current information flows of traffic management to enable automated traffic rescheduling and, as a consequence, more timely and accurate information to customers.

Previous works in this research context have developed algorithms to support timetabling, rescheduling and speed control. In particular, the work by Fuchsberger (2012) set the basis for the present thesis. He developed a rescheduling algorithm for major station areas with complex topology and very dense traffic. The algorithm was based on the *resource conflict graph*, which is a mathematical formulation of the railway traffic rescheduling problem enabling retiming, reordering, rerouting, reservicing and speed advisory decisions. The present thesis aims at further developing that algorithm and makes four main contributions to this end:

- an extension of the resource conflict graph model that enables decisions on several operation points to be modelled explicitly, which makes the formulation more suitable for rescheduling in simple topologies;

- a simulation-based approach, based on the *green wave* principle, to generate the blocking time stairways linked to the decision variables of the resource conflict graph formulation, which enables different train routings and speed profiles to be modelled precisely;

- a column generation framework that considerably reduces the computational effort to solve the resource conflict graph formulation;

- a non-hierarchical coordination approach for rescheduling in adjacent zones of the railway network that enables large train rescheduling problems to be decomposed according to the available compute server.

The validity and the effectiveness of these methods have been successfully tested using simulations of railway operations. Two very different real-world railway networks from Switzerland have been used to show that the proposed methodology is applicable to a large variety of situations:

- The double track section between Berne and Worblaufen managed by the *Regionalverkehr Bern Solothurn (RBS)*, which has a simple topology but extremely dense traffic;

- The area around station Brugg managed by the *Swiss Federal Railways (SBB)*, which presents both complex and simple topology sections and very heterogeneous traffic.

The tests showed experimentally that the column generation framework for the extended resource conflict graph formulation can solve train rescheduling problems within the short times available during real-time traffic management and the new adapted schedules are feasible, i.e. the model extension is valid. The new adapted schedules significantly reduce conflicts with respect to a first come first served policy, which improves the traffic flow and potentially reduces the overall energy consumption. However, the tests have also highlighted that these improvements are associated with an increase of the overall train delays (with respect to first come first served), as the new schedules include larger headways and possibly some additional stops at operation points. This indicates a conflict between the objective of dynamic capacity improvement and the green wave principle, which has been used to generate the blocking time stairways linked to the decision variables. Thus, there is the need for further development of the simulation-based blocking time stairways generator in future research. Finally, the tests showed experimentally that the non-hierarchical coordination approach improves the consistency of decisions at portals between zones, and, given that the local rescheduling problems are much less complex than the centralised one, it has computation times compatible with real-time railway operations.

Furthermore, the integration of the algorithms proposed in this thesis into the control loops at SBB has been investigated, and it has been shown that linking them with the current information flows of the traffic management and customer information systems is straightforward. Thus, it can be concluded that these algorithms can actually be exploited to

support real-time railway operations and, as a consequence, to improve timeliness and consistency of customer information.

Zusammenfassung

Diese Dissertation befasst sich mit Algorithmen zur Dispositionsunterstützung des Eisenbahnbetriebs. Sie gliedert sich in den breiten Forschungskontext der *dynamischen Kapazitätssteigerung* ein, die durch Nutzung der bestehenden Zeitreserven im Bahnsystem zusätzliche Trassen zur Verfügung zu stellen sucht. Die Grundidee der dynamischen Kapazitätssteigerung besteht darin, eine höhere Planungs- und Betriebspräzision zu erreichen, so dass die Zeitreserven an den Kapazitätsengpässen verringert oder ganz eliminiert werden können. Die Algorithmen aus dieser Arbeit sollen in die aktuellen Informationsflüsse der Eisenbahnbetriebssteuerung einbindbar sein, um die Automatisierung der Steuerung zu ermöglichen und damit die Konsistenz und Aktualität der Kundeninformation zu verbessern.

In früheren Arbeiten in diesem Kontext wurden Unterstützungsalgorithmen für Fahrplanung, Disposition und Geschwindigkeitsanpassung entwickelt. Insbesondere hat die Arbeit von Fuchsberger (2012), in welcher er einen Dispositionsalgorithmus für hochkomplexe Bahnhofsbereiche mit hochverdichtetem Verkehr entwickelte, die Grundlagen für diese Dissertation gelegt. Die Basis des Algorithmus ist ein gemischt-ganzzahliges lineares Modell des Dispositionsprozesses, ein sogenanntes *Resource Conflict Graph* Modell, welches Entscheidungen zu Laufwegen, Zeitanpassungen, Anschlüssen und Geschwindigkeitsvorgaben ermöglicht. Diese Dissertation erweitert das bestehende Wissen um die folgenden vier Elemente:

- Verallgemeinerung der Formulierung des *Resource Conflict Graph* Modells, um Entscheidungen an mehreren Betriebspunkten explizit zu modellieren, so dass sie für die Disposition in einfachen Gleistopologien geeignet ist.

- Simulationsbasiertes Verfahren für die Erzeugung der Sperrzeitentreppen, die mit den Entscheidungsvariablen des *Resource Conflict Graph* Modells übereinstimmen, so dass verschiedene Laufwege und Geschwindigkeitsprofile präzis modelliert werden können. Die

erzeugten Sperrzeitentreppen sind konform zum Prinzip der grünen Welle.

- *Column-Generation*-Verfahren, welches das Lösen des Modells beschleunigt, so dass die Rechenzeit für einen neuen Fahrplan die kurze, in der Eisenbahnbetriebssteuerung verfügbare Zeit nicht überschreitet.

- Ein nicht-hierarchisches Koordinationsverfahren für benachbarte Dispositionszonen, welches die Verteilung von grossen Dispositionsproblemen, die nicht zentralisiert lösbar sind, ermöglicht.

Diese Methoden wurden mittels Eisenbahnbetriebssimulationen validiert. Um die Aussagekraft der Experimente zu optimieren wurden zwei unterschiedliche Eisenbahnnetze betrachtet:

- Die vom Regionalverkehr Bern-Solothurn (RBS) betriebene zweispurige Strecke zwischen Bern und Worblaufen mit einer einfachen Gleistopologie aber sehr dichtem Verkehr.

- Die Region um Brugg mit heterogenem Verkehr und komplexer Gleistopologie, welche von der Schweizerischen Bundesbahn (SBB) betrieben wird.

Die Validität der Modellerweiterung wurde bestätigt. Das *Column-Generation*-Verfahren für das erweiterte *Resource Conflict Graph* Modell kann das Dispositionsproblem innerhalb der verfügbaren Zeit lösen. Die Anzahl Konflikte werden im Vergleich zu einer First-Come-First-Served Strategie verringert, so dass die Verkehrsflüsse verbessert werden. Die Experimente haben jedoch auch gezeigt, dass diese Verbesserungen im Vergleich zur First-Come-First-Served-Strategie mit grösseren Zugsverspätungen verbunden sind, weil die neu generierten Fahrpläne längere Zugfolgezeiten vorsehen. Diese Resultate zeigen, dass die Grundidee der dynamischen Kapazitätssteigerung und das Prinzip der grünen Welle in Konflikt stehen. Dies liesse sich durch eine Weiterentwicklung des simulationsbasierten Verfahrens für die Erzeugung der Sperrzeitentreppen korrigieren und wird als vielversprechendes Thema für die zukünftige Forschung empfohlen. Des Weiteren haben die Experimente gezeigt, dass das nicht hierarchische Koordinationsverfahren die Konsistenz der Entscheidungen an den Grenzen der Dispositionszonen verbessern kann, wobei die Rechenzeiten den Ansprüchen der Betriebssteuerung genügen. Dies wird durch die Aufteilung des grossen zentralen Dispositionsproblems auf mehrere kleinere, signifikant einfacher zu lösende, lokale Dispositionsprobleme erreicht.

Dazu wurde eine mögliche Integration der entwickelten Methodik in die bestehenden Betriebssteuerungsprozesse der SBB untersucht und nachgewiesen, dass sie einfach in die bestehenden Informationsflüsse eingebunden werden kann. Damit können die Algorithmen dieser Dissertation zur Dispositionsunterstützung angewandt werden. Dadurch können die Konsistenz und Aktualität der Kundeninformation verbessert werden.

Riassunto

Questa tesi tratta di algoritmi per il supporto della gestione del traffico ferroviario e si inserisce nell'ambito di ricerca di *aumento dinamico della capacità*. Quest'ultimo mira a sviluppare soluzioni automatizzate per accrescere la precisione del piano orario e della sua esecuzione, in modo da poter utilizzare parte delle riserve attualmente esistenti nel sistema per pianificare nuovi treni, senza conseguenze negative sulla qualità del servizio. L'integrazione di questi algoritmi negli attuali flussi di informazioni dei sistemi di controllo del traffico ferroviario ha un ruolo fondamentale in questa tesi in quanto chiave per l'automatizzazione della gestione del traffico e, come conseguenza, per il miglioramento in termini di tempo e accuratezza dell'informazione ai passeggeri e agli attori coinvolti nel trasporto merci.

I lavori di ricerca precedenti in questo ambito si sono concentrati sul supporto alla costruzione del piano orario, alla gestione in tempo reale del traffico e sul controllo proattivo della velocità dei treni. La tesi di Fuchsberger (2012), in particolare, ha posto le basi per il lavoro riportato di seguito in quanto ha sviluppato un algoritmo, basato sulla rappresentazione del problema tramite un modello *resource conflict graph*, per la gestione del traffico in tempo reale in grandi stazioni ferroviarie con una topologia complessa e grandi volumi di traffico. Questa tesi mira ad estendere tale algoritmo e porta, principalmente, quattro contributi:

- l'estensione del modello resource conflict graph per la rappresentazione diretta di decisioni in più di un punto di esercizio, così da poterlo utilizzare per la modellizzazione del problema non solo in grandi stazioni con topologie complesse ma anche in tutti gli altri tipi di area.

- un metodo basato su simulazioni per generare le traiettorie alternative legate alle variabili del modello, secondo il principio dell'onda verde. In questo modo è possibile modellare accuratamente sia la dinamica dei treni che quella legata agli apparati centrali, e quindi considerare un grande numero di percorsi e profili di velocità alternativi.

- un approccio basato sulla generazione di colonne che permette di accorciare i tempi di risoluzione del modello per ottenere un nuovo piano orario adattato alla situazione attuale. In questo modo è possibile considerare una grande quantità di alternative per i percorsi, gli orari di partenza e arrivo, e i profili di velocità, senza eccedere il poco tempo disponibile per la gestione in tempo reale del traffico.
- un sistema non gerarchico basato su negoziazioni per coordinare la gestione di zone adiacenti, che quindi permette di decomporre il problema a seconda della potenza di calcolo disponibile.

Per garantire l'applicabilità dei metodi a una vasta gamma di situazioni, la validità e l'efficacia di questi quattro elementi è testata empiricamente utilizzando simulazioni dell'esercizio in due porzioni della rete ferroviaria svizzera con caratteristiche molto diverse:

- La tratta a doppio binario tra Berna e Worblaufen controllata da *Regionalverkehr Bern Solothurn (RBS)*, dove la topologia è semplice ma il traffico estremamente denso.
- La regione centrata su Brugg gestita dalle *Ferrovie Federali Svizzere (FFS)*, che si trova all'incrocio di due corridoi con traffico molto disomogeneo e dove convivono topologie di diversa complessità.

I test dimostrano empiricamente la validità dell'estensione del modello resource conflict graph nel generare piani orari eseguibili e l'efficacia del metodo basato sulla generazione delle colonne nel risolvere il modello nell'intervallo di tempo disponibile durante la gestione in tempo reale del traffico. Nella maggior parte degli scenari simulati, l'applicazione del piano orario generato usando i metodi sviluppati in questa tesi riduce il numero di conflitti che occorrono tra i punti di esercizio rispetto a una strategia First-Come-First-Served. Ciò migliora il flusso del traffico e potenzialmente riduce la quantità di energia consumata complessivamente, ma allo stesso tempo aumenta il ritardo cumulato a causa di tempi molto lunghi nelle riservazioni dei percorsi e, in alcuni, casi, anche di fermate aggiuntive ai punti di esercizio. Questi risultati indicano un'incompatibilità tra il concetto di aumento dinamico della capacità e il principio dell'onda verde, che è stato utilizzato per generare le traiettorie alternative che possono essere selezionate dal modello. Pertanto, un approccio per la generazione delle traiettorie che permetta ai treni di viaggiare più ravvicinati tra loro incontrando un numero di segnali dagli aspetti gialli dovrebbe essere al centro di lavori di ricerca futuri. Gli esperimenti hanno anche dimostrato che la strategia non gerarchica di coordinazione migliora notevolmente la coerenza delle decisioni ai portali tra le aree gestite da modelli indipendenti e che i tempi di calcolo dello schema distribuito sono compatibili con un utilizzo in tempo reale di questi algoritmi, dato che la complessità dei modelli locali è molto minore di quella del modello centralizzato.

Infine, l'integrabilità di questi metodi negli attuali sistemi di controllo del traffico ferroviario di FFS è stata investigata ed è stato possibile mostrare che il loro inserimento nei flussi di informazioni dei sistemi di gestione del traffico e di informazione alla clientela è semplice. Pertanto, si è potuto concludere che questi algoritmi possono effettivamente essere utilizzati come supporto per la gestione del traffico in tempo reale e che, come conseguenza, porterebbero a un miglioramento della coerenza e della tempestività delle informazioni per la clientela.

Acknowledgement

During the past four years at IVT, I had the opportunity to learn a lot about railways. I could spend most of my working time focussing on an extremely interesting topic, which has become the core of the the present thesis. This was only possible thanks to the funding by IVT, SBB Infrastructure Division, and SCCER and to the many persons who have contributed in different ways to this research project.

First of all, I would like to thank Prof. Dr. Ulrich Weidmann, the main advisor of this thesis, who gave me the opportunity to start this research project and provided continuous guidance along its development. Despite his many duties as vice-president of ETH Zurich and advisor of a large number of PhDs, he has always taken the time to discuss the progress of my work and give helpful advices for the successive steps. His deep understanding of the railway system has been an invaluable reference during these years, and the discussions with him an inexhaustible source of stimuli for the further development.

I am also very grateful to Prof. Dr. Anita Schöbel, who co-advised this thesis. Her always kind and detailed feedbacks helped me to refine this thesis and bring it to the current state.

During these years, I have learnt a lot about mathematical optimisation and its applications to railway problems. For this, I am very obliged to PD Dr. Marco Laumanns, who provided guidance and support from the very first stages of this thesis to its final redaction. The discussions with him and his suggestions have largely contributed to the shaping of the methodology proposed hereinafter. In addition, by proof-reading the manuscript he helped to considerably improve the quality of the English and of the use of the technical notation.

Having a co-referee at SBB has been very beneficial for the verisimilitude of the research assumptions and consequently for the validity of the methodology proposed in this thesis. This would have been impossible without Dr. Andreas Dietzsch and Dr. Peter Grossenbacher, who have ensured a constant very fruitful exchange with the railway industry and organized a huge number of contacts with SBB collaborators.

This work has highly benefited from regular interactions with the Swiss railways. SBB and RBS provided data for a realistic proof-of-concept and competent contact persons who promptly answered to most of my sometimes very detailed questions. In this context, I wish to express my gratitude to Dirk Abels, Adrian Egli and the PFI team for welcoming me into their offices in Berne, where I could acquire not only the data for the computational experiments, but also a broad overview of new algorithmic support methods for railway planning and operations being currently developed at SBB. I am very grateful to Roland Ackermann for opening the doors of the SBB Operation Center East and to Steve Bräker for letting me observe the actual work of a dispatcher. Besides, many thanks go to Daniel Achermann, Stefan Strässle, Steffen Oettich and Roman Caspar for their explanations of the systems ADL, CUS, and HOT currently used by SBB for energy efficient speed advisory, customer information and traffic flow optimisation at junctions. Moreover, I thank Médard Fischer and Peter Hug for supplying actual SBB operations and passenger flow data to calibrate the numerical experiments. I am very obliged to Helmut Iffländer for organising the collaboration with RBS and PSI Software AG. I am very grateful to Ulrich Reinert, Markus Enzler, and Hans Haldimann for introducing me to RBS railway operations and to PSI Software AG for the OpenTrack data I used for the computational experiments.

The collaborative and friendly atmosphere at IVT has greatly contributed to the quality of the present thesis and to my personal development. In particular, I thank my office mate Michael Schwertner for his availability and patience in answering my numerous questions concerning both railways and the German language. My gratitude goes also to Martin Sojka, Axel Bomhauer-Beins, and Patrick Braess who shared the incumbency of answering those questions after his leave. I am very obliged to our IT-support Peter Lorch who was always very attentive to the computational needs of this research project. In addition, I am grateful to Valerio de Martinis and Francesco Corman for the encouragement and support in making my work visible by writing scientific papers and attending conferences. Furthermore, I thank Gabrio Caimi and Martin Fuchsberger for finding the time to meet and share their knowledge on the topic in the earliest stage of the project.

Finally, this thesis would not have been possible without the support of my family and my partner. Ringrazio soprattutto i miei genitori per il sostegno ricevuto prima nella scelta dell'indirizzo di studio e poi nell'affrontare le difficoltà che ho trovato lungo il percorso che mi ha portata fino a questa tesi. Un grazie anche al resto della mia famiglia e al mio compagno che ha fatto del suo meglio per entusiasmarsi come la sottoscritta nell'osservare qualche grafico sul resoconto dei risultati della giornata.

Contents

Chapter 1

Introduction

During the last decades, demand for rail transport has grown considerably in Switzerland, and the number of public transport passengers and rail freights are estimated to continue growing rapidly (+51% public transport passenger-kilometers and +40% rail freight ton-kilometers by 2040 with respect to 2010, Federal Office for Spacial Development, 2016). In order to face the challenge to serve more passengers and transport more goods on the existing infrastructure without loss of service quality, there is a need for new support systems for railway traffic planning and control which consider customers' requirements in mixed-traffic railway networks.

The next subsection gives further details on the motivation of the present thesis. Section 1.2 states the research question and hypotheses. Section 1.3 outlines the structure of the thesis.

1.1 Motivation

In February 2014, the referendum for the funding and expansion of railway infrastructure (FABI, Bundesversammlung der Schweizerischen Eidgenossenschaft, 2013) has been accepted by Swiss voters, and now it ensures substantial economic resources to railways. Nevertheless, most of these resources have to be used to maintain the existing infrastructure, and only little remains for expansion projects. Furthermore, network expansions are not only expensive but also need a long time to be implemented and have irreversible effects on landscape and urban development. Thus, better utilisation of the existing infrastructure should be preferred to network expansion.

The higher the utilisation, the larger the conflict potential[1]. This

[1]see, e.g., Müller (2017) who analysed the conflicts being resolved in one of the SBB control centres and concluded that both the increased quantity of passengers in rush hours and the higher number of freight trains in off-peak times increase the number of conflicts

suggests that, without an improvement of the current planning and control processes, an increase of infrastructure utilisation to allocate the rapidly increasing demand might have negative consequences on the service quality, which might cost a shift to more pollutant motorised individual transport.

The increase of infrastructure utilisation to allocate additional travel demand without affecting the service quality has been denoted as *dynamic capacity improvement* by Weidmann et al. (2015), who identified automation as the key to achieve it. Railway automation is enabled by the following five technical elements:

- Automated real-time rescheduling: adapting the published timetable in real-time according to the current traffic situation to prevent and resolve conflicts;
- Speed profiles optimisation: computing speed profiles that ensure precise implementation of the schedule and prevent unplanned stops;
- Automatic train operation: driving trains automatically in order to apply the optimal speed profiles precisely;
- Integrated control for rescheduling and train driving: combining the previous three elements within a cascade control loops system to fully exploit the benefits of each one;
- Optimised distribution of time-reserves: recognising where buffer times are needed and tuning them accurately by considering the service intention and the characteristics of trains.

The key idea of dynamic capacity improvement is to increase planning and operating precision such that time reserves at capacity bottlenecks can be reduced or even eliminated. By using technologies that improve the precision of planning and operation, the Swiss Federal Railways, for example, anticipate that they would be able to plan around 15%-30% more trains on their network without the construction of new infrastructures (Meyer, 2017).

Figure 1.1 shows the research context of railway automation, in which this dissertation is situated. Railway automation enables higher precision at all five levels of planning and operations. At the first, more strategic, level it contributes to the definition of the functional requirements for customer oriented railway services with better estimations of transport chains, acceptable time windows, and required levels of services. These

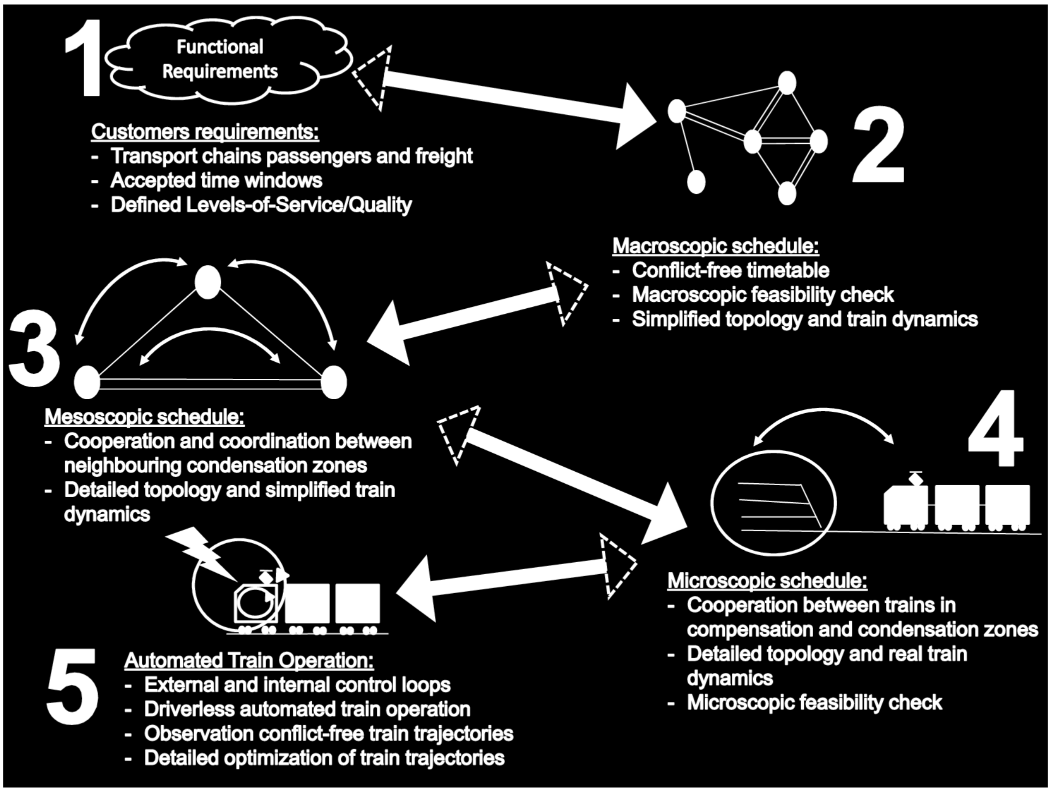


Figure 1.1: Conceptual framework for railway automation (adapted from Weidmann, 2015).

functional requirements are propagated to the other levels (solid arrows in Figure 1.1), where they act as the backbone of the boundary conditions and objectives for the corresponding tasks. Thus, a preciser definition enables the most appropriate resources and degrees of freedom to be exploited for planning and operating the different services. If the task of one of the following levels results infeasible due to the functional requirements, a request to soften them can be propagated up (dashed arrows).

At the second level, automation enables the generation of network-wide macroscopic schedules and vehicle assignments that better reflect the functional requirements and in shorter times than current processes. This allows the macroscopic scheduling and vehicle assignment processes to be moved from the current tactical timeframe (i.e., usually occurring once a year) to the operational one (i.e., the schedule and the corresponding vehicle assignment are computed every day or more often). Therefore, the system can react more quickly and reassign vehicles and infrastructures appropriately in case of major disruptions (e.g. suppress single trains or lines, or reroute them to bypass the affected area) and drastic demand changes (e.g., select appropriate vehicle size, or tune dwell times). The macroscopic schedule and the vehicle assignment plan are propagated down to the following levels, where they constitute the reference values for computing quality measures (e.g., delays) and train dynamics.

At the third level, automation improves coordination of neighbouring capacity bottlenecks and supports dynamic network decomposition in areas of tractable size (depending on, e.g., traffic density and complexity of track topology) for the next levels. In fact, planning and operations of railway networks are usually too complex to be managed at a microscopic detail level in a fully centralised way (see Section 3.3 for further reading). Thus, appropriate decomposition and coordination enable safeguarding global feasibility at a microscopic level and set the basis for generating microscopic local schedules that target a common global optimum. The network partitioning and the coordination targets are propagated down to the next level.

At the fourth level, automation enables fast generation of microscopic local schedules that are optimal with respect to the functional requirements, the time windows and the coordination targets fixed by the previous levels. Consequently, local schedules can be computed in real-time according to the current traffic situation, which further improves the reaction capability of the system considerably. Thus, the negative consequences of higher infrastructure utilisation mentioned above can be mitigated. Then, the microscopic schedules can be passed to the last level for application.

At the fifth level, automation enables trains to follow the instructions received by the previous levels more precisely. This prevents conflicts between trains, unnecessary braking actions and stops in front of red signals that would result in longer infrastructure occupations and consequently capacity losses. If the problem at any level results infeasible, it is reported to the upper one, which tries to enlarge the solution space of the following ones by softening the requirements that it propagates down.

This dissertation contributes to the third and fourth levels of railway automation (Figure 1.1). The thesis focuses on algorithms for coordinated real-time rescheduling in highly congested main station zones (or *condensation zones*[2]) and inbound lines in neighbouring regions (or *compensation zones*[2]). Figure 1.2 shows a conceptual view. These algorithms are designed to cope with disturbances, as defined by Cacchiani et al. (2014), and provide optimal schedules which minimise delays and/or customer inconvenience for the entire area considered. Disturbances are little deviations from the published timetable or failures of single technical components, which should be solved without modifying the rolling stock circulation and the crew shifts (Cacchiani et al., 2014). Examples are:

[2]Definitions in Section 4.1.1

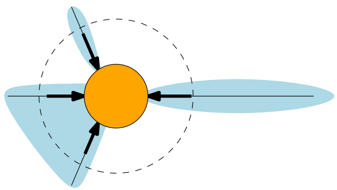


Figure 1.2: The orange circle denotes a condensation zone, and the blue areas indicate compensation zones. The dashed circle indicates the focus of this PhD: Rescheduling in condensation zones and inbound lines.

- Prolonged dwell time of a passenger train at a station, which might require some follow-up train to be assigned a different platform from planned;
- Delayed trains on the open track, which might cause secondary delays for follow-up trains on the same track or at junctions;
- Single tracks or switches that cannot be used properly, thus requiring trains to use alternative infrastructure resources than planned.

Disruptions, which are major events that can only be solved by changing rolling stock and crew assignments, are not in the scope of the current research. Examples could be:

- A power outage in a station or line, requiring all trains to by-pass the area;
- The failure of several tracks and switches, which forces trains to by-pass the area or to do short-turns.

Both operators and customers would benefit from the implementation of these rescheduling algorithms. Infrastructure managers could schedule more trains with reduced buffer times, without loss of stability. Information to customers could be improved, thanks to the pieces of information provided by these algorithms. In fact, the new schedule computed considering the current traffic situation can be used to inform customers more timely and consistently about train delays, platform changes, and connections missed.

1.2 Research question and hypotheses

This dissertation aims at answering the following question:

> How can algorithmic real-time rescheduling procedures support the resolution of small disturbances in railway operations in condensation zones and inbound lines, in order to make traffic management automatable and, as a consequence, improve consistency and timeliness of passenger information?

In order to answer this question, the following ten hypotheses are proposed:

H1 The relevant functional requirements of traffic management can be described in a way, which makes uniquely determinable whether a rescheduling method satisfies them or not.

H2 One or more mathematical models can represent the functional requirements from H1.

H3 The parameters for these models can be obtained with either the current safety system based on lineside signalling and tracking or the European Train Control System (ETCS) Level 2.

H4 Traffic management consists of subtasks, whose boundary conditions and goals can be written using the models of H2. Then, the subtasks of traffic management are optimisation problems in these variables.

H5 For each optimisation problem of H4, there is an algorithmic method that solves it within the short time available during real-time operations without the solution quality failing under a minimal definable threshold.

H6 There is a combination of the algorithms of H5 that returns either a feasible new schedule adapted to the current traffic situation or the proof that no such schedule exists within the modelling assumptions.

H7 The most common rescheduling measures (e.g., retiming departures, changing the train orders, giving up passenger connections, etc.) can be handled within the algorithms of H6.

H8 The application of the optimal rescheduling measure can be automated.

H9 Consistency and timeliness of passenger information can be improved thanks to the algorithms of H6.

H10 It is possible to define a category of ”small disturbances” and the proposed approach provides feasible schedules for every scenario rising from this category.

1.3 Outline

The thesis is structured as follows. Chapter 2 summarises the basic principles of railway operations and control and gives an overview of current traffic management and customer information at Swiss Federal Railways (SBB). The chapter is concluded with the functional requirements that new algorithmic rescheduling systems must meet in order to be applicable to real-world railway operations. Chapter 3 reviews the literature on mathematical models and algorithmic methods for railway operations planning and management, including approaches to problem decomposition. Chapter 4 presents the main contributions of the dissertation:

- a local rescheduling approach suitable for both condensation and compensation zones,
- a column generation based solution method for fast solution finding,
- and a coordination strategy for rescheduling over multiple areas.

Computational experiments are performed in Chapter 5. The experiments are designed to check the validity of the models and algorithms for railway traffic rescheduling, as well as to test their solution quality per computation time performance and their scalability in three test scenarios of different network size and timetable characteristics. In Chapter 6, the results of the experiments are discussed, the validity of the proposed approaches assessed, and the hypotheses checked. Furthermore, the resulting rescheduling approach is placed into the control loops of railway operations. The information flows requirements and time thresholds for decision making and implementation are specified. Chapter 7 summarises the main findings of the thesis, answers the research question, and outlines open problems for future research.

Chapter 2

Background and functional requirements

This chapter introduces the reader to the matter of this thesis and defines general concepts and boundary conditions that will be used hereinafter. Section 2.1 presents the basic principles of railway operations and control. Section 2.2.1 describes traffic control systems and processes currently applied by SBB[1], who serves as a case study for this work. Section 2.2.2 outlines the functioning of customer information system CUS that currently manages information for the entire Swiss public transport network. Section 2.3 summarises the functional requirements of traffic management and customer information that a new algorithmic rescheduling system must satisfy to be implementable in practice.

2.1 Principles of railway operations and control

The essential elements of a railway system are the infrastructure, the rolling stock and the operational rules and processes that ensure safety and efficiency (Pachl, 2015). Given the low resistance between the wheels and the rails, trains have long braking distances, and the look-ahead capability of train drivers is not sufficient to ensure at the same time safe operations and high infrastructure utilisation. For this reason train separation is safeguarded by interlocking and signalling systems. Furthermore, a timetable provides the train routes, orderings, and departure times at the relevant infrastructure points.

Safe railway operations are currently enforced by fixed block sig-

[1]Swiss Federal Railways, railway infrastructure manager and train operator in Switzerland

nalling. Railway networks are partitioned into blocks that are delimited by signals. Interlockings prepare train routes by moving switches, blocking the corresponding infrastructure elements, and regulate the access of trains into blocks through signals. Each block can host at most one train at a time, and trains must be able to stop within the distance they have reserved ahead (*Blocking time theory*, Pachl, 2008). The time an infrastructure resource is blocked by a train is referred to as *blocking time interval*, and the sequence of blocking time intervals on a train path is referred to as *blocking time stairway*. The blocking time interval begins when the train head passes the corresponding detection point and ends when the tail passes the axle counter or insulated joint situated at the end of the infrastructure resource plus a small release time needed by the interlocking to actually release the resource. Some state-of-the-art technologies[2] enable continuous communication with trains. Thus, loosing the dependence on track-side signals, such that rights of way can be transmitted at any moment and trains may be allowed to travel faster by reserving several blocks at a time. In a long term perspective, this signalling system may be replaced by moving block signalling.[3] Moving block signalling defines blocks dynamically depending on train position and speed. From a traffic management perspective, this is equivalent to fixed block signalling where infrastructure resources have infinitesimal length and trains can reserve several blocks at a time (Wendler, 2007). Thus, the current work focuses on fixed block signalling with special attention to those systems allowing trains to reserve several resources at a time.

Railway services are organised in timetables. Different timetable granularities exist. Usually, customers can visualise the itineraries on the network and the departure and arrival times at stops, while traffic managers work with a much finer schedule including local routes within junctions and passing times at all relevant infrastructure points (e.g., signals, track-circuits, axle-counters, etc.). In the context of this thesis, unless explicitly specified, the latter is considered. In general, according to Schranil and Weidmann (2013), *"an operation disturbance is an unexpected deviation from scheduled operations that requires the use of unplanned resources to resolve"*, where the resources are the set of vehicles, infrastructure, information (including traffic control systems), staff, operations (procedural operation aspects), and customers' time, money and comfort. The disturbances investigated in this thesis are relatively small and meet the definition of Cacchiani et al. (2014) as they *"can*

[2]e.g., ERTMS(/ETCS Level 2, with which the main European lines are currently being equipped.

[3]e.g., ERTMS/ETCS Level 3.

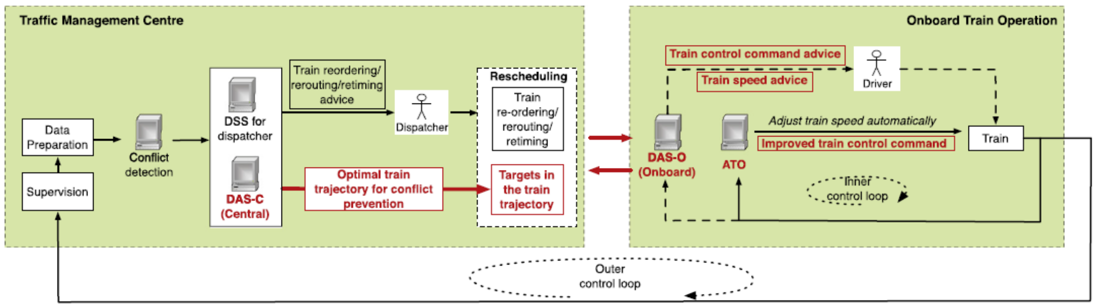


Figure 2.1: Railway control (Rao et al., 2016)

be handled by rescheduling the timetable only, without rescheduling the [vehicles and staff] *resource duties"*.

Railway traffic control ensures the compliance with the previously mentioned safety processes and service quality targets and is usually organised over two control levels (Lüthi, 2009). An "outer" control loop (often referred to as *traffic management*) is responsible for supervising **traffic** and intervening to prevent or resolve conflicts. Intervention consists in modifying the timetable with rerouting, reordering, and retiming trains, re-servicing (e.g., give up passenger connections, cancel trains, Corman and Meng, 2013), and speed advisory. An "inner" control loop focuses on the single **train** operation and adapts acceleration, cruising, coasting, and breaking phases to achieve the targets prescribed by the outer loop and respect the safety standards.

Conventionally, human dispatchers and train drivers are the decision makers in the outer and inner control loops, respectively. Thanks to recent developments in both algorithms and technical equipment, dispatchers and train drivers are increasingly supported by automatised advisory systems, which enable them to take more informed decisions and increase the overall system performance (Weidmann et al., 2015). Figure 2.1 shows an overview of the mentioned control loops with integrated decision support system (DSS) for dispatchers, driver advisory system (DAS), and automatic train operation (ATO). An important output of the outer control loop which is not mentioned in Figure 2.1 concerns customers. In fact, passengers and freight forwarders need to be informed about decisions that impact their transport chains. This interaction of traffic management with customer information is investigated by the current work.

The current work aims at improving decision support for the outer control loop (i.e., traffic management). The reference state for this control loop is given by the timetable with predetermined train orderings and local routings. The input and output data depend on infrastructure and rolling stock equipments. With classical lineside signalling equipment and no mobile communication, the position of trains can only be recorded at track circuits and trains can receive information only at fixed signals, which usually can display only a limited selection of information. With newer signalling systems such as ETCS Level 2, which is being rolled out on the main European lines, it is possible to communicate with trains continuously and a wider range of information can be exchanged. Given the long life span and the high cost, several generations of railway signalling equipments usually coexist within a railway system. Thus, the current work aims at developing approaches that are flexible with respect to granularity and frequency of input and output.

2.2 Current railway operations and customer information in Switzerland

The Swiss Federal Railways (*Schweizerische Bundesbahnen*, SBB) are Switzerland's largest infrastructure manager with a controlled territory of 3'172 track kilometres (SBB AG, 2015). The average traffic on each main track amounts to 101.5 trains per day. Traffic is generally mixed, with a daily average of 1.21 millions passengers and 205'000 net tons of goods transported. SBB also manages the customer information system for the entire Swiss public transport network. In this section, the main features of traffic management and customer information at SBB are presented.

2.2.1 Rail traffic management at SBB

The principles of traffic management at SBB are described in SBB AG (2014b). Railway operations are planned and controlled from five national centres: an operation centre (*Operation Centre Infrastruktur*) in Berne is responsible for network-wide planning, supervision, and reporting; while the control centres (*Betriebszentrale*) in Zurich, Olten, Lausanne and Pollegio supervise specific parts of the network and are involved in rescheduling, train control, and customer information. The Berne operation centre is also in charge for scheduling extra trains through the network and intervening in case of large disruptions. Up to one day before the schedule is executed, in each control centre, planners define local routes and refine the schedules of train movements through their controlled areas. Then, these pieces of information are used to generate train routing data (*Zuglenkdaten*, ZLD) for the interlocking control system ILTIS (Siemens, 2018). During railway operations, train routing data are automatically implemented by the centralised train control system (CTC,

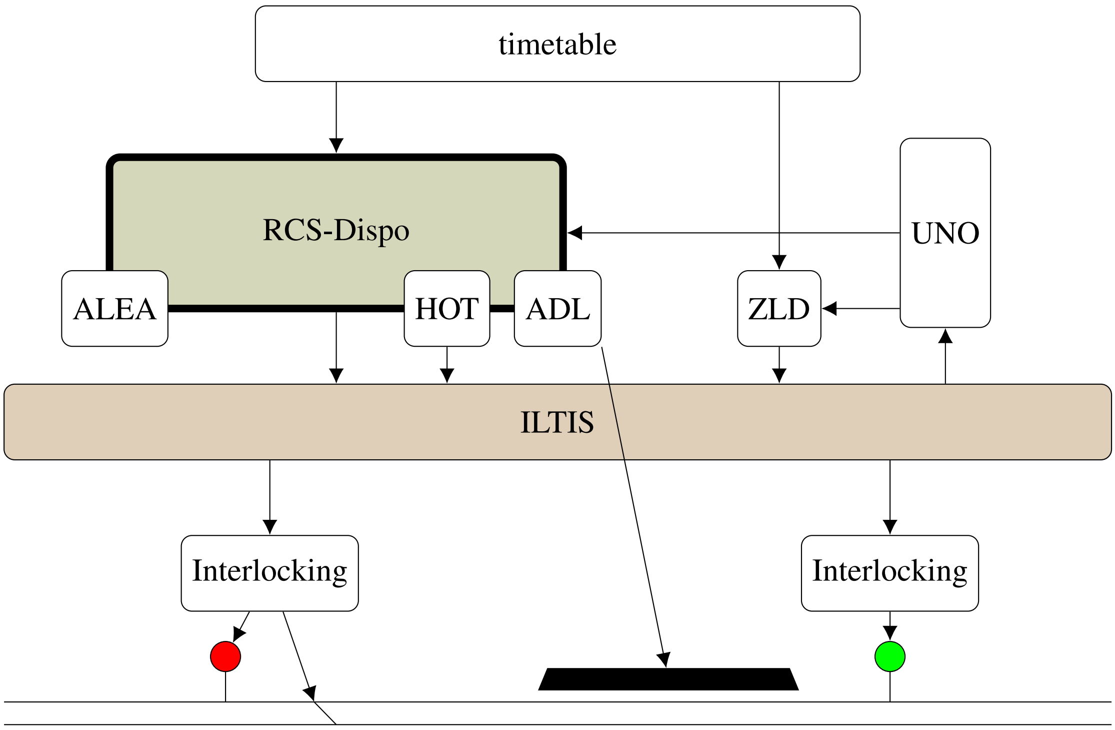


Figure 2.2: Railway Traffic Control at SBB (adapted from Oettich and Caspar, 2015)

refer to Pachl, 2015, for details) , which manages interlockings, unless an operator modifies them manually.

The railway infrastructure is composed by *operation points*, which are stations, stops, and junctions. To each control centre is assigned a number of adjacent operation points. In control centres, rail traffic management is organised hierarchically. Each dispatcher supervises a specific sector, which is a relatively large set of adjacent operation points. Each CTC operator manages a specific part of a sector (often a minor line with several small operation points or a part of a large one) in which he/she implements the dispatcher's decisions. Visualization and implementation support is available for both dispatchers and CTC operators. Interlocking plants are remotely managed with CTC system ILTIS. Timetable data including local routings and train orders at junctions saved as ZLD are applied by ILTIS to reserve train paths when detection points are passed. CTC operators can change ZLD data to reroute and reorder trains. At the upper decision level, system RCS-Dispo supports dispatchers by providing an overview of current and predicted traffic and highlighting conflicts (Dolder et al., 2009). Coordination among neighbouring sections is usually achieved via oral communication between dispatchers. Figure 2.2 shows the current rail traffic control support at SBB.

Specific decisions start to be automatised at SBB, and some fully automated systems have already been rolled out. RCS-Dispo autonomously applies wait-depart thresholds for passenger connections. Note that, even in case of fully manual operations, Swiss dispatchers must apply given thresholds for waiting delayed feeder trains (Z/C 505.1; Z/C 505.2). These thresholds are agreed between the infrastructure manager and the train operators during the definition of the yearly timetable. System RCS-HOT autonomously manages trains at Killwangen junction by setting train orders and speeds (Oettich and Caspar, 2015). In the Lötschberg base-tunnel, where a double track line converges to a single track, unplanned stops are prevented thanks to speed advices provided by the Automatic Function (Mehta et al., 2010). All other decisions to resolve small disturbances are still taken by human dispatchers and manually applied by CTC operators. Contingency plans to resolve large disruptions (entire blockade of a track or a line) developed by experienced traffic managers are stored in system ALEA. Dispatchers can adapt these solution patterns for real-time intervention. In addition, the entire network is equipped with system ADL (*Adaptive Lenkung*), which provides train drivers with speed advices to prevent full stops in front of red signals (Völker, 2013).

2.2.2 Customer information in Swiss public transport system

In many countries, each train operating company is responsible for information to the own customers. Differently, the Swiss Federal Office for transportation has mandated SBB to put on and manage a platform (CUS) which provides customer information for the entire public transport system in Switzerland (including regional and city networks, buses, and other train operating companies, Strässle and Schneeberger, 2017). This platform automatically receives data from several sources and sends information to many interfaces.

Figure 2.3 shows the information in- and outflows of customer information system CUS. Static databases CUSMDM and INFO+ contain the trains that are supposed to run through each station and the published timetable. Control system LTA and schedule database VDV update information about train, tram and bus positions and routes. The traffic management systems MIMI CUS and RCS-Dispo give information about predicted delays and connections. Train operating companies keep information about the planned service, train compositions (Kompo-EVU) and seat reservations (PLABE) up to date. Data are used to provide stations, online systems and staff with information about the current traffic situation

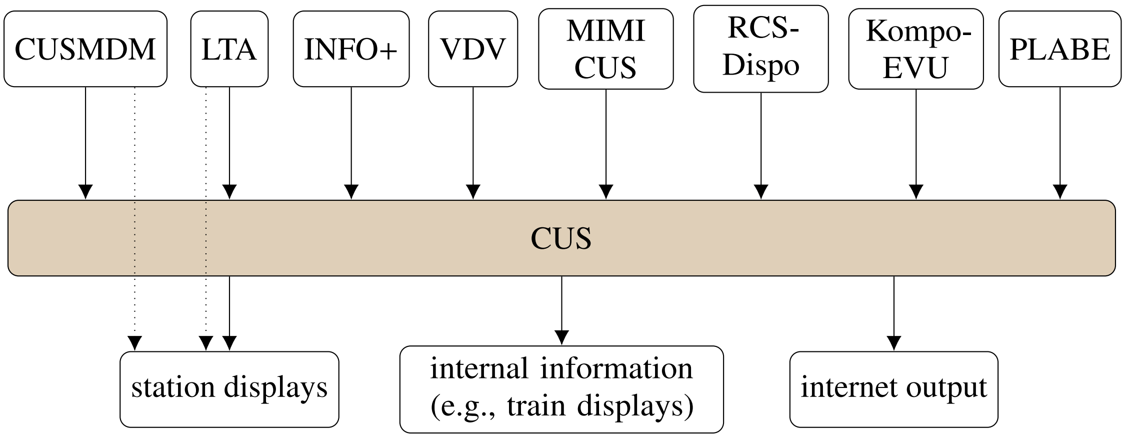


Figure 2.3: Customer information in Switzerland (adapted from Strässle and Schneeberger, 2017)

and the effects on passengers' transport chains. Most data are recorded and transmitted automatically. In control centres, about 50 customer information specialists care about the few manual inputs (i.e., source of disturbances and guiding passengers in case their transport chain is broken). An ongoing project aims at improving automation also in these tasks. Currently, this sophisticated system is only available for passenger traffic. Information about freight trains is still transmitted by phone. However, there is a trend towards refining tracking of freights.

The system is centralised, but in case of failure, customer information systems at stations still receive information about the scheduled trains from the static database and local information about train positions and routings from the interlocking plants. Time horizons for taking decisions about connections are agreed with the involved stakeholders to be six minutes for regional and eight minutes for long-distance traffic. There is no official time limit to change platforms or communicate delays because it strongly depends on station topology and alternative connections.

2.3 Functional requirements for decision support systems[4]

To reflect the actual railway operations and be useful for real-time traffic management, models and algorithms for rescheduling should satisfy a number of functional requirements. This section summarises these requirements which are drawn from the previous sections and the general

[4]This section is based on Toletti et al. (2015b), section 2

features of railway operations. The requirements can be grouped in three categories: (1) infrastructure; (2) rolling stock; (3) operations. In addition, systems aiming at supporting traffic management have to consider the data flows; i.e., the pieces of information that can be gathered from real-time operations, the messages that can be passed to customers, infrastructure and trains, and the times needed for such communications (Weidmann and Sinner, 2016).

2.3.1 Infrastructure

Railway infrastructure prescribes the static physical boundaries of railway operations. Infrastructure networks are usually represented using graphs. The granularities of these representations differ considerably depending on the applications. Macroscopic infrastructure models (or macroscopic topologies) usually contain only stations, junctions, and links between them. Information on quantity and length of tracks, average running time, and capacity is usually associated with these macroscopic links. Detailed representations of railway infrastructure including permissible speeds, gradients, radii, signals, block sections, and track-circuits are usually referred to as microscopic infrastructure models (or microscopic topologies).

According to Radtke (2014), conflict detection and resolution requires a microscopic infrastructure model. In fact, according to the blocking time theory (see Section 2.1 and Pachl, 2008), when the blocking time intervals of a resource of different train movements overlap, there is a conflict. The resources are defined by the train protection system and delimited by insulated joints and axles counters, which can only be represented on a microscopic infrastructure model. Thus, microscopic models are of particular relevance for the purpose of the present work.

Note that, if the energy supply of a network is not uniform, also this element should be considered by the infrastructure model. As this thesis aims at identifying suitable models for rescheduling traffic on the Swiss railway network, and this network is completely and uniformly[5] electrified, this element is neglected here.

2.3.2 Rolling stock

The rolling stock represents the dynamic physical boundaries of railway operations. The length of trains influences the occupation time of infrastructure resources and may limit overtaking and routing and influence

[5]with 15 kV, 16.7 Hz AC railway electrification system

rescheduling solutions (as shown, e.g., by Toletti et al., 2015a). The maximum speed on a track is determined by acceleration and braking capabilities of rolling stock combined with train length, mass, resistance, and infrastructure properties (curvature, gradient, signalling system and block length). The feasible dynamics for each train on each track can be obtained using these elements and considering the timetable (see, e.g., Brünger and Dahlhaus, 2014, for further information on train dynamics). Cordeau et al. (1998) highlighted that speed can be represented either by fixed profiles or by variable ones. Considering fixed speed profiles reduces the number of variables, which decreases the size of the model but also limits the degrees of freedom for finding solutions. For instance, energy consumption cannot be optimised using fixed speed profiles because the relevant degrees of freedom are missing.

Furthermore, nonuniform gauge, energy supply or safety system may limit the access to some parts of the network to specific types of rolling stock. Hereinafter, this will be neglected due to two reasons. First, the current work focuses on the adaptation of an existing timetable. And it is assumed that the rolling stock assigned to the trains in the original timetable is compatible with the scheduled routes. Second, this thesis focuses on small disturbances that do not require the modification of rolling stock assignments. Thus, it is not necessary to consider track access limitations due to the gauge, the energy supply, and the safety system. Only train dynamics and track access limitations due to the train length are considered among the functional requirements for the decision support systems to be developed in this work.

2.3.3 Train protection and control processes

Additional requirements are prescribed by safety, monitoring and intervention features, and operational interdependencies. As mentioned in Section 2.1, safe operations are currently ensured by fixed block signalling systems and there is a trend towards systems allowing trains to block several resources at a time (e.g., ETCS Level 2). Thus, to account for both the current and future safety requirements, models for traffic management must be able to ensure safe operations with the current fixed block signalling system with the possibility to reserve several blocks at a time. [6]

Given that trains are not self-organising, if a train cannot reserve the

[6]If the signalling system of the reference network is not uniform, then the representations of infrastructure and rolling stock should include the corresponding pieces of information.

next resources on its itinerary, it cannot continue its run and has no other possibility than to stop in front of a red signal. This can be due to

conflict case 1 another train is blocking one required resource,

conflict case 2 the resource is out of service (e.g., for maintenance),

conflict case 3 the waiting for an ”enabling event” such as a scheduled meet-pass or a passenger connection or even the arrival of the rolling stock at the departure point.

To prevent or resolve this situation dispatchers have different possibilities:[7]

- retiming events, e.g., delay the arrival at or the departure from an infrastructure resource such that the train uses the contended resource after the one currently occupying it (conflict case 1) or after the ”enabling event” has taken place (conflict case 3);
- reordering trains on shared infrastructure, e.g., give the right of way to the first train arriving at a convergence of multiple lines even if it was scheduled to follow another train which has not reached the point yet (conflict case 3);
- rerouting trains locally, e.g., a platform change, or globally, e.g., let a train use a siding instead of the scheduled main track in order to overtake another slower train (conflict case 1), or by-pass a switch which is closed due to maintenance works (conflict case 2);
- re-servicing, e.g., modify the stopping pattern or turnaround before destination to bypass unavailable resources (conflict case 2) or break a connection if the feeder train is delayed or cancel a train if the previous service using the same rolling stock is delayed (conflict case 3).

Reservicing actions are usually only applied to resolve large perturbations (see, e.g., Schranil and Weidmann, 2013, page 12). As the focus of this work is to provide support in resolving small instances, only connections breaks and cancellations will be considered here. The reason for considering cancellations is the following. In situations with very dense traffic, even minor disturbances might temporarily oversaturate some part of the network. In these cases, some movement must be deferred until capacity becomes available again. But if it requires that a passenger train departs after the next service on the same line, it is better to cancel the train movement.

[7]Nomenclature from Corman and Meng (2013).

2.3.4 Data availability and information flow

Knowing the current train positions, the state of the infrastructure resources, and the amount of customers that can be affected by disturbed operations is very important for real-time traffic management. Weidmann and Sinner (2016) outline the functional data flow requirements for traffic management at SBB. With current communication systems it is only possible to position trains when they travel on track circuits. With ETCS Level 2, it will be possible to update the position of trains almost continuously (position and speed information are transmitted to the Radio Block Centre every six seconds). SBB infrastructure data are managed by system UNO, which reports whether the infrastructure elements are in service or out-of-order. The average number of passengers served by each SBB service (daily granularity) is available thanks to automatic counters installed on the doors of newest trains, estimates by train staff (older trains) and yearly origin-destination surveys.

Safety relevant information can only be passed by the interlocking system. Thus, currently, only signals can be applied to give right of way. Other information such as speed advices can be continuously transmitted to trains using GSM-R system. With Radio Block based signalling (e.g., ETCS Level 2) it will be possible to transmit rights of way directly to trains. Customer information can be passed to CUS. As highlighted in the previous sections (Sections 2.2.1 and 2.2.2) most of the rescheduling decisions must be fixed some time before the corresponding events occur (e.g., entering a block or departing from a station). In particular, the following thresholds should be considered for the current systems:

- retiming: no hard limit, but the sooner the better;
- reordering: no specified limit but before the route is locked for the first train;
- rerouting (local): six minutes for safety reasons, but larger times may be required by platform changes in those stations where trains have to be charged with luggage or supplies for restaurant cars;
- rerouting (global): not considered here;
- reservicing: six/eight minutes for connection break (regional/long-distance), no hard limit for train cancellations.

Table 2.1: Summary of functional requirements

	requirements	model decision variables	model constraints	model objective	solution method	input	output
infrastructure	microscopic track topology (possibly at track-circuit level)		x				
rolling stock	train length		x				
	train dynamics	x	x				
operations	safety system enabling the reservation of multiple blocks		x				
	target the published timetable		x	x			
	minimise train delays			x			
	minimise customer inconvenience			x			
	exclude closed tracks from train paths		x			x	
	retiming of train departures, arrivals and passing	x	x	x			x
	reordering of trains at junctions	x	x	x			x
	rerouting trains locally (including platform assignment)	x	x	x			x
	keep/break connections	x	x	x			x
	train movements' cancellations	x	x	x			x
information	current train location (and speed)					x	
	current status of infrastructure elements					x	
	adapted route setting information for interlocking (including train ordering)						x
	adapted departure, arrival and speed profile information for trains						x
	adapted departure, arrival and connection information for customer information systems						x
	adapted schedule information is ready for transmission at a suitable time for implementation				x		

2.3.5 Summary of requirements

Table 2.1 summarises the requirements that the traffic management algorithms developed by the current research must satisfy. The structure of such algorithms will be explained at a later point, but for now it can be assumed that they consist mainly of a mathematical programming model which describes the problem, a solution method which fixes the values of the model decision variables, an input which sets the parameters of the model from train location data, and an output which converts the decision variables into commands for trains and interlocking plants. The first column indicates the kind of resource each requirement is linked to, the second lists the requirements, and the others indicate which part of the algorithm is responsible for each of them.

Chapter 3

Literature review

In this chapter the most relevant literature for the development of an automated rescheduling system is reviewed. Section 3.1 summarises the mathematical models that have been used within previous works to represent railway operations and analyses them with respect to the functional requirements listed in Section 2.3 and the optimisation objectives proposed so far. Section 3.2 recalls the algorithmic methods that have been applied to solve these models and outlines the possible applications in the current research. Finally, Section 3.3 examines the problem decomposition approaches that have been exploited to split optimisation problems related to railway operations to reduce the computational effort.

3.1 Mathematical models[1]

Several mathematical models to support different tasks in railway traffic planning and operations have been developed during the last decades. Different models have been developed to represent different properties of railway operations depending on the intended application: macroscopic timetabling, microscopic timetabling[2], or rescheduling. Hereinafter, *timetabling* and *rescheduling* are used accordingly to the definition given by Törnquist (2006):

Definition 3.1 *(Törnquist, 2006)*

> Scheduling (or timetabling) *is the process of constructing a schedule from scratch, while* rescheduling (or dispatching) *indicates that a schedule already exists and will be modified*[3].

[1]This section is based on Toletti et al. (2015b)

[2]A macroscopic(microscopic) timetable is a schedule on a macroscopic(microscopic) topology.

[3]Remark on the notation: The term *dispatching* is sometimes used to indicate a timetabling process taking place short before the train departure. Thus, to avoid confusion, this term is not used in this thesis.

This section presents and analyses the representatives of the main classes of formulations for timetabling and rescheduling that are most prominent in the literature and that were identified in the overview of papers by Cacchiani et al. (2014), Corman and Meng (2013), Törnquist (2006), Lusby et al. (2011), and Cordeau et al. (1998). The presentation follows the outline of Baccelli et al. (1992)'s *"guidelines for modelling of dynamic systems"*: (1) describe the evolution of each unit in the system individually; (2) integrate the interactions among the units; (3) tackle the problem of initialisation. In this framework, the *units* are the train runs, which are fully defined by sequences of discrete events coinciding with arrivals and departures at relevant infrastructure points.

The complexity of initialisation depends on *when* the timetabling or rescheduling process is executed. Rescheduling is usually performed in real-time. Thus, initialisation must consider the up-to-date position of the units in the system. Timetabling can be performed both in real-time or some time in advance (usually one year for passenger services in Europe). In the first case, initialisation is the same as for rescheduling. In the latter case, it is much simpler. Given that long-term predictions of the system evolution are not very reliable (see, e.g., Kecman et al., 2015), real-time timetabling and rescheduling are often inserted into a *rolling horizon framework*. The main idea is that, at predetermined times, the optimisation problem is redefined with the most up-to-date information and resolved. Each time the evolution of the investigated system is only evaluated for a limited time period, and only decisions that are due in the first time interval of that period are made (see, e.g., Sethi and Sorger, 1991, for further reading on general rolling horizon frameworks).

In Section 3.1.1, the formulations modelling time as a continuous variable are presented. In Section 3.1.2, time-indexed formulations are introduced. All models are illustrated using the network of the *Railway Operations Laboratory* (EBL) at the ETH Zurich. Figure 3.1 shows the macroscopic topology with the six stations considered by the following examples. Two trains are assumed to travel through it. The first train travels from Testadt (T) to Pewald (P) passing Ypslikon (Y) and Zetthausen (Z), and the second train travels from Utal (U) to Pewald (P) passing Testadt (T), Ypslikon (Y), and Wedorf (W). In Ypslikon, a passenger connection is expected. Figure 3.2 shows station Ypslikon, which is used as an instance of a microscopic topology. To make the modelling task more challenging, it is assumed that the first train reaches Ypslikon from signal $A105$ and the second one from $A205$.

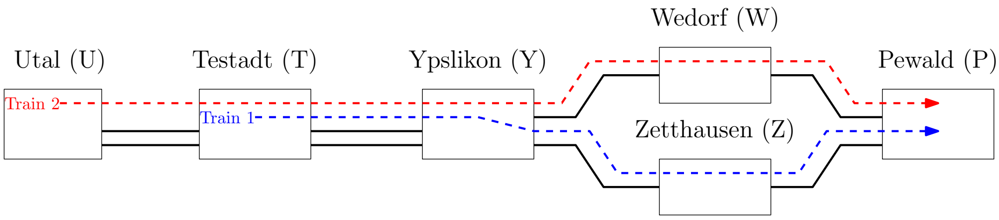


Figure 3.1: Macroscopic topology of the network of the *Railway Operations Laboratory* at the ETH Zurich.

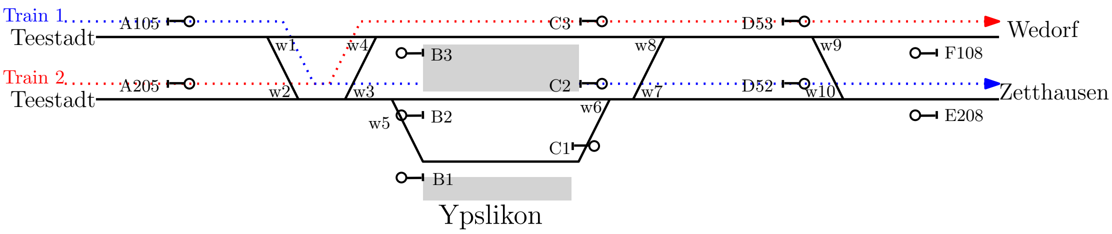


Figure 3.2: Microscopic topology of station Ypslikon in the *Railway Operations Laboratory* at the ETH Zurich.

3.1.1 Continuous-time formulations

This section presents the best known continuous-time formulations that have been applied to railway timetabling and rescheduling, which are:

- Event Scheduling Problem (ESP);
- Alternative Graph (AG);
- FlexiblePath (FP);
- Route-lock-Route-release formulation (RR).

The Periodic ESP (PESP) has been extensively used to produce periodic macroscopic timetables for given sets of train lines (Serafini and Ukovich, 1989; Peeters and Kroon, 2001; Caimi, 2009; Kroon et al., 2009; Herrigel et al., 2013b). AG has been primarily exploited to generate microscopic schedules both off- and on-line (Mascis and Pacciarelli, 2002; Mazzarello and Ottaviani, 2007; D'Ariano et al., 2007a, 2008; Corman et al., 2010b; D'Ariano et al., 2014). A macroscopic extension of AG has been proposed by Kecman et al. (2013) for macroscopic rescheduling. FP has been applied to find train routes and schedules in a microscopic topology in real-time (Yan and Yang, 2012; Mu and Dessouky, 2014). Törnquist and Persson (2007) proposed a very similar formulation based on a topology with a coarser granularity, which was applied for network-wide rescheduling. RR has been developed by Pellegrini et al. (2014) for microscopic rescheduling.

Modelling individual train runs Let v_S^z denote a discrete event that is related to some train z and takes place at some node S of the underlying topology. In all four formulations, continuous variables t_S^z represent the times when the discrete events v_S^z occur.

$$t_v^z \in \mathbb{R}_{\geq 0}, \quad \forall v_S^z \tag{3.1}$$

The relevant discrete events for ESP and for Kecman et al. (2013)'s AG extension are the arrivals and departures at nodes of a macroscopic topology, which coincide with stations where services begin or end or connections take place. In contrast, the discrete events of AG, FP, and RR are associated with nodes of microscopic topologies. AG considers not only arrivals and departures at all stations, but also at signals. The discrete events associated with FP are the entrances in and the exits from sections of infrastructure that correspond to tracks between junctions and can host at most one train at a time (refer to Lu et al. (2004) for a complete description of the underlying network partitioning). The granularity for RR is finer, as the sections are delimited by track-circuits. Figure 3.3 shows the *event-activity network* (i.e., the graph associated with ESP model) for the sample macroscopic timetabling problem presented in Section 3.1 on the macroscopic topology depicted in Figure 3.1. Figure 3.4 shows the sample microscopic timetabling problem in station Ypslikon (the track topology is shown in Figure 3.2). In both figures, the white nodes represent the discrete events. The black nodes indicate *zero events*, which are events that take place at time zero independently from any other event.

Assuming that the route of a train z is known in advance, its run can be fully described as the sequence of discrete events (v_S^z) associated with the topology nodes S on that route. Zero events can be employed for fixing time intervals, when necessary. In continuous-time formulations, the relations between the events are described by inequalities of the form

$$cType : t_{S_2}^z - t_{S_1}^z \geq f_{(S_1,S_2)}^z \tag{3.2}$$

These constraints fix the minimum time separation $f_{(S_1,S_2)}^z$ allowed between two events $v_{S_1}^z, v_{S_2}^z$, and they can describe:

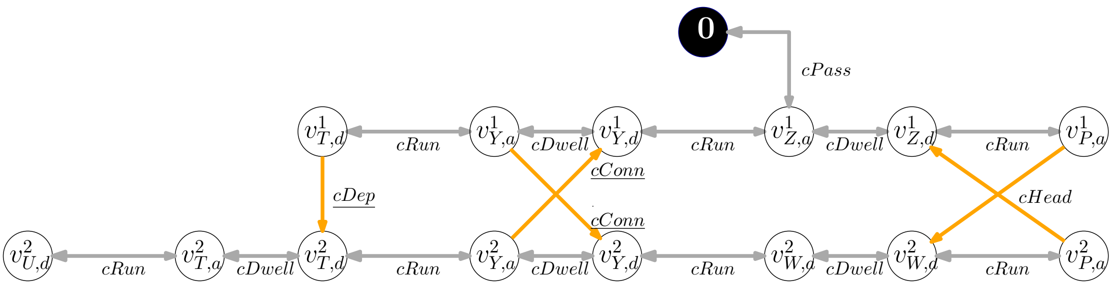


Figure 3.3: ESP for two train runs on the infrastructure shown in Figure 3.1. The white nodes represent the discrete events, and the black node represent a zero event. The grey left-right arrows coincide with constraints (3.3) and the orange with (3.10).

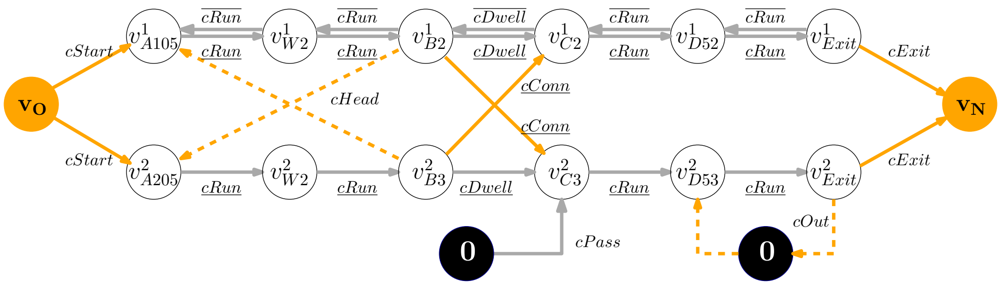


Figure 3.4: AG for two train runs in Ypslikon (see track topology in Figure 3.2). The white nodes correspond to the discrete events, the black nodes indicate zero events, and the orange ones are the start node (v_0) and the end node (v_N). The solid grey lines represent constraints (3.2), the solid orange constraints (3.10), and the dashed constraints (3.14).

- $\underline{cRun}$: the minimum running time $f^z_{(S_1,S_2)}$ from a departure $v^z_{S_1}$ to the following arrival $v^z_{S_2}$;
- $\overline{cRun}$: the maximum running time $-f^z_{(S_1,S_2)}$ from a departure $v^z_{S_2}$ to the following arrival $v^z_{S_1}$;
- $\underline{cDwell}$: the minimum dwell time $f^z_{(S_1,S_2)}$ between an arrival $v^z_{S_1}$ and the following departure $v^z_{S_2}$;
- $\overline{cDwell}$: the maximum dwell time $-f^z_{(S_1,S_2)}$ between an arrival $v^z_{S_2}$ and the following departure $v^z_{S_1}$;
- $\underline{cPass}$: $v^z_{S_1}$ corresponds to a zero event, and $v^z_{S_2}$ cannot take place before $f^z_{(S_1,S_2)}$ (e.g., the departure from a station cannot take place

before the timetabled time, Mazzarello and Ottaviani, 2007, refer to these constraints as *passing constraints*);

- $\overline{cPass}$: $v^z_{S_2}$ corresponds to a zero event, and $v^z_{S_1}$ cannot take place after $-f^z_{(S_1,S_2)}$;
- $\underline{cOverall}$: the minimum running time $f^z_{(S_1,S_2)}$ from the departure from the first station $v^z_{S_1}$ to the arrival at destination $v^z_{S_2}$ (usually defined for ESP only);
- $\overline{cOverall}$: the maximum running time $-f^z_{(S_1,S_2)}$ from the departure from the first station $v^z_{S_2}$ to the arrival at destination $v^z_{S_1}$ (usually defined for ESP only);

ESP usually contains for each minimum time constraint also the corresponding maximum time constraint. Therefore, the inequalities are identified by $cRun$, $cDwell$, $cPass$, and $cOverall$ and are associated with intervals $\left[\underline{f}^z_{(S_1,S_2)}, -\overline{f}^z_{(S_2,S_1)}\right]$:

$$cType : \underline{f}^z_{(S_1,S_2)} \leq (t^z_{S_2} - t^z_{S_1}) \leq -\overline{f}^z_{(S_2,S_1)} \tag{3.3}$$

Figures 3.3 and 3.4 show the relations for ESP and AG listed above as solid grey lines.

Note that, if a train stops, it cannot enter the successive section with maximum speed. This issue was addressed by Rodriguez (2007), which inserted the computation of speed profiles in a model similar to RR. D'Ariano et al. (2007b) iteratively updated the realisable running times (i.e., the right-hand-side of (3.2) for constraints of type $\underline{cRun}$) according to the solution given by AG in the previous step. Oettich and Caspar (2015) modelled the different speed profiles for each train using the concept of *alternative arcs*, to be explained later in this section.

FP does not need the routes to be fixed in advance but includes a binary variable x^z_i for each train z and infrastructure node i indicating whether z uses i. O_z and D_z denote the origin and destination node for z. $\delta_-(J)$ and $\delta_+(J)$ denote sets of nodes connected with junction J from the two travel directions. As each train follows a unique continuous route, the following additional constraints must be satisfied:

$$x^z_i = \begin{cases} l1, & \text{if train } z \text{ travels through infrastructure} \\ & \text{node } i \\ 0, & \text{else} \end{cases} \tag{3.4}$$

$$x^z_{O_z} = x^z_{D_z} = 1 \quad \forall z \tag{3.5}$$
$$\sum_{i \in \delta_-(J)} x^z_i = \sum_{i \in \delta_+(J)} x^z_i \quad \forall J, \forall z \tag{3.6}$$

D'Ariano et al. (2014) extend the AG formulation to feature rerouting. This formulation and RR include a binary variable x^z_i for each train z and for each route i from the entrance of z in the considered area to its exit. This variable denotes whether z follows route i. In this case, the route continuity constraint (3.6) is no longer needed, and the uniqueness constraint (3.5) becomes

$$x^z_i = \begin{cases} 1, & \text{if train } z \text{ uses route } i \\ 0, & \text{else} \end{cases} \tag{3.7}$$
$$\sum_i x^z_i = 1 \quad \forall z \tag{3.8}$$

In all three cases (FP, extended AG, and RR), the constraints (3.2) become

$$cType : t^z_{S_2} - t^z_{S_1} + M(1 - x^z_i) \geq f^z_{(S_1,S_2)} \tag{3.9}$$

where M is large positive constant value.

Modelling interactions The interactions between the different runs can be represented using the time variables t^z_S. In the classical ESP and AG formulations (i.e., no routing), a relation between an event $v^z_{S_1}$ of a train z and an event $v^w_{S_2}$ of another train w can be represented as

$$cType : t^w_{S_2} - t^z_{S_1} \geq f^{z,w}_{(S_1,S_2)} \tag{3.10}$$

which can model:

- $\underline{cConn}$: minimum connection time $f^{z,w}_{(S_1,S_2)}$ from the arrival $v^z_{S_1}$ of z to the departure $v^w_{S_2}$ of the destination train w;
- $\overline{cConn}$: maximum connection time $-f^{z,w}_{(S_1,S_2)}$ from the arrival $v^w_{S_2}$ of w to the departure $v^z_{S_1}$ of the destination train z;
- $\underline{cDep}$: minimum separation time $f^{z,w}_{(S_1,S_2)}$ between the departures $v^z_{S_1}, v^w_{S_2}$ of trains z, w with similar services;
- $\overline{cDep}$: maximum separation time $-f^{w,z}_{(S_2,S_1)}$ between the departures $v^w_{S_2}, v^z_{S_1}$ of trains z, w with similar services.

In ESP, these interactions can also be modelled using intervals (crf. (3.3)). Despite no explicit mention in the literature, these interactions could be modelled within FP and AG's extension by modifying (3.10) as follows:

$$cType : t^w_{S_2} - t^z_{S_1} + M(1 - x^z_i) + M(1 - x^w_j) \geq f^{z,w}_{(S_1,S_2)} \tag{3.11}$$

where i, j are the either infrastructure resources (FP) or routes (RR and AG) connected with the discrete events $v^z_{S_1}$ and $v^w_{S_2}$ respectively. To coordinate the movement of all trains and to measure the total time needed to bring all trains to destination, AG contains a *Start Node* v_O corresponding to a zero event preceding all other events in the system and an *End Node* v_N corresponding to an event that is enabled by the completion of all other events in the system. Orange disks represent these nodes in Figure 3.4. A constraint $cStart$ connects the node of the first event $v^z_{S_1}$ of every train to the Start Node. A constraint $cExit$ imposes that the event of the End Node takes place after the last event $v^z_{S_2}$ of every train run.

$$cStart : t^z_{S_1} - t_O \geq 0 \quad cExit : t_N - t^z_{S_2} \geq 0 \tag{3.12}$$

This for RR and D'Ariano et al. (2014)'s AG extension becomes

$$cStart : t^z_{S_1} - t_O + M(1 - x^z_i) \geq 0 \quad cExit : t_N - t^z_{S_2} + M(1 - x^z_i) \geq 0 \tag{3.13}$$

The interactions listed above are shown as solid orange lines in Figures 3.3 and 3.4. Pellegrini et al. (2014) use equations similar to (3.13) for modelling connections that have to be forced and rolling stock circulations.

The order of trains at conflict points is not usually known in advance but has to be chosen by the timetabling or rescheduling procedure. PESP models headway constraints applying a modulo operator to constrains of type (3.10) on the events corresponding to the trains entering and leaving common sections. If periodicity is dropped, these constraints must be reformulated in disjunctive form. AG represents headway constraints as pairs of *alternative arcs*, which correspond to disjunctive constraints of the form

$$cHead : (t^w_{S_3} - t^z_{S_2} \geq f^{z,w}_{(S_1,S_2,S_3,S_4)}) \vee (t^z_{S_1} - t^w_{S_4} \geq f^{w,z}_{(S_1,S_2,S_3,S_4)}) \tag{3.14}$$

where $v^z_{S_1}$ corresponds to train z entering the common section; $v^z_{S_2}$ to z leaving it; $v^w_{S_3}$ to train w entering it; $v^w_{S_4}$ to w leaving it. Note that with some safety systems, the number of blocks reserved for a train run depends on its speed. Thus, the alternative arcs in AG should be updated according to the speed of the trains within the iterative procedure proposed by D'Ariano et al. (2007b). Alternative arcs can also be represented using binary variables $h_i^{z,w}$ as follows:

$$h_i^{z,w} = \begin{cases} 1, & \text{if train } z \text{ travels through infrastructure} \\ & \text{resource } i \text{ before train } w \\ 0, & \text{else} \end{cases} \tag{3.15}$$

$$\begin{aligned} cHead: \quad & t^w_{S_3} - t^z_{S_2} + Mh_i^{z,w} \geq f^{z,w}_{(S_1,S_2,S_3,S_4)} \\ & t^z_{S_1} - t^w_{S_4} + M(1 - h_i^{z,w}) \geq f^{w,z}_{(S_1,S_2,S_3,S_4)} \end{aligned} \tag{3.16}$$

Analogously, FP, RR, and the mentioned AG extension model headway constraints as

$$\begin{aligned} & cHead: \\ & \quad t^w_{S_3} - t^z_{S_2} + Mh_i^{z,w} + M(1 - x_j^z) + M(1 - x_k^w) \geq f^{z,w}_{(S_1,S_2,S_3,S_4)} \\ & \quad t^z_{S_1} - t^w_{S_4} + M(1 - h_i^{z,w}) + M(1 - x_j^z) + M(1 - x_k^w) \geq f^{w,z}_{(S_1,S_2,S_3,S_4)} \end{aligned} \tag{3.17}$$

where $i = j = k$ in FP and j, k are routes containing i in RR and AG. A basic assumption of FP is that each section is longer than the longest train, while AG and RR model longer trains with longer alternative arcs (i.e., going from the end of a section to the beginning of a previous section) and larger $f^{w,z}_{(S_1,S_2,S_3,S_4)}$ parameters, respectively. The dashed orange lines in Figures 3.3 and 3.4 represent this second type of interactions.

Corman et al. (2012a) use alternative arcs for modelling connections. While connections represented using (3.10) are forced, the formulation via alternative arcs allows connections to be given up. Disjunctive constraints (3.14) can also model out-of-order constraints (*cOut*), which represent the closure of an infrastructure resource (Mazzarello and Ottaviani, 2007). In this case, trains are forced to pass the closed section either before the closure or after the reopening time.

Initialisation While the timetabling versions of ESP, AG, and FP need no particular initialisation, their real-time and rescheduling versions and RR need the initial positions of all the trains. Mazzarello and Ottaviani (2007) model the initial position of each train in AG with a *position node* which is inserted into the train path. The position node is connected with v_0 through a constraint of type (3.2), where $f^z_{S_1,S_2}$ denotes the current time. Due to their similarity with AG, one can imagine to apply the same initialisation to the other models presented so far. Pellegrini et al. (2014)

propose a rolling horizon framework which considers even the decisions taken in previous steps.

3.1.2 Time-indexed formulations

Several timetabling and rescheduling approaches that allow routing choices are based on discrete-time (sometimes referred to as time-indexed) models. Their goal is finding conflict-free routes and schedules simultaneously. Also delay management formulations usually model time discretely. The classical delay management problem does not consider operational feasibility but only the effects of delays and wait-depart decisions for scheduled connections on passengers. Recent developments have pushed delay management towards an integration of railway operations, which makes them worth of consideration for the purpose of the current work. This section presents the following models:

- Arc Packing Problem (APP) and its weak version (APP');
- Path Packing Problem (PPP);
- Arc Configuration Problem (ACP);
- Path Configuration Problem (PCP);
- Resource Tree Conflict Graph (RTCG) and Tree Conflict Graph (TCG);
- Resource Conflict Graph (RCG);
- REFormulated Simultaneous train Rerouting and Rescheduling (REF-SRR);
- Capacitaded Delay Management (Cap-DM).

Packing problems APP and PPP are the the most prominent approaches in the literature. APP, APP', ACP, PPP, and PCP have been used to route trains on macroscopic topologies and produce aperiodic timetables (Caprara et al., 2002; Borndörfer and Schlechte, 2007; Fischer and Helmberg, 2010). RTCG has been applied to allocate blocks in microscopic timetabling (Caimi, 2009; Caimi et al., 2011). RCG has been used by Caimi (2009) for microscopic timetabling, and a version called Static Train Dispatching has been used by Fuchsberger (2012) for microscopic rescheduling. Lusby et al. (2013) developed a variant of RCG that permits partial allocation of resources for real-time timetabling on junctions.

REF-SRR has been proposed by Meng and Zhou (2014) for microscopic rescheduling as a reformulation of the continuous-time formulation FP. Cap-DM has been first proposed by Schöbel (2009) for wait-depart decisions about passenger connections including line capacity considerations. Dollevoet et al. (2015) extended it to consider station capacity and alternative platform assignments.

Modelling individual train runs As for continuous-time models, the discrete events modelled by time-indexed formulations are arrivals at and departures from relevant infrastructure points. The relevant infrastructure points coincide with stations in APP, APP', ACP, PPP, and PCP, and with the endpoints of the infrastructure resources in RTCG, TCG, RCG, and REF-SRR. Each event can take place at several times. APP, APP', ACP, PPP, PCP, and REF-SRR contain a node for each time and place that can host such an event. In contrast, RCTG, TCG, and RCG contain only times when trains may enter the controlled area or start runs from station platforms. Similar to continuous-time formulations, Cap-DM contains only one decision variable per discrete event modelling the time a train arrives to or departs from a station. All formulations contain a source node s_z and a sink node t_z for each train z. These nodes coincide with the origin and destination infrastructure node or event in REF-SRR and Cap-DM, while they correspond to artificial source and sink nodes in all other formulations.

APP, APP', ACP, RTCG, and TCG include a binary variable x_a^z for each action a separating two events that can be consecutive for train z.

$$x_a^z = \begin{cases} 1, & \text{if action } a \text{ of train } z \text{ is chosen} \\ & \text{for inclusion into the schedule} \\ 0, & \text{else} \end{cases} \tag{3.18}$$

Each variable indicates whether the corresponding action is scheduled or not. These actions coincide with directed edges and can be:

- $aStart$: arcs connecting a source node to all nodes corresponding to the first station of a train;
- $aEnd$: arcs connecting all nodes corresponding to the last station of a train to the sink node;
- $aRun$: arcs connecting the specific time and place where a run starts to the end time and place (i.e., these arcs define the route choice and the running time);

- *aDwell*: dwells in stations from specific arrival times to specific departure times;

- *aInfeasibility*: arcs connecting a node that does not coincide with the last station to the sink node.

Figure 3.5(a) shows a time-space grid for the macroscopic timetabling problem presented in Section 3.1 for APP, APP', ACP, PPP, and PCP. Figure 3.6(a) shows the routing trees for the sample microscopic timetabling problem in Figure 3.2. Source and sink nodes are depicted as black disks and some of the actions mentioned above are shown as grey edges. For each node v, let $\delta_+(v)$ and $\delta_-(v)$ be the outgoing and ingoing arcs respectively. As each train can be scheduled at most once, and the path should be continuous, the following constraints must be satisfied.

$$\sum_{a \in \delta_+(s_z)} x_a^z \leq 1 \quad \forall z \tag{3.19}$$

$$\sum_{a \in \delta_+(v)} x_a^z - \sum_{a \in \delta_-(v)} x_a^z = 0 \quad \forall v \notin \{s_z, t_z\}, \forall z \tag{3.20}$$

In APP and APP', both route and time continuity are ensured by (3.20).

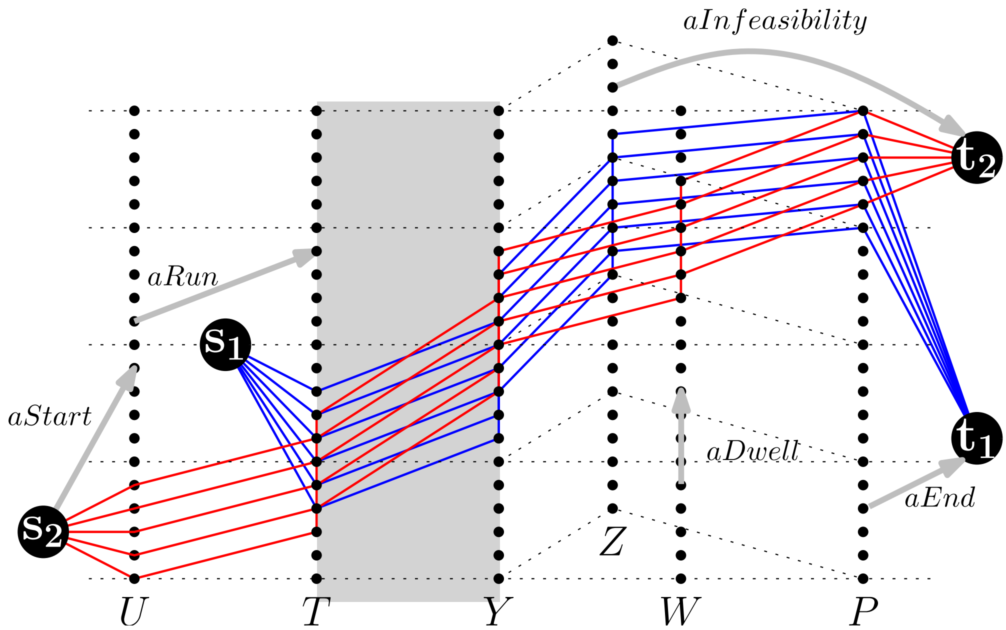
PPP, PCP, and RCG include a unique binary variable x_p^z for an entire chain of such actions from the beginning of a train path to the end. The left-hand side of Figure 3.5 depicts these variables as blue and red paths; the left-hand side of Figure 3.6 shows these variables as blue and red nodes. In these cases, continuity constraints (3.20) are implicitly assumed in the choice of the paths. The variables only have to satisfy the equivalent of (3.19):

$$x_=^z \begin{cases} 1, & \text{if the chain of actions } p \text{ of train } z \text{ is chosen} \\ & \text{for inclusion into the schedule} \\ 0, & \text{else} \end{cases} \tag{3.21}$$

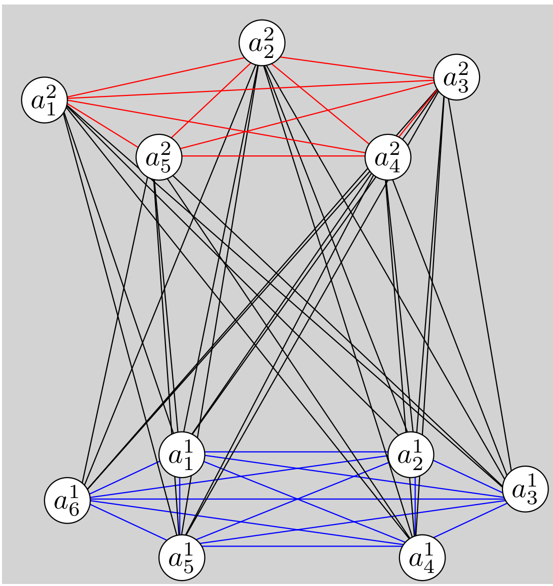
$$\sum_p x_p^z \leq 1 \quad \forall z \tag{3.22}$$

As the the stopping pattern influences the speed that is actually realisable, Fischer and Helmberg (2010) extend ACP to consider both the rolling stock and the stopping pattern for defining the realisable speed on a section. While the paths in Fuchsberger (2012)'s RCG formulation are associated with fixed speed profiles, Caimi (2009) proposed a RCG formulation for timetabling in regions with low traffic density which considers different speed profiles. If the number of block sections reserved by a train run

Figure 3.5: APP/APP'/ACP arcs and PPP/PCP model for the sample macroscopic timetabling problem in Figure 3.1.

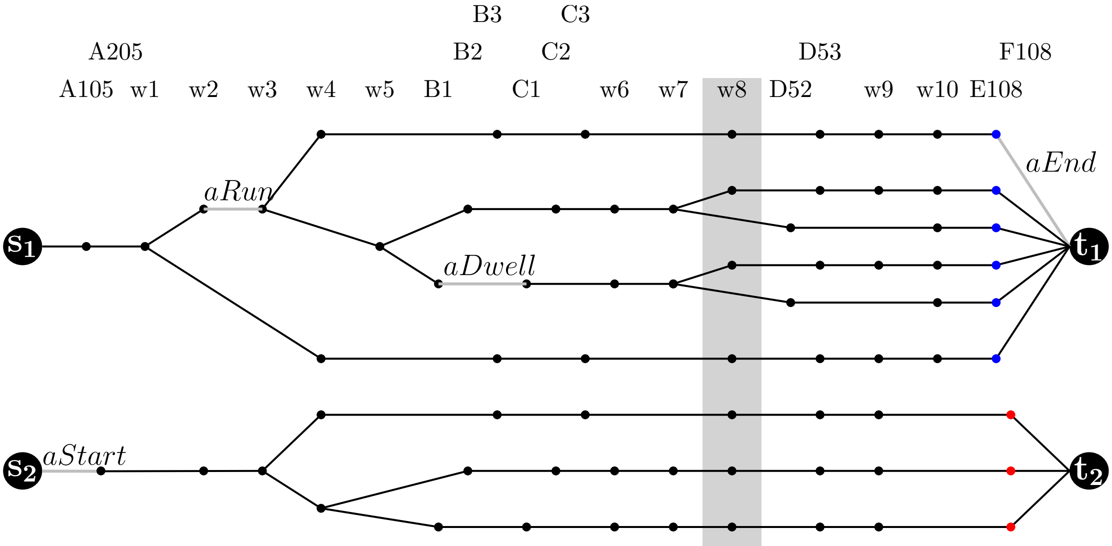


(a) The small nodes correspond to possible departure and arrival times at stations; the large nodes represent sources and sinks; the grey arcs represent the different types of constraints for APP/APP'/ACP; the blue and red paths correspond to possible runs for the first and the second train in PPP/PCP respectively.

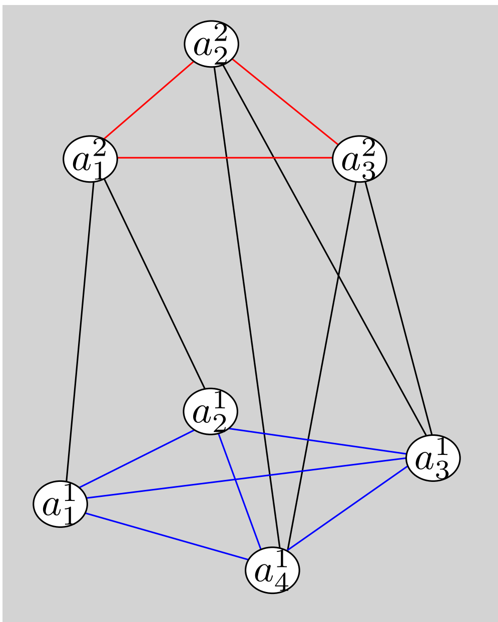


(b) The conflict graph between T and Y, corresponding to the section in the grey box in Figure 3.5(a).

Figure 3.6: RTCG/TCG tree and RCG model for the sample microscopic timetabling problem in Figure 3.2.



(a) The small nodes correspond to the route choice points in the topology of Ypslikon (Figure 3.2); the large nodes indicate the sources and sinks; the arcs correspond to the actions between the events in RTCG and TCG; the blue and red nodes correspond to the possible paths of the first and the second train in RCG, respectively.



(b) Conflict graph connected with resource $w8$. The nodes correspond to the nodes in Figure 3.6(a). Red and blue edges connect all nodes associated with one train, and black edges indicate overlapping blocking time intervals .

depends on the speed, this particularity is reflected in the blocking time stairways corresponding to the different profiles.

Similarly to the time-indexed formulations mentioned so far, REF-SRR contains binary variables $x^z_{i,j}$ indicating whether train z passes infrastructure nodes i and j sequentially. Route continuity and uniqueness are ensured by constraints (3.19) and (3.20). In addition, REF-SRR contains the following (binary) decision variables:

$$x^z_{i,j} = \begin{cases} 1, & \text{if train } z \text{ travels through infrastructure node } i \\ & \text{and then through infrastructure node } j \\ 0, & \text{else} \end{cases} \tag{3.23}$$

$$a^z_{i,j,t} = \begin{cases} 1, & \text{if train } z \text{ has arrived at infrastructure node } j \\ & \text{from infrastructure node } i \text{ by time } t \\ 0, & \text{else} \end{cases} \tag{3.24}$$

$$d^z_{i,j,t} = \begin{cases} 1, & \text{if train } z \text{ has departed from infrastructure node } i \\ & \text{towards infrastructure node } i \text{ by time } t \\ 0, & \text{else} \end{cases} \tag{3.25}$$

$$y^z_{i,j,t} = \begin{cases} 1, & \text{if train } z \text{ is between infrastructure node } j \\ & \text{and infrastructure node } i \text{ at time } t \\ 0, & \text{else} \end{cases} \tag{3.26}$$

As the decision variables correspond to the arrival and departure times and to the route choices, this model includes constraints modelling the time-space interdependencies for each train. The relations between time variables $a^z_{i,j,t}$, $d^z_{i,j,t}$ correspond to running and dwell times and are modelled via constraints of type (3.2) similarly as in continuous-time formulations. Passing constraints (cfr. time-indexed formulations) prevent departures ahead of schedule.

Cap-DM models single train runs in the same way as ESP (see previous section). The only difference is that in Cap-DM the decision variables are integer.

Modelling interactions Interactions between trains in time-indexed formulations usually correspond to infrastructure allocation conflicts. If a line between two stations that can host at most one train at time or a single infrastructure resource is considered, then a conflict occurs if the blocking time intervals of different trains overlap. If x^z_a and x^w_b correspond to conflicting allocations of an infrastructure resource r, APP' and TCG prevent the simultaneous allocation through

$$x_a^z + x_b^w \leq 1 \tag{3.27}$$

APP, PPP, RTCG, and RCG model conflicting allocations using conflict graphs. For a given infrastructure resource r, a conflict graph contains a node for each alternative train movement that requires that resource (i.e., for each edge in figures 3.5(a) and 3.6(a) connected with that resource). Two nodes are linked together if either they correspond to the same train, or their blocking time intervals of the resource overlap (see, e.g., Herrmann, 2006, for further information about conflict graphs). Let C_r be the set of maximal cliques of the conflict graph for the infrastructure resource r. Then, the following constraints prevent conflicts:

$$(APP, APP', RTCG) \sum_{(z,a)\in C} x_a^z \leq 1 \quad \forall C \in C_r, r \tag{3.28}$$

$$(PPP, RCG) \sum_{(z,p)\cap C\neq\emptyset} x_p^z \leq 1 \quad \forall C \in C_r, r \tag{3.29}$$

In Lusby et al. (2013)'s version of RCG, train routes that block a resource but do not occupy it physically (e.g., the unused branch of a switch) receive coefficient of 0.5 in (3.29). The right-hand side of Figure 3.5 shows the conflict graph for the sample macroscopic timetabling problem presented in Section 3.1, assuming that both trains use the same track between Testadt and Ypslikon. The nodes correspond to the edges between Testadt and Ypslikon highlighted in the grey box on the left-hand side. The red and blue edges of the conflict graph connect all nodes corresponding to one train, and the black edges coincide with conflicts. The right-hand side of Figure 3.6 shows the conflict graph for resource $w8$ (highlighted in the left-hand side) in the sample microscopic timetabling problem presented in Section 3.1. Train length is implicitly considered by TCG, RTCG, and RCG with the size of the resource allocation interval. Also REF-SRR prevents conflicting allocations using a sort of conflict clique. Conflict-free operations on track (i, j) are modelled as

$$\sum_{z:z \text{ passes } (i,j)} y_{i,j,t}^z + \sum_{z:z \text{ passes } (j,i)} y^z j, i, t \leq Cap(i, j, t) \quad \forall t \tag{3.30}$$

where $Cap(i, j, t)$ is the capacity of (i, j) at time t.

Fischer et al. (2008) include capacity constraints within stations in PPP. These constraints indicate how many trains can be hosted and how many movements can be assigned to each direction during the considered time. Fuchsberger (2012)'s RCG model contains variables indicating whether connections have to be forced or broken during operations. ACP and PCP are extensions of APP and PPP respectively. Instead of using conflict graphs directly, ACP and PCP use them to identify configurations. A configuration q is a set of trips on a track r that are not conflicting with each other. Thus, it satisfies

$$|q \cap C| \leq 1 \quad \forall C \in C_r \tag{3.31}$$

A configuration q for a track r is a path from an artificial source node s_r to an artificial sink node t_r. The edges on the path either correspond to runs on the track r or connect s_j to departure nodes, t_j to arrival nodes, or an arrival to possible departures of follow-up runs. For each such edge e, let y_e be a binary variable, indicating whether the edge is part of the configuration. $\delta_+(v)$ and $\delta_-(v)$ denote the outgoing and ingoing arcs for each node v of this graph. A conflict-free schedule is obtained by choosing a unique continuous path in this graph and limiting the choice of actions a in APP to the ones that are contained in the configuration path. This corresponds to substituting (3.28) with the following constraints, and the obtained model is ACP:

$$y_e = \begin{cases} 1, & \text{if the edge } e \text{ of a configuration is chosen} \\ & \text{for inclusion into the schedule} \\ 0, & \text{else} \end{cases} \tag{3.32}$$

$$\sum_{e \in \delta_+(v)} y_e - \sum_{e \in \delta_-(v)} y_e = 0 \quad \forall e \notin \{s_r, t_r\}, \forall r \tag{3.33}$$

$$\sum_{e \in \delta_+(s_r)} y_e \leq 1 \quad \forall r \tag{3.34}$$

$$x_a - y_a \leq 1 \quad \forall a \tag{3.35}$$

For each configuration q, PCP includes a unique binary variable y_q indicating whether q is assigned in the schedule. Thus, continuity constraints (3.33) are no longer needed. Let Q_r be the set of configurations on track r. PCP is obtained by substituting (3.29) with the following constraints:

$$y_q = \begin{cases} 1, & \text{if the configuration } q \text{ is chosen} \\ & \text{for inclusion into the schedule} \\ 0, & \text{else} \end{cases} \tag{3.36}$$

$$\sum_{q \in Q_r} y_q \leq 1 \quad \forall r \tag{3.37}$$

$$\sum_{a \in p} x_p - \sum_{a \in q} y_q \leq 0 \quad \forall a \tag{3.38}$$

Interactions in Cap-DM are passenger connections and headways. These are modelled using disjunctive constraints (3.16). In Dollevoet et al. (2015)'s extension to track assignment, additional constraints ensure that station platform capacity is respected. These constraints yield to a similar result as (3.17).

Initialisation As for continuous-time formulations, while approaches for long-term timetabling need no initialisation, Fuchsberger (2012)'s and Meng and Zhou (2014)'s rescheduling approaches and Schöbel (2009)'s delay management problem need the initial position of the trains. Fuchsberger (2012) uses a rolling horizon framework, which fixes the blocking time stairways of train movements that have already started or are starting too early in future and uses predictions for the entrance times of trains that are approaching the controlled area. In their numerical experiments, Meng and Zhou (2014) applies a scenario-based rolling horizon solution approach.

3.1.3 Comparison of the different models

Two main categories of mathematical models of railway operations can be distinguished. The formulations in the first category model time with **continuous** variables and safety with minimal time differences between **pairs** of consecutive trains on each infrastructure resource. This category includes ESP, AG, FP and RR. Such models are often referred to as big-M formulations because of the disjunctive form of the conflict avoidance constrains. These constraints make the formulation weak, meaning that they slow down considerably the solution process of standard algorithms (Lamorgese et al., 2016).

The second category contains models based on **discrete** representation of time, modelling conflicts as **cliques** of a conflict graph. APP, ACP, RTCG, PPP, PCP, RCG, and REF-SRR are in this second category. While variables in APP, ACP, and RTCG represent single train actions, variables in PPP, PCP, and RCG model entire paths, and variables in REF-SRR correspond to either route choices or cumulative flow variables that count the trains that have departed from or arrived at some infrastructure points. Such models are often referred to as time-indexed formulations and are

Table 3.1: Models vs. functional requirements from Table 2.1): × means that the model satisfies the functional requirement; (×) means that the requirement is satisfied off-line (i.e., for timetabling); ○ means that there is a model extension which satisfies the requirement.

		continuous-time				time-indexed							
		ESP	AG	FP	RR	Cap-DM	APP'	TCG	APP/ ACP	RTCG	REF– SRR	PPP/ PCP	RCG
infra-structure	macroscopic	×	○	○		×	×		×			×	
	microscopic		×	×	×			×		×	×		×
rolling stock	length		×		×			×		×			×
	max. speed	×	×	×	×	×	×	×	×	×	×	×	×
	real. speed		○		○				○			×	×
operations	#blocks		○										×
	timetable		×	×	×	×					×		×
	closed tracks		×	×	×	○					×		×
	retiming	(×)	×	×	×	×	(×)	(×)	(×)	(×)	×	(×)	×
	reordering	(×)	×	×	×	×	(×)	(×)	(×)	(×)	×	(×)	×
	rerouting		○	×	×	○	(×)	(×)	(×)	(×)	×	(×)	×
	break connections		○			×							×
	cancel train		○	×	×		(×)	(×)	(×)	(×)	×	(×)	×
		pairwise conflicts							on tracks			on paths	
									conflict cliques				

thought to be stronger than big-M formulations but are usually much larger, which, again, results in a huge solving effort (Lamorgese et al., 2016). Some models cannot be classified in any of these categories because they rely on **discrete** representations of time and model conflicts **pairwise**. Examples of such models are Cap-DM, APP' and TCG (i.e., the weak versions of APP and RCTG).

Table 3.1 shows the results of the analysis of the models from Section 3.1 with respect to the functional requirements listed in Section 2.3.

Infrastructure ESP, Cap-DM, APP, ACP, APP', PPP, and PCP are based on a macroscopic topology, which is not suitable for modelling safety constraints of fixed block signalling systems precisely. Still, for each of these macroscopic models, there is a microscopic model that considers time and conflicts analogously. ESP, AG, FP and RR model time with continuous variables and require a minimum time separation between pairs of trains using the same infrastructure resource. Cap-DM, APP' and TCG model discrete time choices and require that at most one allocation from each pair of conflicting allocations of an infrastructure resource is assigned. The other models limit time choices to discrete sets and prevent (track occupation) conflicts using conflict graphs. APP, ACP, RTCG describe single activities (run, dwell); PPP, PCP, and RCG describe entire paths, and REF-SRR describe cumulative flows passed through infrastructure resources.

Rolling Stock All microscopic models but FP and REF-SER consider train length: AG by allowing longer alternative arcs (i.e., spanning over signal nodes), the others by assuming longer infrastructure occupation times. All models are able to describe train dynamics considering the maximum speeds permitted on tracks. For timetabling the stopping patterns are usually known in advance. Thus, the maximum speeds (that consider the planned stops) suffice to generate feasible schedules. If stopping patterns are not fixed in advance, it is necessary to consider the effect of stopping on the minimum feasible running time. In fact, if a train has to stop or it runs with limited speed on a track, it is not possible to enter the next contiguous track at maximum speed. As PPP, PCP, and RCG consider entire train paths, they represent realisable speed profiles, because the running times are computed during the preprocessing phase, after having chosen which paths to include into the model.

In the other models, the dependence of running times on stops has to be modelled within the optimisation problem. D'Ariano et al. (2007b) proposed an iterative approach that combines an AG with a second step updating the realisable speeds according to the solution found by the AG. Oettich and Caspar (2015) proposed to model the different speed profiles for each train with alternative arcs. Rodriguez (2007) inserted the explicit computation of speed profiles in a model similar to RR. Fischer and Helmberg (2010)'s ACP extension provides the actually realisable speed profiles as functions of rolling stock and stopping pattern. Note that RTCG computes conflicts using the realisable maximal speed profiles, but it does not consider lower speed profiles that may result by successive branching.

Operations Only microscopic models represent the blocks of the interlocking system and, thus, can consider the reservation of a number of blocks that depends on the speed of the train. In addition, as explained above, only few models are based on the actually realisable speed profiles. Consequently, basing on the literature analysed by this thesis, only the

RCG and D'Ariano et al. (2007b)'s and Oettich and Caspar (2015)'s AG extensions satisfy this requirement.

Models conceived for timetabling rather than rescheduling do not usually include representations of operations related features. In addition, these features are represented differently by the models that have already been applied to real-time operations (i.e., AG, FP, RR, Cap-DM, RCG, and REF-SRR).

First, the constraints given by the planned timetable may be included as a set of passing constraints (i.e,. inequalities imposing the departure times from stations to be greater to or equal to the scheduled times), as part of the objective function (i.e., terms that penalise early and late departures as well as too early or late arrivals and which usually depend on the amount of delay), or by considering them in the preprocessing phase when choosing the paths.

Second, FP, RR, RCG, and REF-SRR represent routing possibilities by binary variables that can be forced to zero to represent closed tracks. The classical AG contains no such variable and does not support rerouting. If a track is closed, trains are forced to pass it either before its closure or after its reopening time. D'Ariano et al. (2014)'s AG extension contains binary variables for routing possibilities. Thus, closed tracks can be represented as in the RR model. Similarly, the classical Cap-DM does not consider microscopic routes, but Dollevoet et al. (2015)'s extension allows rerouting within a station in a similar way as FP. Thus, the same logic can be applied to model closed platforms too.

Third, all models presented include representations of either intervention features or their off-line counterparts. Formulations for timetabling support timing, ordering, routing, and/or scheduling/not scheduling trains, which are the off-line counterparts of retiming, reordering, rerouting, and cancellation of trains. In models with continuous time variables and Cap-DM, (re)timing is achieved by changing the value of time variables. In the other models considering discrete time choices, (re)timing corresponds to a different choice of time-space combination. ESP, AG, FP, RR and Cap-DM formulate (re)ordering options as pairs of alternative arcs. The other formulations contain no such arc, and reordering is performed implicitly by choosing times and routes. All formulations based upon discrete times feature (re)routing. Also D'Ariano et al. (2014)' AG extension, RR, and FP feature rerouting. Breaking connections is featured only by Cap-DM, RCG and Corman et al. (2012a)'s AG extension. ESP, the classical AG and Cap-DM formulations aim at finding suitable times for previously selected discrete events. Thus, if a train cannot be scheduled because it conflicts with other services, the solution may either not exist or schedule the train after all other services. Train cancellation is represented by all other models as not scheduling the train. This coincides with all routing or path variables taking value zero. Note that Meng and Zhou (2014)'s REF-SRR formulation imposes that each train is scheduled exactly once, but this constraint can be relaxed to model that each train is scheduled at most once.

3.1.4 Model objectives

The most conventional objective of real-time timetabling and rescheduling approaches is to minimise some measure based on train delays. Thus, finding the most convenient solution from the operator's perspective. Two representative examples are: Yan and Yang (2012)'s real-time timetabling approach of freight services that minimises the operational costs, including costs associated with delays; and Samà et al. (2015)'s rescheduling approach (based on the alternative graph) evaluating multiple train delay based criteria (punctuality, (weighted) train delays, etc.) via Data Envelopment Analysis.

Recently, several approaches started considering customer perspectives and energy issues. In the remaining of this section, literature about real-time rescheduling is reviewed with respect to passenger expectations[4], freight traffic representation, and energy efficiency consideration.

Passengers [5] Several approaches to support operations have considered the customer perspective, particularly in the framework of delay management (e.g., Schöbel, 2009; Kanai et al., 2011; Dollevoet et al., 2012; Rückert et al., 2017). Rückert et al. (2017) develop a webtool that supports dispatchers with simulations of the consequences of waiting/non-waiting decisions on passengers. The the webtool is able to consider the entire German railway network but it does not consider stations' and tracks' capacity nor other operational requirements. In approaches aiming at optimising railway operations with respect to the passengers' delays, the consequences of delays on passengers are usually approximated by a weighted sum of dropped connections and train delays, which generally is a very accurate approximation for the sum of additional delays over all passengers (refer to Schöbel, 2007, for further details). Dollevoet et al. (2012) include passenger rerouting possibilities into the delay management problem by inserting the OD matrices and assuming that all

[4]Literature about timetabling will be neglected and the interested reader can refer to the recent work by Parbo et al. (2016).

[5]This paragraph is based on Toletti and Weidmann (2016)

passengers take the shortest path. Kanai et al. (2011) model passengers' disutility as a weighted sum of the on-board time, the waiting time at stations, the number of transfers, and the running time weighted by a congestion-related factor. The weighting parameters are set according to the results of a survey conducted in Japan.

Passenger satisfaction is very important for macroscopic rescheduling too. Binder et al. (2015) propose a passenger centric macroscopic rescheduling for severe disruptions with passengers rerouting. They model the problem as an IP based on a time-expanded graph including arc activities connected to both trains (depart, run dwell) and passengers (enter/leave the system, ride, wait, transfer). The objective is to minimise the combination of operating costs (expressed as the sum of running times for all services) and cumulated generalised travel times of passengers, which are a weighted sum of travel and waiting times, penalties for transfers and early/late departures. Tomii et al. (2005) propose the minimisation of passengers dissatisfaction as rescheduling objective. Passenger dissatisfaction is computed as a weighted sum of arrival and departure delays, prolonged dwell times, prolonged running times, the interval between consecutive trains of the same line, and missed connections. Sato et al. (2013) reschedule train operations minimising passenger discomfort resulting from disruptions, which is a weighted sum of running time, waiting time and transfers with respect to the planned trip. The model also includes a representation of passengers' decisional behaviour when facing schedule changes. Almodóvar and García-Ródenas (2013) provide a vehicle rescheduling problem to bring vehicles to lines experiencing extremely huge unexpected demand (e.g., in case of temporary unavailability of alternative transport systems). The problem is tackled using predictive simulation and on-line optimisation aiming at minimising the time of passengers in the system.

Given the local view of microscopic rescheduling, few works consider passengers explicitly. Fuchsberger (2012) models passenger dissatisfaction for the RCG model as a weighted sum of train cancellations, arrival and departure delays, and connections dropped. Corman et al. (2012a) and Espinosa-Aranda and García-Ródenas (2013) combine rescheduling and delay management via alternative graphs. Corman et al. (2012a) include decisions about connections as weighted terms in the objective function, while Espinosa-Aranda and García-Ródenas (2013) include estimations of origin-destination-matrices such that the objective function represents the total passenger delay.

Freight [6] Only few papers address freight train rescheduling. A possible reason for this lack is that in some countries there is no planned freight schedule (Pachl, 2015). Thus, real-time freight train management is actually a timetabling task that is executed up to a day prior the concerned train movements (see, e.g., Murali et al., 2016). This is not the case in Switzerland, where the yearly timetable includes a large number of slots for freight trains. Some slots are allocated to trains and some remain unallocated and can be used by trains that are too early or too late for their assigned slot. Kraay and Harker (1995) propose a macroscopic real-time timetabling model which minimises the operational costs, represented by a convex function of the departure and arrival delays, the penalty for extending a crew shift beyond the legal amount of hours and a penalty for blocks missing a connection with another train.

Another reason why freight rescheduling has received less attention is that freight trains usually have different punctuality targets from passenger trains. In Europe, they usually have arrival time slots rather than precise arrival times. Consequently, in the case of a conflict, they usually have lower priorities than passenger trains. With a similar perspective, Davydov et al. (2014) state that the local delays of freight trains are not as significant as the ones of passenger trains, and freight train rescheduling should focus on energy efficient operations. However, some freight trains (e.g., transporting mail, just-in-time delivery) already share the same priority of passenger trains in dispatching. It is reasonable to assume that with the liberalisation of the railway sector, freight train operating companies will expect more transparency in conflict resolution, which implies that freight train rescheduling objectives have to be modelled explicitly.

Due to this lack of references, an empirical study about current practices in freight train rescheduling was carried out by Ulmke (2017). Based on interviews with both the infrastructure manager (SBB) and the freight train operator (SBB Cargo), Ulmke (2017) concluded that objective function for mixed traffic rescheduling is best represented by monetary costs and should consider energy consumption, the operating costs of the railway undertaking, the value of time of passenger delays and freight train delays.

Energy efficiency[7] The main factors affecting energy-efficient driving strategies are speed uniformity and loss of kinetic energy caused by braking (Bai et al., 2009). Most of the literature on traction energy reduction focuses on speed profile optimisation considering the single train as an

[6]This paragraph is extracted from Toletti et al. (2017).

[7]This paragraph is adapted from Toletti et al. (2016)

isolated system. The problem has been approached within control theory by considering the traction force as the control variable under simplified conditions and different control cases (Howlett, 2000; Khmelnitsky, 2000). Further improvements have been made on multi-stage optimisation to include track variability (Franke et al., 2000; Balmelli, 2010), and analytic solutions for the sequence of optimal control (Liu and Golovicher, 2003; Albrecht et al., 2015).

Several authors use speed profile parameters instead of traction efforts as control variables (Aradi et al., 2013; De Martinis et al., 2014). The model by Aradi et al. (2013) assumes that the arrival time is no longer fixed but rather that the difference between the actual and planned time is a term in the objective function that needs to be minimised. The model by De Martinis et al. (2014) takes a supply design modelling approach and uses a simulation based framework to define the amount of additional travel time that can be spent to reduce energy consumption. Then, simulation is used to check the effects of speed profile modification on railway traffic.

Only few works explicitly consider energy efficiency during rescheduling and most of them focus on to metro lines, where traffic is not mixed (e.g., Su et al., 2015; Li and Lo, 2014). One of the few approaches for mixed traffic has been presented by Corman et al. (2009): it implements the green wave policy with fixed speed profiles in AG and evaluates energy consumption reduction as an effect of conflict resolution. Another approach focuses on the modification of speed profiles during operation when small disturbances occur (D'Ariano and Albrecht, 2006; Mehta et al., 2010). It is based on forecasting conflicts and then modifying the speed profiles (sequence of instructions for drivers) towards conflict-free trajectories. Implementing this method requires an overview of rail traffic together with the length, position, speed and ongoing acceleration/deceleration of all trains. In particular, a case study for the Loetschberg tunnel in Switzerland has shown a 12% reduction of energy consumption compared to the non-optimised trajectories (Mehta et al., 2010).

3.2 Solution methods

Given a mathematical optimisation problem, like the models described in the previous section, the task is to find the values for the decision variables which give the best value for the objective and satisfy all constraints. Several methods to solve mathematical optimisation problems and obtain one or more solutions exist. The suitability of these methods for a particular formulation usually depends on the problem's size, the time and space available for computation, and the variables being linear, integer or mixed. All the models described in the previous section that feature rerouting contain some integer or binary variables. Exact methods that aim at finding an optimal solution of (mixed) integer linear programs are usually unsuitable for solving large problems in short times (e.g., branch and bound (BB), cutting planes, and dynamic programming see, e.g., Galli, 2017). Relaxation approaches and (meta-)heuristics put more emphasis on the computation time but cannot ensure optimality.

Literature on solution methods to optimisation problems is very vast and spans several decades with the first works having been redacted during World War II and new advances being continuously published. As a consequence, nowadays, the researcher can decide to use commercial software and open source libraries along with own implemented algorithms. This section summarises the most relevant solution methods for the scope of this thesis. In particular, algorithms and tools applied to railway timetabling and rescheduling problems are reviewed. The goal is to identify promising methods to exploit and further develop in this thesis.

3.2.1 State-of-the-art commercial and open-source software

Many of the models for timetabling and rescheduling presented in the previous section are solved via commercial MILP/LP/IP solvers such as CPLEX (IBM, 2015) or Gurobi (Gurobi Optimization, 2016). Such standard solution approaches are often combined with ad-hoc preprocessing routines to improve the algorithm's efficiency in terms of computation time and solution quality. Caimi (2009), Caimi et al. (2011, 2012), and Fuchsberger (2012) use CPLEX (with built-in preprocessing) for timetabling and rescheduling using Conflict Graph based formulations. Borndörfer and Schlechte (2007) solve LP relaxations of APP' and ACP using CPLEX. Yan and Yang (2012) combine CPLEX with variable fixing (e.g., forbidding very unlikely takeovers). Similarly, Törnquist and Persson (2007) limit the rescheduling options (e.g., "only track change and no reordering", or "at most x reorderings"). Peeters and Kroon (2001) remove redundancies before solving the problem using CPLEX. Pellegrini et al. (2014) propose a two-steps-procedure: first the rescheduling problem is solved with no rerouting possibility, then the solution is used as a starting value if it is found that a rerouting is necessary. Lusby et al. (2013) embed CPLEX in a variable generation scheme where pricing is achieved via enumeration of time-space extensions and a recursive tree transversal algorithm. Kroon et al. (2009) use CPLEX for microscopic routing but the upper level is solved by constraint programming. Herrigel et al. (2013a) use CPLEX within a hierarchical decomposition approach based on train classes to solve PESP. Kersbergen et al. (2016) use Gurobi to reschedule in a decomposed Model Predictive Control (MPC) setting.

Other researchers use open source libraries such as GNU linear programming kit, used by Schöbel (2007) for solving the delay management problem. Another promising open source library for the current work is LEMON (Dezsõ et al., 2011), which implements data structures and algorithms for combinatorial optimisation. Although to the author's knowledge it has not been applied to real-time railway traffic management yet, LEMON has been used for similar problems such as network expansion planning (Bärmann, 2016, combined with Gurobi) and subway passenger flow assignment Stasko et al. (2016).

Timetabling and rescheduling problems are usually too large to be solved with a simple application of commercial or open source software. Nevertheless, such tools can be very useful if embedded into more elaborate routines (see, e.g., Lusby et al., 2013; Kroon et al., 2009; Herrigel et al., 2013a). The next sections review some concepts that help coping with the problem size.

3.2.2 Column generation and Lagrangian relaxation

Column generation and Lagrangian relaxation are known methods for solving large-scale linear programming problems. Barnhart et al. (1998), Desaulniers et al. (2005), and Lübbecke and Desrosiers (2005) explain the theoretical and algorithmic features of column generation approaches with examples from the literature. The three works highlight the interrelations between column generation and Lagrangian relaxation:

1. The Lagrangian dual of a MIP corresponds to the optimal value of the LP relaxation of its column generation form obtained by Dantzig-Wolfe decomposition;

2. the Lagrangian relaxation of a MIP has the same form as the column generation sub-problem of the LP-relaxation of its column generation form.

The phases of column generation are (adapted from Barnhart et al., 1998):

Step 1 Determine an initial feasible restricted master problem.

Step 2 Solve the current restricted master problem.

Step 3 Delete nonbasic columns with high negative reduced costs from the restricted master problem.

Step 4 Solve the pricing problem(s) to prove optimality or generate one or more columns with positive reduced costs. If columns are generated, add them to the restricted master problem and go to 2.

Step 5 Stop.

A variant of column generation called branch and price is commonly applied to (mixed) integer linear programs: If the solution after step 4 does not satisfy integrality conditions, branching (i.e., fixing the value of an integer variable) is applied and the algorithm goes back to step 2.

Some works have focused on applying column generation and Lagrangian relaxation to railway timetabling or rescheduling problems. Brännlund et al. (1998) use Lagrangian relaxation for decomposing the railway timetabling problem into train-specific sub-problems for column generation. Each column corresponds to a feasible schedule for a train and the use of capacity is penalised by the Lagrangian multipliers in the objective function of the master problem. Caprara et al. (2002) develop a heuristic based on Lagrangian relaxation for timetabling. Cacchiani et al. (2008) apply column generation to the same formulation. The columns correspond to feasible schedules for each train. In a following work, Cacchiani et al. (2010) apply Lagrangian relaxation for inserting additional freight trains into a schedule where the timetables of passenger trains cannot be changed. Since it is preferable to have as few dualised constraints as possible, Meng and Zhou (2014) propose to dualise a subset of capacity constraints of the Simultaneous Retiming and Rerouting formulation (the ones that have not been used by trains so far). The Lagrangian multipliers can be interpreted as the (time-dependent) costs for using those resources and the train specific sub-problems consist in finding the least cost path from origin to destination.

Borndörfer and Schlechte (2007) solve the LP-relaxation of the reduced master problem of PCP using CPLEX and apply a rounding heuristic, while the longest path is used for pricing. Fischer et al. (2008) solve the train timetabling problem using ConicBundle library (Helmberg, 2005) and CPLEX on the Lagrangian relaxation. Basing on the observation that most trains only use the "earlier" partition of their time-expanded graphs, Fischer and Helmberg (2010); Fischer (2013) combine Lagrangian relaxation and column generation for timetabling. The innovation of this approach is that it reduces the memory required by the sub-problems.

The time-expanded network that is used to build the columns is restricted to the partition that was interesting for previous computations and it is enlarged only if needed.

Min et al. (2011) use column generation to coordinate rescheduling over a large metropolitan area. Each column of the master problem corresponds to a feasible schedule in a segment and is obtained via heuristics. Lusby et al. (2013) apply a column generation inspired approach for real-time rescheduling. The columns correspond to paths of trains through a junction and are obtained using a tree-based algorithm instead of defining and solving a proper optimisation problem. Foglietta et al. (2014) apply a column and row generation approach for solving an AG formulation with routing choices. This approach starts with a restricted master problem that contains only the time variables and the constraints connected with the movements of the individual trains and adds variables and constraints modelling interactions between trains as needed until the problem is solved.

Column generation and Lagrangian relaxation are not only well understood from the theoretical point of view, but they have been applied to many industries (Desaulniers et al., 2005) . Degraeve and Schrage (1997) developed a column generation approach for tire production scheduling. Huisman et al. (2005) describe the advantages of combining column generation and Lagrangian relaxation for the integrated vehicle and crew scheduling in bus transport. In this approach, each column consists in a set of duties. Bélanger et al. (2006) apply column generation for the periodic airline fleet assignment problem. In this case, the columns correspond to feasible aircraft itineraries. Westphal and Krumke (2008) apply column generation for the vehicle routing problem associated with dispatching for the German automobile club. In this application, each column represents a feasible tour for a vehicle.

3.2.3 Heuristics and metaheuristics

Heuristics are usually very fast solution methods that often provide good quality solutions, but whose optimality cannot be proven. Several heuristic algorithms have been developed for continuous-time formulations. These are often based on local search or consists of a problem-specific modification of the branch and bound method (BB). Serafini and Ukovich (1989) apply a refined search tree algorithm to solve PESP. Mascis and Pacciarelli (2002) and Mazzarello and Ottaviani (2007) solve AG with a BB method based on greedy algorithms. D'Ariano et al. (2007a) propose an algorithm to solve AG in which it forbids BB to select alternative pairs of arcs that would result in higher delays. The same strategy is applied by D'Ariano et al. (2007b, 2014, 2008); Corman et al. (2010b, 2012a); Kecman et al. (2013). D'Ariano et al. (2008) combine it with a local search for routing. Corman et al. (2012a) combine it with a Pareto local search for connections. Zhou and Zhong (2007) apply BB with lower bound rules. Tamannaei et al. (2016) speed up a BB method to rescheduling (AG formulation) using fathoming rules based on the lower bound and setting an upper bound to the computation time based on the number of block sections covered by the considered sub-problem.

Heuristic approaches have also been applied to speed up the resolution of timetabling problems formulated as constraint programming or Boolean satisfiability problems (SAT). Constraint programming has long been considered the most time-efficient way to solve some NP-hard problem such as PESP, but more recent results have shown that these might be outperformed by newer SAT solvers (Großmann et al., 2012). Nevertheless, real world problems are often too large even for such methods, which are then integrated by heuristics. Rodriguez (2007) applies a search tree with BB for solving the constraint programming formulation for real-time timetabling. Kümmling et al. (2015) adopt a hierarchical decomposition approach for network-wide SAT-based timetabling which is combined with local conflict search and corridor analysis to improve the solution of the decomposed problem.

Metaheuristics are enhanced heuristics with special techniques to avoid getting stuck in local optima or in loops (Galli, 2017). Corman et al. (2010b) and D'Ariano et al. (2014) complete D'Ariano et al. (2007a)'s BB algorithm with a tabu search for routing. The resulting algorithm is much faster than the analogous problem formulated as mixed-integer linear programming (MILP) and solved via CPLEX in several numerical experiments proposed by D'Ariano et al. (2014). Samà et al. (2017) propose a variable neighbourhood search algorithm to explore the alternative routes for an AG formulation time-efficiently. Herrmann (2006) applies a fixed point iteration method to route trains and a random restart local search to increase timetable stability in stations. Samà et al. (2016) improve the solution time of Pellegrini et al. (2014)'s formulation by selecting the routes to be used by each train with a "real-time Routing Selection Problem" solved via ant colony optimisation. In addition, there are many approaches which transform train schedules into binary strings and apply Genetic Algorithms to maximise some quality measure (see, e.g., Dündar and Şahin, 2013; Gholami and Sotskov, 2012).

3.2.4 Comparison of the different solution methods

Commercial solvers are usually a good option if the problem size is limited. Successful preprocessing strategies, heuristics based on local search and/or branch and bound, Lagrangian relaxation and column generation frameworks are motivated by some characteristics of the model representation of the problem. Column generation yielded good schedules with respect to the stated objective when applied to models with many integer variables within short computation times. This because it enables the number of alternatives in the model, which is the main source of complexity in time-indexed formulations, to be reduced. Several heuristics and metaheuristics could solve continuous-time formulations within short computation times and find near-to-optimal schedules with respect to the objective functions of the models. The reason is that they reduce (or remove) the variables involved in Big-M constraints which usually slow down the convergence of otherwise fast branch and bound algorithms.

This thesis investigates different combinations of commercial solver with column generation, relaxation and heuristic techniques trying to exploit the aforementioned strengths of each approach. The main goal is to define a solution method which is able to find a feasible schedule with a satisfactory value of the objective function of the mathematical model within the time available for taking decisions about departure, passing, and arrival times, connections, and speed advisory as prescribed by the functional requirements in Table 2.1.

3.3 Partitioning and coordination approaches

Large and complex problems such as traffic management are hardly treatable by a fully centralised approach because of both the large computational effort required and the low acceptance from dispatchers and train controllers. In fact, even state-of-the-art computers might not be able to solve network-wide centralised traffic management problems or need long times that are incompatible with real-time operations. In the long term perspective, this problem may be solved by more powerful computers. However, the energy required by high performance computers is very high (see, e.g., the power consumption of the world top machines, `top500.org`), and the energy issue is not expected to become less important in future. Moreover, Brüngger et al. (2014) highlighted the importance that users of automated systems can understand and overview what automatic functions do, otherwise they will try to avoid using them. This implies that a system reflecting the decentralised traffic control structure of many railway infrastructure managers might be adopted more easily than a centralised one. For these reasons, automatic solutions have usually been developed to control single junctions, stations, or lines (see, e.g., Corman and Meng, 2013, for a literature review; and Mehta et al., 2010, and Oettich and Caspar, 2015 for real-world applications). The same trend has been observed in the literature on control of automated road vehicles, where most approaches address single isolated intersections (see, e.g., Li et al., 2014).

To the author's knowledge, no commonly-agreed decomposition and coordination approach for real-time railway traffic rescheduling exists. Coordinating automated rescheduling in different areas is not only relevant for future applications of the current research but also for facilitating the adoption of automated traffic management support systems by practitioners. In fact, the number of critical spots in the Swiss railway network is doomed to increase with traffic density. Thus, it cannot be excluded that two (or more) of the above mentioned automated systems will work close enough to directly affect each other.

To extend the size of control areas, different partitioning and coordination approaches are often considered (Rinaldi and Tampère, 2015):

- *decomposition* divides the problem into simpler subproblems and solve them using centralised computation while maintaining the original problem dynamics;
- *distribution* solves the subproblems separately and ensures the original problem dynamics using a central coordinator, if necessary;
- *decentralisation* solves the subproblems separately with no explicit coordinating mechanism.

The following sections review and compare decomposition, distribution and decentralisation approaches for railway operations.

3.3.1 Decomposition schemes

The size of a centralised railway rescheduling problem has been considered problematic by many researchers who proposed to decompose it into smaller instances that are easier to solve. In many cases, finding a feasible solution is delegated to trains and infrastructure partitions and negotiation approaches are applied for coordination. However, in some cases, the decomposition is not geographical. Herrigel et al. (2013a), for instance, propose a hierarchical decomposition approach for timetabling in which the different classes of trains are added to the timetable stepwise starting with the ones with higher priority.

Mazzarello and Ottaviani (2007) identify the need for decomposing large alternative graph formulations into sub-problems of tractable size that are coordinated by an upper level that checks the feasibility of the global solution. In this framework, Corman et al. (2010a, 2012b, 2014) coordinate rescheduling using a so-called *border-graph* (see, e.g., Strotmann, 2007). Corman et al. (2010a) emphasise the need to consider not only the geography but also the amount of traffic to produce instances of tractable size, but do not outline any more precise strategy to network partitioning. Dollevoet et al. (2014) alternate delay management and station rescheduling iteratively for network-wide control. Note that in this case the areas at the microscopic level are disjoint. Caimi (2009) formalises the network partitioning originally proposed by Laube et al. (2007)and applies it to timetabling. Again, the network partitioning depends on geography and traffic density (cfr. Salido et al., 2007; Corman et al., 2010a). *Condensation zones* are the bottlenecks of the network, the congested infrastructures according to UIC-code 406 (2004). There, trains are scheduled at maximum speed to reduce the resources' occupation times. *Compensation zones* coincide with regions with lower traffic density. There, time reserves for delay absorption are planned. It is assumed that each two condensation zones are separated by a condensation zone. This network partitioning is adopted by Fuchsberger (2012) for a Model Predictive Control approach for rescheduling in condensation areas.

Other approaches solve the burden of centralised decision making through rule based decision making and Multi-Agent Systems. Jia and Zhang (1994) propose a hierarchically distributed fuzzy decision-making system that mimics the decisions of human dispatchers via IF-THEN rules. An upper level takes the strategic decisions, while the operational decisions are taken by local controllers in real time. Proença and Oliveira (2005) decompose the traffic management problem via a Multi-Agent System, in which agents correspond to trains (speed control), stations (platform management and train orderings) and a supervisor (which ensures safety). In addition, a learning module adapts the rules based on past decisions. Narayanaswami and Rangaraj (2015) propose to reschedule rail traffic on a single track line using Multi-Agent Systems (MAS). A supervisor agent detects conflicts and informs an auctioneer, which asks the affected train agents to place their bids and declares the winner. The look-ahead ranking bid ensures deadlock-avoidance. Then, supervisors determine the departure times of the conflict trains and reschedule the rest of the trains using a MILP and station agents allocate the needed resources and check the feasibility of the solution.

3.3.2 Distribution schemes

Salido et al. (2007) model the train timetabling problem as a Constraint Satisfaction and Optimisation problem. They show that the distributed modelling can be solved much faster than a centralised one and that a partitioning which considers the problem properties outperforms ”domain-independent” problem partitioning, which only aims at generating sub-problems of equal size (e.g., generated by a commercial graph partitioning software). Two strategies for partitioning were proposed: The first groups the variables depending on the train type; the second divides the network with respect to geographic regions. In the second case, to balance the solution effort and increase the success rate, partitions containing bottlenecks are smaller and have higher priority during the solution process. Partial states are passed between the agents solving the sub-problems until a compatible assignment is reached. The two strategies result in similar solution times.

Kersbergen et al. (2014) and Kersbergen et al. (2016) apply a Distributed Model Predictive Control for macroscopic rescheduling at network-wide level. In this case, no higher-level supervision exists, but the local instances are also responsible for coordination with neighbours. The constraints of the MILP model (based on Max-Plus algebra, but similar to macroscopic AG, refer to van den Boom et al., 2012, for further details) are rearranged into smaller matrices which refer to vectors of decision variables that are as independent as possible from each other. Kersbergen et al. (2014) proposes two approaches to solve the model. The first solves it sequentially by re-optimising with respect to one vector of decision variables. The second iteratively optimises the sub-problems consisting of small matrices and independent vectors of decision variables separately and joins the solutions until convergence is reached. All test cases show convergence. Kersbergen et al. (2016) compare the performance, in terms of objective value and computation time, of different decomposition strategies (local/global reorder, local/global retime) and coordination settings (fixed/traffic-dependent weights at partition boundaries). They highlight that the most efficient strategy consists of limiting the decisions in each subproblem (i.e., local reorder and retime) and set weights at boundaries that depend on the outbound traffic volumes.

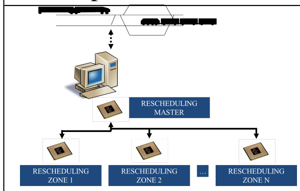
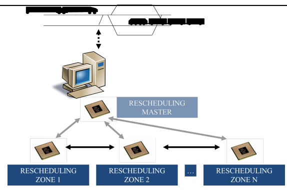
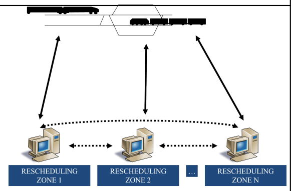
Tamannaei et al. (2016) consider a double track line where a block section is disrupted such that all trains must circulate on a single track. The authors partition the problem into subproblems covering either the only available track in the disrupted area or the one-directional tracks outside. The problems are then solved sequentially according to trains' travel directions.

3.3.3 Decentralisation schemes

Vernazza and Zunino (1990) propose a decentralised intelligence approach for assigning tracks between nodes, in which each two (or more) neighbouring nodes negotiate assignments of the track in-between according to urgency for decision, train priorities and node overcrowding. Parodi et al. (1996) complete this approach including deadlock-avoidance algorithms. Iyer and Ghosh (1995) highlight the advantages of decentralised traffic management, in that the need for computation time and memory in centralised systems increases non linearly with the number of trains, stations and tracks. They propose that each "natural entity" (e.g., trains, stations) contributes to conflict resolution by negotiating with other entities and taking its own decisions. Train on-board processors compute suitable paths through stations, and station units optimise the track allocations by accepting or denying the trains' requests for routes. The coordination works through continuous message passing. Lee and Ghosh (1998) extend the look-ahead capability of this approach and introduce the concept of "soft-reservation": A station can accept the request of a train to use a track either for the requested interval or for the first interval available. The tracks between stations are controlled by one of the neighbouring stations. Parkes and Ungar (2001) reformulate the auction's winner determination problem for track allocation in stations as a MILP and determine the bidding prices using dynamic programming.

Lamma et al. (1997) partition the railway network in equal modules, each one controlling one station and some (contiguous) branches. Coordination is ensured by message passing. The authors highlight the difference between timetabling in stations and on branches. This intuitive decomposition has been implicitly assumed by many more recent timetabling and rescheduling methods. DONS, for instance, contains a module for network-wide timetabling and one for microscopic timetabling in station areas (Hooghiemstra et al., 1999). It is implicitly assumed that no refinement of the macroscopic schedule is necessary outside stations. Chou et al. (2009) propose a collaborative rescheduling approach, where each agent is responsible for rescheduling traffic on a junction and can choose any strategy independently from the others. Coordination between

Table 3.2: Problem partitioning schemes

	decomposition	distribution	decentralisation
			
literature	Caimi (2009); Corman et al. (2010a, 2012b, 2014); Dollevoet et al. (2014); Herrigel et al. (2013a); Jia and Zhang (1994); Narayanaswami and Rangaraj (2015); Proença and Oliveira (2005)	Kersbergen et al. (2014, 2016); Salido et al. (2007); Tamannaei et al. (2016)	Chou et al. (2009); Iyer and Ghosh (1995); Lamma et al. (1997); Lee and Ghosh (1998); Parkes and Ungar (2001); Parodi et al. (1996); Vernazza and Zunino (1990)
communication	always through master; shared memory (no message passing needed);	through master or peer-to-peer; shared or distributed memory possible;	peer-to-peer; distributed memory (message passing needed);
higher level coordination (e.g., prediction, delay management)	master	master	should be embedded in one peer
scalability	limited	high	highest
fault tolerance	zero	limited	high

a subset of junctions is achieved via negotiations. Each junction computes its own optimal solution given initial boundary conditions, then the interface-coordination method checks whether the local schedules of adjacent areas are compatible, if not, new local schedules are computed, until a common solution is found.

3.3.4 Comparison of problem partitioning approaches

Rinaldi and Tampère (2015) point out that problem partitioning approaches are usually classifiable into three categories: decomposition, distribution and decentralised schemes. Examples of each scheme exist in railway traffic management literature. Table 3.2 summarises the literature presented in the previous sections and the strengths and weaknesses of the different paritioning schemes.

Decentralisation schemes have the highest scalability and are usually more fault tolerant (Lamma et al., 1997). However, they are characterised by more complex communication protocols because the computing nodes do not have access to some commonly shared memory. Note that distribution schemes can be converted into decentralisation schemes by adjusting memory management and communication.

In almost all the reviewed approaches, partitioning considers geographic attributes. Salido et al. (2007) show that such partitioning yields better results than domain-independent ones. Furthermore, even the experiments by Törnquist and Persson (2007), where no partitioning approach was applied, indicate that the region where the disturbance occurs is more important than the magnitude of the corresponding initial delay. In addition, a geographic partitioning is very close to current traffic management processes at many railway infrastructure managers. Thus, geographic partitioning is expected to have higher acceptance in practice (cfr. Brüngger et al., 2014).

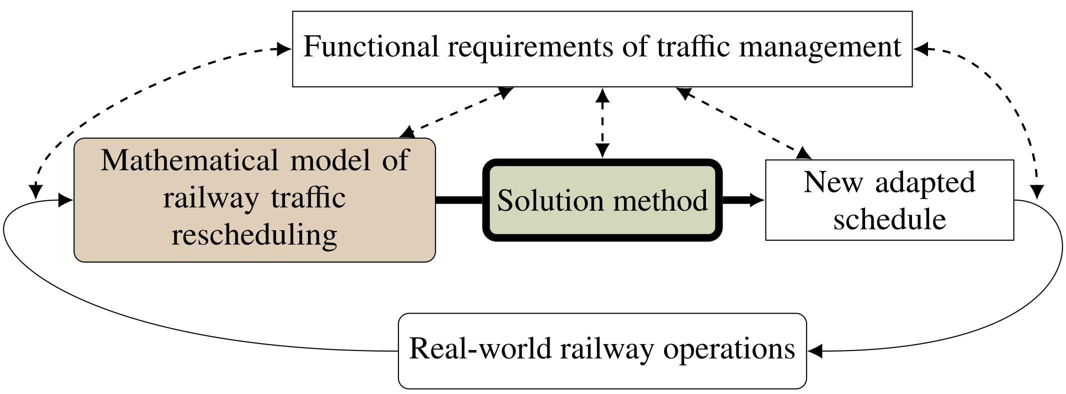
Chapter 4

Coordinated rescheduling over multiple zones

This chapter introduces a methodology for railway traffic rescheduling over multiple zones. Figure 4.1 shows the general idea. First, the railway traffic rescheduling problem is translated into a mathematical model. Then, a solution method is applied to solve the mathematical model and obtain a new schedule. The satisfaction of the functional requirements from Table 2.1 is a sine qua non for all elements of this methodology, including the initialization of the mathematical model from the current traffic situation and the communication of the new schedule to the interlocking system, to the trains and to customers.

The chapter is organised as follows. Section 4.1 defines the problem partitioning and the types of zones. Section 4.2 describes the mathematical representation of the local traffic rescheduling problem in each zone as an optimisation model. The model constraints correspond to the boundary conditions for safe and reliable railway operations, while the objective

Figure 4.1: Elements of railway traffic rescheduling



function measures the quality of the new adapted schedule according to predefined criteria. Section 4.3 highlights that the complexity of the resulting formulation (in the following referred to as the *monolith*) is too high for the model to be solved in a short time by simply applying a commercial software. This motivates the development of alternative resolution approaches based on column generation techniques and problem partitioning schemes which are described in Sections 4.4 and 4.5 respectively. Section 4.4 introduces a column generation framework with different pricing approaches which aims at speeding up the resolution of the local rescheduling problem by limiting the number of alternative train trajectories considered concurrently by the model (in the following these approaches will be shortly referred to as *cgApP*, *cgApC*, *cgPpP*, and *cgPpC*). Section 4.5 proposes a non-hierarchical coordination approach for neighbouring zones. This aims at enabling the partitioning of the rescheduling problem according to the available compute server and, thus, at speeding up the solution process by reducing the size of each local rescheduling model and by limiting the decision measures to the most relevant ones for the underlying topology and train density.

4.1 Network and problem partitioning

This section introduces the main principles of problem partitioning and defines types of zones for local traffic rescheduling. The literature review in Section 3.3 suggests that the geographic extension and the amount of traffic considered by a railway traffic rescheduling model have to be limited to get problems of tractable sizes, which can be solved rapidly. Consequently, it is sensible to split network-wide rescheduling problems into smaller sub-problems. The following definitions of *network partitioning* and *problem partitioning* are used hereinafter.

Definition 4.1 *A* network partitioning *of a network $\mathcal{N}$ consists of a set $\mathcal{N}_0, \mathcal{N}_1, \ldots, \mathcal{N}_m$ of connected sub-networks or* zones *of $\mathcal{N}$ such that*

(R1) the zones cover the entire network, i.e., $\bigcup_{i=0}^{m} \mathcal{N}_i = \mathcal{N}$,

(R2) the boundaries $\partial\mathcal{N}_i$ of the each zone $\mathcal{N}_i$ are operation points, i.e., $\partial\mathcal{N}_i \subset \mathcal{S}$, $\forall 0 \leq i \leq m$, where $\mathcal{S}$ denotes the set of operation points in $\mathcal{N}$.

The set $S_{i,j} := \left\{s \in \mathcal{S} | s \in \mathcal{N}_i \cap \mathcal{N}_j\right\}$ denotes the portals *between zones $\mathcal{N}_i$ and $\mathcal{N}_j$, i.e., the operation points at the common boundary. And $\mathcal{S}_{portals} := \bigcup_{i=0}^{m} \bigcup_{j>i} S_{ij}$ indicates the set of all portals in the network $\mathcal{N}$*

in the network partitioning. Each operation point $s \in \mathcal{S} \setminus \mathcal{S}_{portals}$ is in exactly one sub-network $\mathcal{N}_{\cdot}$. Zones sharing a boundary, i.e. $S_{ij} \neq \emptyset$, are said adjacent *or* neighbouring.

Definition 4.2 *Let a network partitioning $\mathcal{N}_0, \mathcal{N}_1, \ldots, \mathcal{N}_m$ of $\mathcal{N}$ be given. And let $\Phi = \min_{x \in X} f(x)$ be the mathematical representation of the rescheduling problem in network $\mathcal{N}$, where x is a vector of decision variables, X is the set of decisions satisfying the constraints of railway traffic management, and f represents the objectives of railway traffic rescheduling (e.g., minimise train delays). A* problem partitioning *of Φ is a set of local rescheduling problems $\phi^0, \phi^1, \ldots, \phi^m$ in sub-networks $\mathcal{N}_0, \mathcal{N}_1, \ldots, \mathcal{N}_m$ that can be solved in a distributed way and such that combining their solutions one obtains a feasible global schedule.*

Given that solutions to decomposed optimisation problems are usually suboptimal and the coordination effort might be large enough to even annihilate the time savings produced by decomposition, it is important that the railway network is partitioned into as few sub-problems as manageable. In case of small disturbances, one can assume that the number of trains to be considered for the new schedule does not vary significantly from the yearly timetable, which is fixed. Thus, the *capacity consumption of railway infrastructure* of the yearly timetable plays a key role for defining a low number of sub-problems and, at the same time, for keeping the computational effort of solving each sub-problem within the limits of current processors.

The next section gives some basic definitions that are used in this section and describes a concept, first introduced by Laube et al. (2007), to partition the railway network into condensation and compensation zones, according to the capacity consumption. Section 4.1.2 outlines different rescheduling policies for both zone types that also account for the complexity of the railway topology in each zone.

4.1.1 Network partitioning and types of zones

In the current work, traffic density is measured via capacity consumption of the underlying infrastructure. According to UIC-code 406 (2004), capacity consumption is the infrastructure occupation plus time supplements for stability requirements. Mixed traffic lines with infrastructure occupation greater than 75% in peak hours (60% off-peak) are said to be congested. Capacity consumption is defined by the number of trains, heterogeneity, timetable, and stability requirements (see Appendix A for some examples). Infrastructure occupation of line sections is computed by compressing the timetable over a period of time (e.g., one hour) and dividing the resulting occupation at the first block section by the length of the period (UIC-code 406, 2004).

The application of UIC-code 406 (2004) to stations and junctions is not straightforward and has been investigated by several researchers. Landex (2008) suggests not to split these entities during capacity consumption analyses in order to detect all possible conflicts. In addition, larger stability buffers should be included in large station areas to implicitly consider shunting movements that are not reported in the published timetable. Lindner (2011) concludes that UIC-code 406 (2004) cannot be used to compute the absolute capacity of station areas because it strongly depends on the routes assigned to the trains (one route may exclude all other routes, while other ones may be used simultaneously). Alternatively, Frank (2013) proposes a mesoscopic approach to compute infrastructure occupation based on the headway times required between scheduled services. Station areas are divided into conflict points which are analysed separately according to the main traffic flows. This approach neglects the interdependence of different conflict points and shunting movements but suggests that capacity consumption does not only depend on the actual occupation on infrastructure but also on the type of conflicts that may arise (e.g., crossing, follow-up, etc.).

A network partitioning considering capacity consumption and resulting in a relatively low number of zones was suggested and used by Laube et al. (2007), Caimi (2009), and Fuchsberger (2012). The following definitions have been associated by Caimi (2009) to the network partitioning in condensation and compensation zones:

Definition 4.3 *(Caimi, 2009)*

> Condensation zones *are the saturated zones*[1] *of the network. The available unused capacity is scarce, thus they represent the bottlenecks of the network. Here, the network shall be used at the maximal possible capacity that allows stable operation. Therefore, the timetabling policy for these zones is such that no time reserves are introduced, and the specified track paths must be followed very precisely. Usually, but not necessarily, condensation zones are situated around major stations.*
>
> Compensation zones *are the zones that connect different condensation zones and mostly have additional capacity available. They serve as recovery zones for the trains with respect*

[1]e.g., according to UIC-code 406 (2004).

> *to punctuality and following of a determined track path. In compensation zones, trains will be controlled in order to enter the next condensation zone precisely at the predetermined time and speed. This way, the condensation zone can be passed without losing any capacity. Time reserves are introduced to reduce possible delays coming from previous condensation zones and thereby to improve timetable stability.*

The indivisibility of resources *prescribes that each infrastructure resource is in at most one zone.*

Given that traffic heterogeneity and timetable structure may vary from day to day or even within the same day (e.g., rush hour, off peak, and night), condensation and compensation zones may change over the time. In addition, these characteristics may deviate from the planned values due to real-time operational issues. This suggests the need for flexible and dynamic network decomposition. However, the time available for real-time rescheduling is usually very limited. Thus, a trade-off is needed between dynamic network decomposition, which is the most proper theoretical approach, and time-to-solution, which has to be short. Given that the currently investigated approach is supposed to solve relatively small disturbances, which do not significantly modify traffic heterogeneity and timetable structure, a static network decomposition is applied. The network decomposition is determined off-line using the yearly timetable. This way, zones may change over time according to planned railway operations but not to deviations occurring during real-time operations.

4.1.2 Rescheduling policies in condensation and compensation zones

Ideally, all possible rescheduling options should be considered everywhere: retiming, speed advisory, reordering and rerouting. However, as it will be shown in Section 4.3, this usually results in huge problems that are not tractable in the short time available during real-time traffic management. In particular, dense traffic implies a large number of trains, thus a large number of retiming, speed advisory and reordering decisions. Similarly, complex track topologies such as large stations or junctions provide many paths, thus a large number of rerouting alternatives for each train. In addition, the paths in complex topologies are usually highly interdependent, given that each path blocks all paths sharing the same infrastructure elements.

Caimi (2009)'s condensation zones usually coincide with station areas and the following timetabling policy applies:

> *Policy 5.6 (Scheduling in condensation zones)* In condensation zones, each train is scheduled to travel at the maximally allowed speed on the designed route. This maximal speed profile can vary depending on the given route.

Fuchsberger (2012) successfully applies this policy to real-time rescheduling in condensation zones around main station areas. However, this policy might not be suitable for real-time rescheduling in highly congested areas with simpler topologies. For instance, Montigel (2009) and Oettich and Caspar (2015) report on successful implementations of speed advisory systems for conflict avoidance at heavily congested junctions in the Lötschberg base tunnel and in Killwangen (where occupation ratios are larger than 75% Weidmann et al., 2014, p. 82). Thus, in the following, the timetabling policies by Caimi (2009) are refined to be applicable for real-time rescheduling in condensation and compensation zones and to consider track topology too. [2]

Topologies where conflicts occur between trains having different entry and exit points from the shared resource (German *Konfliktfall Kreuzung*) are denoted hereinafter as *complex topologies*. These are opposed to *simple topologies* where conflicting trains have at least one entry or exit point in common (German *Konfliktfälle: Folgefahrt, Einfädelung, Abfolge, Ausfädelung*). With this definition, complex condensation zones are usually situated around congested major freight terminals or stations where many passengers alight or transfer to connecting trains. These usually correspond to the nodes of the periodic timetable.

Four rescheduling policies can be defined: (1) for condensation zones situated in complex topological areas; (2) for condensation zones in simple areas; (3) for compensation zones in complex areas; and (4) for compensation zones in simple area.

(1) Dense traffic on complex topologies yields to a huge number of rerouting, retiming (speed) and reordering decisions. Here, capacity is usually limited by switch regions. Topologically, these areas are characterised by many switches, which implies many routing possibilities as well as high route interdependence. The latter point prescribes that trains should traverse the area at maximum speed[3]

[2] Note that this dependency on infrastructure is in line with several research approaches outlined in Section 3.3, which consider different rescheduling policies and models for lines and junctions/stations.

[3] Note that this speed is usually low due to the switches

in order to affect the minimum number of routes of other trains (scheduling policy 5.6 Caimi, 2009). Usually, complex topologies are built where traffic is dense. However, outside peak hours they might experience a low amount of traffic and become part of a compensation zone.

(2) Simple condensation zones contain few switches and tracks, and possibly some minor station. Traffic might be very intense and heterogeneous. In simple condensation zones, capacity is usually limited by track capacity. Here, speed plays a central role in conflict resolution, because there is usually no alternative routing.

(3-4) Compensation zones can be either topologically complex or simple. Due to the lower traffic density, it should not be necessary to limit the speed profiles to the maximal ones. However, in complex compensation zones, if the amount of trains is not critical, it might be unnecessary to use routes requiring many track changes.

In condensation zones, traffic is dense and very little capacity reserve is planned. Thus, the main objective during rescheduling is maintaining feasibility while minimising the inconvenience for operators and customers (cfr. Caimi, 2009, objective in timetabling). In compensation areas, further objectives such as the minimisation of energy consumption can be included to exploit the time reserves not needed to delay absorption.

It has usually been assumed that no rescheduling is needed in compensation zones. Thus, by requiring that each two condensation zones are separated by a compensation zone that has enough reserves to allow each train to enter the next condensation zone at the requested time, there is no need for coordinating rescheduling of neighbouring zones. However, in very densely utilised network, this requirement might yield to the definition of extremely large condensation zones. In fact, if a compensation zone does not have enough reserves, then it is merged with its neighbours into a huge condensation zone. This requirement can be dropped if a suitable coordination process for rescheduling in adjacent zones is specified. Coordination processes are tackled in Section 4.5. Given a suitable coordination process, it is possible to divide even a condensation zone into multiple zones if the traffic density or the infrastructure extension are too large and exceed the performance of the available compute server. In this way it might be possible to partition every railway traffic rescheduling problem into a set of sub-problems that are solvable in the short time available during real-time traffic management.

In the rest of this thesis, it is assumed that a network partitioning in condensation and compensation zones is given. The determination of a general network partitioning strategy is left for future research.

4.2 Resource conflict graph model for local rescheduling (RCG)

The analysis in Section 3.1.3 highlights that the resource conflict graph model satisfies all the functional requirements of real-time traffic management listed in Section 2.3. In addition, preliminary studies showed that this model is very flexible with respect to the objective function (Toletti and Weidmann, 2016; Toletti et al., 2015a, also summarised in Chapter 5). This motivates its further development and future application to local rescheduling.

Fuchsberger (2012) and Caimi et al. (2012) successfully applied the RCG model for predictive control in a condensation area around a large station with highly complex topology. The current work further develops that approach in three ways. First, the next section extends the model to consider multiple operation points such that it can be applied to zones with simple topologies. Second, Section 4.2.2 presents different objective functions and a framework to integrate multiple objectives using the RCG model. Third, a simulation-based approach for blocking time stairways pattern generation is proposed in Section 4.2.3.

4.2.1 RCG model for compensation areas and condensation zones with simple topology [4]

The RCG model for railway traffic rescheduling is an integer linear program in which the (binary) decision variables correspond to decisions about deferring train runs, cancellations, passenger transfers and trajectories between operation points (stations, stops, and junctions). A train trajectory is the time-space-speed representation of a train run. Thus, a train trajectory defines the departure time and track from an operation point and the arrival time and track at the successive operation point, as well as the routes and speed profiles in-between. Train trajectories are protected by blocking time stairways as explained in Section 2.1.

Given a geographical area, let $\mathcal{S}$ be the set of operation points in it, $\mathcal{T}$ be the set of trains travelling in it, and $\mathcal{CN}$ be the set of scheduled

[4] This section is based on Toletti et al. (2017)

passenger connections at the operation points in $\mathcal{S}$. The set of tracks in operation point $s \in \mathcal{S}$ is denoted by P_s, and $P_{s,t}$ is the set of tracks that a specific train $t \in \mathcal{T}$ can use. It is assumed that each train runs through a predefined sequence $(s_0, s_1, \ldots, s_{n_t})$ of operation points, where s_0 is either the departure station, depot or portal from which the train enters the control area and s_{n_t} either the final station, depot or portal from which it leaves the control area. Let $\mathcal{B}_{t,i}$ be a set of alternative blocking time stairways for the run of train t from s_{i-1} to s_i.

In this framework, a *resource* is a subset of infrastructure elements that are locked and released simultaneously according to the *Blocking Time Theory* (see, e.g., Hansen and Pachl, 2014). $\mathcal{R}$ denotes the set of resources in the control area and $\mathcal{B}^r$ the set of all blocking time stairways using some resource $r \in \mathcal{R}$. $\underline{b(r)}$ and $\overline{b(r)}$ denote the lock and release times of resource r by the blocking time stairway $b \in \mathcal{B}^r$. For each resource $r \in \mathcal{R}$, the nodes of the *conflict graph* correspond to the blocking time stairways $\mathcal{B}^r$ and the edges link together the nodes of blocking time stairways overlapping on resource r.

Let $m^{t,t'}_{p,p'}$ be the minimum time needed between an event of train t on track p and an event of t' on p', e.g., the minimum dwelling time if $t = t', p = p'$ or the minimum transfer time for a passenger connection from train t on platform p to train t' on platform p'. Blocking time stairways that are not separated by this minimum time are said to be *incompatible* (nomenclature by Fuchsberger, 2012). The sets of maximal incompatible sets are denoted by $\Omega^{t,t'}_{p,p'}$. Similarly, let mv^t_p be the maximum difference tolerable between the arrival speed of train t at track p and its departure from the same track. Blocking time stairways whose speed difference is larger than the tolerated one are said *speed-incompatible* and the set of maximal speed-incompatible sets is denoted by Ωv^t_p.

The binary decision variables of the problem indicate whether runs between two consecutive stations are scheduled in the current horizon, whether they are cancelled, whether a trajectory corresponding to a blocking time stairway is selected, and whether scheduled connections are kept, i.e.,

$$r_{t,i} = \begin{cases} 1, & \text{if the run of train } t \text{ from } s_{i-1} \text{ to } s_i \text{ is scheduled} \\ & \text{in the current horizon} \\ 0, & \text{else} \end{cases} \tag{4.1}$$

$$h_{t,i} = \begin{cases} 1, & \text{if the run of train } t \text{ from } s_{i-1} \text{ to } s_i \text{ is cancelled} \\ 0, & \text{else} \end{cases} \tag{4.2}$$

$$x_b = \begin{cases} 1, & \text{if the schedule uses } b \\ 0, & \text{else} \end{cases} \tag{4.3}$$

$$c_{s,t,t'} = \begin{cases} 1, & \text{if the connection from } t \text{ to } t' \\ & \text{in station } s \text{ is kept} \\ 0, & \text{else.} \end{cases} \tag{4.4}$$

Since freight and service trains cannot be cancelled, $h_{\cdot}$ variables are defined for passenger trains only. If a train run is neither scheduled in the current horizon nor cancelled, then it is deferred to the next horizon.

Linear constraints model the functional requirements of railway operations, time-space-speed consistency, conflict freedom, and feasibility of passenger transfers, as follows:

$$r_{t,i} \leq r_{t,i-1} \quad \forall t \in \mathcal{T}, i = 2, \ldots, n_t \tag{4.5}$$

$$r_{t,i} + h_{t,i} \leq 1 \quad \forall t \in \mathcal{T}_p, i = 1, \ldots, n_t \tag{4.6}$$

$$h_{t,i} \geq h_{t,i-1} \quad \forall t \in \mathcal{T}_p, i = 2, \ldots, n_t \tag{4.7}$$

$$r_{t,i} = \sum_{b \in \mathcal{B}_{t,i}} x_b \quad \forall t \in \mathcal{T}, i = 2, \ldots, n_t \tag{4.8}$$

$$\sum_{b \in C} x_b \leq 1 \quad \forall C \in C^r, r \in \mathcal{R} \setminus \bigcup_{s \in \mathcal{S}} P_s \tag{4.9}$$

$$0 \leq \sum_{\substack{b \in A_{\cdot,p} \\ \alpha(b) \leq \alpha}} x_b - \sum_{\substack{b \in D_{\cdot,p} \\ \delta(b) < \alpha}} x_b \leq 1 \quad \forall \alpha \in \{\alpha(b), b \in A_{\cdot,p}\}, p \in P_s, s \in \mathcal{S} \tag{4.10}$$

$$\sum_{\substack{b \in A_{\cdot,p} \\ \alpha(b) = \alpha}} x_b \leq 1 \quad \forall \alpha \in \{\alpha(b), b \in A_{t,p}\}, p \in P_s, s \in \mathcal{S} \tag{4.11}$$

$$\sum_{b \in A_{t,p}} x_b - \sum_{b \in D_{t,p}} x_b \leq 0 \quad \forall p \in P_{\cdot,t}, t \in \mathcal{T} \tag{4.12}$$

$$\sum_{b \in U} x_b + \sum_{b \in V} x_b \leq 1 \quad \forall (U, V) \in \Omega^{t,t}_{p,p}, p \in P_{s,t} \tag{4.13}$$

$$\sum_{b \in U} x_b + \sum_{b \in V} x_b \leq 1 \quad \forall (U, V) \in \Omega v^{t}_{p}, p \in P_{s,t} \tag{4.14}$$

$$c_{s,t,t'} + \sum_{b \in U} x_b + \sum_{b \in V} x_b \leq 2 \quad (U, V) \in \Omega^{t,t'}_{p,p'}, p \in P_{s,t}, p' \in P_{s,t'}, (s, t, t') \in C\mathcal{N} \tag{4.15}$$

$$c_{s_i,t,t'} \leq r_{t,i} \quad (U, V) \in \Omega^{t,t'}_{p,p'}, p \in P_{s_i,t}, p' \in P_{s_i,t'}, (s_i, t, t') \in C\mathcal{N} \tag{4.16}$$

$$c_{s_{i-1},t,t'} \leq r_{t',i} \qquad \begin{array}{l}(U,V) \in \Omega^{t,t'}_{p,p'}, p \in P_{s_{i-1},t}, \\ p' \in P_{s_{i_1},t'}, (s_{i-1},t,t') \in \mathcal{CN}.\end{array} \tag{4.17}$$

Constraints (4.5) ensure that at most one blocking time stairway is allocated to each successive section and none is allocated to trains that have not been scheduled at a previous point. Constraints (4.6) link $r_{t,i}$ to $h_{t,i}$ variables, where $\mathcal{T}_p$ denotes the set of passenger trains. Cancellation consistency is ensured by constraints (4.7). Constraints (4.8) link $r_{t,i}$ to x_b variables. Conflicts are avoided thanks to constraints (4.9-4.11): (4.10-4.11) prevent conflicts at operation points , while (4.9) are used for the other resources, where C^r denotes the set of maximal cliques of the conflict graph for resource r. Constraints (4.12) force trains to depart from operation points from the same track they have arrived to. The *self-connection* constraints (4.13) ensure that trains only depart after they have arrived to the operation point and the minimum dwelling time has expired and that the arrival and departure times coincide (up to a given tolerance value) for trains passing an operation point without stopping. Constraints (4.14) ensure speed consistency. Connection constraints (4.15) model decisions about passenger connections, and (4.16-4.17) link them to decisions on train runs.

The RCG model by Fuchsberger (2012) for rescheduling around a main station area already included the decision variables $x.$ and $c.$ along with the conflict constraints (4.9) and (4.10), the self-connection constraints (4.12) and (4.13), and the connection constraints (4.15). That model considered only one run arriving at and one departing from the main station for each train and no cancellation. Here, the decision variables $r.$ and $h.$ serve to link the different runs of each train. Constraints (4.5), (4.7), (4.6), (4.8), (4.16) and (4.17) have been added to link the decision variables with each other. Constraints (4.12) have been relaxed to enable a train run to arrive to an operation point during the considered horizon and the departure from the operation point to be deferred to the next one. The previous model assumed that all trains stop at the operation point (i.e., the large station). To enable trains to travel through an operation point without stopping, constraints (4.13) have been augmented to enforce time consistency and (4.14) have been defined to enforce speed consistency. In addition, conflict constraints (4.11) have been defined to prevent conflicts involving non-stopping trains at operation points.

The objective function can model train delays, customer inconvenience (delays, waiting times, connections missed, cfr. Caimi et al., 2012; Toletti and Weidmann, 2016) and/or energy targets (Toletti et al., 2015a). Examples of objective functions and a framework for multi-criteria railway traffic rescheduling are presented in the following section, while the notation is summarised in Table 4.1.

4.2.2 Objectives for RCG model[5]

The following objective functions measure train delay, customer satisfaction, and energy efficiency for model (4.1)-(4.14). The **operator's perspective** is reflected by the overall train delay, i.e., the sum of arrival delays at scheduled stops or at the boundaries of the control area, which can be computed in the following ways

$$
\begin{aligned}
td_1 = & \sum_{t\in\mathcal{T}} \sum_{\substack{i=1,\dots,n_t | \\ \exists \text{scheduled} \\ \text{event at } s_i}} (1 - r_{t,i} - h_{t,i}) \cdot \Delta \\
& + \sum_{t\in\mathcal{T}} \sum_{\substack{i=1,\dots,n_t | \\ \exists \text{scheduled} \\ \text{event at } s_i}} \sum_{b\in\mathcal{B}_{t,i}} x_b \left(\alpha(b) - \hat{\alpha}_{t,i}\right)
\end{aligned}
\tag{4.18}
$$

$$
\begin{aligned}
td_2 = & \sum_{t\in\mathcal{T}} \sum_{\substack{i=1,\dots,n_t | \\ \exists \text{scheduled} \\ \text{event at } s_i}} (1 - r_{t,i} - h_{t,i}) \cdot \left(\overline{\mathcal{H}} + mr_{t,i} - \hat{\alpha}_{t,i}\right) \\
& + \sum_{t\in\mathcal{T}} \sum_{i=1,\dots,n_t} (1 - r_{t,i} - h_{t,i}) \cdot (mr_{t,i} + dt_{t,i}) \cdot \sum_{\substack{j=i+1,\dots,n_t | \\ \exists \text{scheduled} \\ \text{event at } s_j}} 1 \\
& + \sum_{t\in\mathcal{T}} \sum_{\substack{i=1,\dots,n_t | \\ \exists \text{scheduled} \\ \text{event at } s_i}} \sum_{b\in\mathcal{B}_{t,i}} x_b \left(\alpha(b) - \hat{\alpha}_{t,i}\right)
\end{aligned}
\tag{4.19}
$$

where Δ denotes the size of the rescheduling horizon, $\alpha(b)$ is the arrival time of blocking time stairway b and $\hat{\alpha}_{t,i}$ the scheduled arrival time of train t at s_i, $\overline{\mathcal{H}}$ is the rescheduling horizon end, $mr_{t,i}$ and $dt_{t,i}$ are the minimum running time of train t in section i and the minimum dwell at s_i respectively. The first case assigns a fixed penalty equals the length of the rescheduling horizon to runs that are scheduled outside the rolling horizon (first row of 4.18); while the latter estimates the earliest possible time of each scheduled event outside the rolling horizon (first row of 4.19) and the effect on the successive sections (second row of 4.19). The last rows of (4.18-4.19) quantify the delay of the chosen blocking time stairways.

[5]This section is based on Toletti et al. (2017)

Customer satisfaction is measured in terms of passenger delay and freight delay. Passenger delay corresponds to the sum of arrival delays of passenger trains at scheduled stops weighted by the number of passengers alighting. Here, missed passenger connections are penalised using fixed delay equivalents multiplied by the number of passengers who were planning to use each connection. Approaches accounting for the effects of the broken transport chain can be found in Schöbel (2007).

$$\begin{aligned} pd_1 = &\sum_{t\in\mathcal{T}_p} \sum_{\substack{i=1,\dots,n_t|\\ \exists\text{scheduled}\\ \text{event at } s_i}} \gamma_{t,i}\cdot(1-r_{t,i}-h_{t,i})\cdot\Delta \\ &+\sum_{t\in\mathcal{T}_p} \sum_{\substack{i=1,\dots,n_t|\\ \exists\text{scheduled}\\ \text{event at } s_i}} \gamma_{t,i}\cdot\left(h_{t,i}\Gamma_{t,i}+\sum_{b\in\mathcal{B}_{t,i}} x_b\left(\alpha(b)-\hat{\alpha}_{t,i}\right)\right) \\ &+\sum_{(s,t,t')\in\mathcal{CN}} \gamma_{s,t,t'}\Gamma_{s,t,t'}(1-c_{s,t,t'}) \end{aligned} \tag{4.20}$$

$$\begin{aligned} pd_2 = &\sum_{t\in\mathcal{T}_p} \sum_{\substack{i=1,\dots,n_t|\\ \exists\text{scheduled}\\ \text{event at } s_i}} \gamma_{t,i}\cdot(1-r_{t,i}-h_{t,i})\cdot\left(\overline{\mathcal{H}}+mr_{t,i}-\hat{\alpha}_{t,i}\right) \\ &+\sum_{t\in\mathcal{T}_p}\sum_{i=1,\dots,n_t}(1-r_{t,i}-h_{t,i})\cdot(mr_{t,i}+dt_{t,i})\cdot\sum_{\substack{j=i+1,\dots,n_t|\\ \exists\text{scheduled}\\ \text{event at } s_j}}\gamma_{t,j} \\ &+\sum_{t\in\mathcal{T}_p} \sum_{\substack{i=1,\dots,n_t|\\ \exists\text{scheduled}\\ \text{event at } s_i}} \gamma_{t,i}\cdot\left(h_{t,i}\Gamma_{t,i}+\sum_{b\in\mathcal{B}_{t,i}} x_b\left(\alpha(b)-\hat{\alpha}_{t,i}\right)\right) \\ &+\sum_{(s,t,t')\in\mathcal{CN}} \gamma_{s,t,t'}\Gamma_{s,t,t'}(1-c_{s,t,t'}) \end{aligned} \tag{4.21}$$

where $\gamma_{t,i}$ is the number of passengers alighting t at s_i, $\gamma_{s,t,t'}$ the number of passengers transferring from train t to t' at s, and $\Gamma_{t,i}$ and $\Gamma_{s,t,t'}$ the delay equivalents of cancelling the i run of train t and the connection from t to t' at s (usually the timetable period). Hereinafter, the numbers of passengers $\gamma_{t,i}$ and $\gamma_{s,t,t'}$ are assumed to be fixed and given as an input to the model. For real world applications these values can be approximated by the travel demand models, which are usually known by train operating companies. For instance, SBB estimates them using the surveys about the transport chains of passengers made several times per year by the train staff. This assumption is made also by Rückert et al. (2017) for the development of a webtool that simulates the consequences of wait-depart decisions on passengers. Although the currently available information about passengers flows come from static demand models, the authors anticipate that data on the actual flows will be available soon thanks to the increasing digitalisation of the railway sector.

Freight trains usually have fixed time windows for arriving at their destination. Out of its corresponding window a train can not be processed, e.g., due to lack of marshalling staff or equipment. This is usually neglected by rescheduling approaches, including the current one. Here, freight delay corresponds to the sum of arrival delays of freight trains either at their final destination or when leaving the control area

$$fd_1 = \sum_{t\in\mathcal{T}\setminus\mathcal{T},p} \sum_{\substack{i=1,\dots,n_t| \\ \exists\text{scheduled} \\ \text{event at } s_i}} \left((1-r_{t,i})\cdot\Delta + \sum_{b\in\mathcal{B}_{t,i}} x_b\left(\alpha(b)-\hat{\alpha}_{t,i}\right)\right) \quad (4.22)$$

$$\begin{aligned} fd_2 = &\sum_{t\in\mathcal{T}\setminus\mathcal{T}_p} \sum_{\substack{i=1,\dots,n_t| \\ \exists\text{scheduled} \\ \text{event at } s_i}} (1-r_{t,i})\cdot\left(\overline{\mathcal{H}}+mr_{t,i}-\hat{\alpha}_{t,i}\right) \\ &+ \sum_{t\in\mathcal{T}\setminus\mathcal{T}_p} \sum_{i=1,\dots,n_t} (1-r_{t,i})\cdot(mr_{t,i}+dt_{t,i})\cdot \sum_{\substack{j=i+1,\dots,n_t| \\ \exists\text{scheduled} \\ \text{event at } s_j}} 1 \\ &+ \sum_{t\in\mathcal{T}\setminus\mathcal{T}_p} \sum_{\substack{i=1,\dots,n_t| \\ \exists\text{scheduled} \\ \text{event at } s_i}} \sum_{b\in\mathcal{B}_{t,i}} x_b\left(\alpha(b)-\hat{\alpha}_{t,i}\right) \end{aligned} \quad (4.23)$$

Energy efficiency and operating costs are measured in terms of overall energy consumption, i.e., the sum of the energy needed by each train to run through the control area

$$e = \sum_{t\in\mathcal{T}} \sum_{i=1,\dots,n_t} \sum_{b\in\mathcal{B}_{t,i}} x_b E(b) \quad (4.24)$$

where $E(b)$ denotes the energy consumption of blocking time stairway b.

Multiple objectives can be simultaneously considered by RCG using the ϵ-constraint method. Unlike the weighted sum method, the ϵ-constraint method can identify all Pareto-optimal solutions to a multi-criteria discrete optimisation problem. The objective space is explored by optimising one of the criteria while constraining the others to given values, usually denoted by ϵ. The classical version (Haimes et al., 1971) assumes that these values are chosen in advance, which requires a deep knowledge of the problem as a too coarse grid might miss some optimal solutions and a too fine grid might result in over-proportioned computational effort with respect to the number of Pareto-optimal solutions obtained. Laumanns et al. (2006) propose an adaptive procedure that starts by solving the ”unconstrained” problem and in each iteration defines the epsilons to be used successively, until all Pareto-optimal solutions are identified.

This method can be applied to the multi-criteria rescheduling problem using the four objectives defined above:

$$\min \quad td. \tag{4.25}$$

$$\text{subject to} \quad (4.1) - (4.17) \tag{4.26}$$

$$\underline{\epsilon}_p < pd. \leq \overline{\epsilon}_p \tag{4.27}$$

$$\underline{\epsilon}_f < fd. \leq \overline{\epsilon}_f \tag{4.28}$$

$$\underline{\epsilon}_e < e \leq \overline{\epsilon}_e. \tag{4.29}$$

the overall train delay is minimised as the primary objective function, while ϵ-constraints are introduced for passenger delay, freight delay, and energy consumption.

Table 4.1: Notation of the resource conflict graph model

Input sets	
$\mathcal{R}$	(infrastructure) resources in control area
$\mathcal{S}$	operation points (stations, stops, junctions, portals) in control area
P_s	tracks at operation point $s \in \mathcal{S}$
$\mathcal{T}$	trains running in the control area during the considered horizon
$\mathcal{T}_p \subseteq \mathcal{T}$	passenger trains running in the control area during the considered horizon
$\mathcal{T}_f \subseteq \mathcal{T}$	freight trains running in the control area during the considered horizon
$\mathcal{CN}$	passenger transfers scheduled in the area during the considered horizon
	Continued on next page

Table 4.1 – continued from previous page

$\mathcal{B}_{t,i}$	blocking time stairway candidates (combinations of routing, timing and speed control) for the run of train $t \in \mathcal{T}$ in section $i = 1, \ldots, n_t$
Input (timetable) parameters	
$(s_0, s_1, \ldots, s_{n_t}) \in \mathcal{S}^{n_t+1}$	ordered operation points on the itinerary of train $t \in \mathcal{T}$
$\hat{\alpha}_{t,i}$	scheduled arrival time of train $t \in \mathcal{T}$ at i^{th} operation point on its itinerary
$\hat{\delta}_{t,i}$	scheduled departure time of train $t \in \mathcal{T}$ from i^{th} operation point on its itinerary
$mr_{t,i}$	minimum running time for train $t \in \mathcal{T}$ from operation point s_{i-1} to s_i
$dt_{t,i}$	minimum dwelling time for train $t \in \mathcal{T}$ at operation point s_i
Decision variables	
$r_{t,i}$	decision of including i^{th} run of train $t \in \mathcal{T}$ in rescheduling solution of considered rescheduling horizon
$h_{t,i}$	decision of cancelling i^{th} run of train $t \in \mathcal{T}$
x_b	decision about including blocking time stairway b in rescheduling solution
$c_{s,t,t'}$	wait-depart decision concerning a scheduled connection from train $t' \in \mathcal{T}$ to $t \in \mathcal{T}$ at station $s \in \mathcal{S}$
Model sets	
$\mathcal{B}^r \subseteq \bigcup_{t\in\mathcal{T}} \bigcup_{i=1}^{n_t} \mathcal{B}_{t,i}$	blocking time stairways using resource $r \in \mathcal{R}$
C^r	maximal conflict cliques for resource $r \in \mathcal{R}$
$A_{t,p}$	blocking time stairway candidates for train $t \in \mathcal{T}$ arriving at operation point track $p \in \bigcup_{s\in\mathcal{S}} P_s$
$D_{t,p}$	blocking time stairway candidates for train $t \in \mathcal{T}$ departing from operation point track $p \in \bigcup_{s\in\mathcal{S}} P_s$

Continued on next page

Table 4.1 – continued from previous page

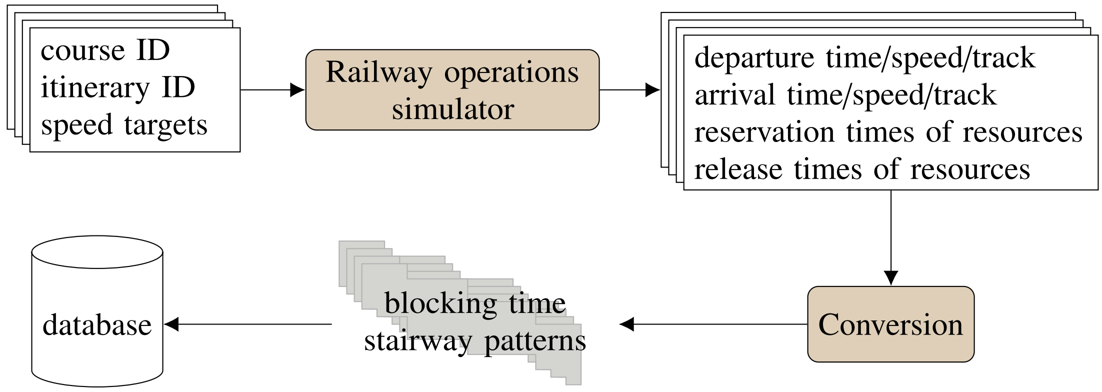
$A_{\cdot,p} = \bigcup_{t\in\mathcal{T}} A_{t,p}$	blocking time stairways arriving at operation point track $p \in \bigcup_{s\in\mathcal{S}} P_s$
$D_{\cdot,p} = \bigcup_{t\in\mathcal{T}} A_{t,p}$	blocking time stairways departing from operation point track $p \in \bigcup_{s\in\mathcal{S}} P_s$
$\Omega^{t,t'}_{p,p'}$	maximal incompatible sets for an event of train $t \in \mathcal{T}$ at operation point track $p \in P_s$ followed by an event of train $t' \in \mathcal{T}$ at $p' \in P_s$ for some operation point $s \in \mathcal{S}$
Functions	
$\alpha(b)$	arrival time of blocking time stairway b
$\delta(b)$	departure time of blocking time stairway b
$E(b)$	energy consumption of blocking time stairway b
Parameters (objectives)	
Δ	size of rescheduling horizon
$\overline{\mathcal{H}}$	end of rescheduling horizon
$\gamma_{t,i}$	number of passengers who alight train $t \in \mathcal{T}$ at s_i
$\gamma_{s,t,t'}$	number of passengers who transfer from train $t \in \mathcal{T}$ to train $t' \in \mathcal{T}$ at $s \in \mathcal{S}$
$\Gamma_{t,i}$	delay equivalent of cancelling i^{th} run of train $t \in \mathcal{T}$
$\Gamma_{s,t,t'}$	delay equivalent for breaking connection from train $t \in \mathcal{T}$ to train $t' \in \mathcal{T}$ at $s \in \mathcal{S}$

4.2.3 Generation of blocking time stairway patterns using simulations[6]

In order to produce realistic solutions that can actually be applied to real-world railway operations, all relevant pieces of information on the train movements must be reflected in the model. With the RCG model described in Section 4.2.1, this means that the blocking time stairways b associated with the decision variables x_b must contain the exact route lock and route-section release points and times, which can only be obtained by considering a very fine infrastructure granularity including track-circuits.

[6]This section is extracted from (Toletti et al., 2017)

Figure 4.2: Simulation based variable generation



In addition, each blocking time stairway must allow at least one trajectory that respects rolling stock traction and brake capabilities as well as vehicle and line speed limits. This level of granularity is often used by railway simulators. Given the infrastructure and the rolling stock, railway simulation tools reproduce train dynamics and interlocking information very accurately. In addition, some software packages are also able to provide energy consumption measures for the simulated train movements.

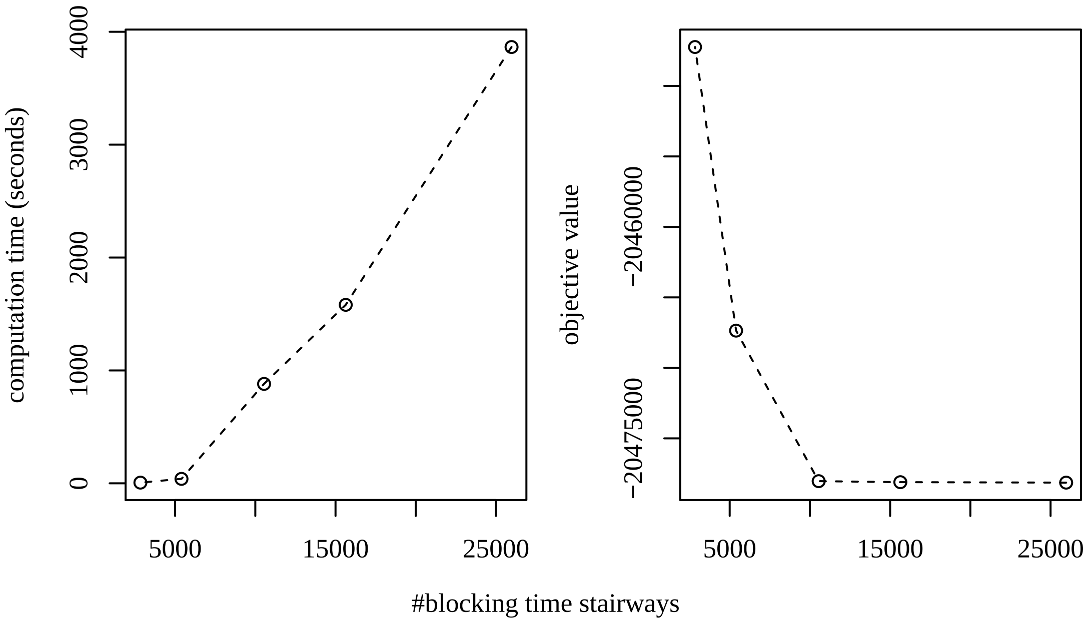
To save time during real-time operations, blocking time stairways to be used as decision alternatives can be prepared offline in a preprocessing step. This can be done taking advantage of the strengths of railway simulators mentioned before. Figure 4.2 depicts the blocking time stairways generation framework. A generator program fixes vehicles, routes, stopping patterns, and speed targets into a simulator and receives information about lock and releasing times of the infrastructure resources and energy consumption. With these pieces of information it generates the blocking time stairway patterns for the rescheduling model and stores them in a database.

Hereinafter, it is assumed that conflict freedom is achieved when all trains encounter only green signals. This setting is usually referred to as *green wave* principle. If trains only encounter green signals, they do not have to stop nor brake and they can free the utilised resources very quickly, which reduces the capacity consumption. For this reason, each train run is simulated separately and conform to the green wave principle.

4.3 Complexity of the ”monolith” approach

In the following, *monolith* refers to a RCG model built with all time-shifted copies of the patters in a database of blocking time stairways that

Figure 4.3: Performance of the monolith (results of the experiment described in Section 5.6.1 on server ivt-pikelot).



fit into the considered rescheduling horizon for a given time-shift size and solved by a commercial solver.

Figure 4.3 shows the performance of the monolith approach in terms of computation time and solution quality (the objective value must be minimised) for a numerical experiment (for detailed information see Chapter 5). The computation time grows over-proportionally with the number of blocking time stairways used to generate the decision variables of the RCG model while the solution quality stagnates when the number of blocking time stairways reaches some threshold. This indicates that a column generation approach might shorten the computation times without a significant loss of solution quality.

Another issue that can become critical is the memory required for storing the blocking time stairway patterns. In fact, a huge number of patterns can be generated for areas with highly heterogeneous traffic, complex infrastructure topology, and/or more possibilities to pass speed advices to trains. The number of blocking time stairway patterns that need to be stored in the database can be approximated as follows. Let $\mathcal{S}$ and $\mathcal{T}$ be the sets of operation points and trains to be considered during rescheduling. It is assumed that each train runs through a predefined sequence $(s_0, s_1, \ldots, s_{n_t})$ of operation points in the area. Let $\mathcal{B}_{t,i}$ be a set of blocking time stairways for the run of train t from s_{i-1} to s_i.

The size of $\mathcal{B}_{t,i}$ can be computed in two steps (Figure 4.4). First, consider only one train t and a fixed path p from operation point s_{i-1} to s_i. If the speed profiles are communicated to trains using the signalling system, then the number of possible speed profiles that can be considered depends on the stopping pattern of t, the number of blocks m_p along p, and on the number of aspects of each signal SA:

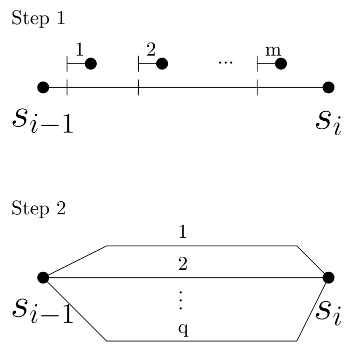


Figure 4.4: Computing the number of decision variables.

$$(2 - \mathbb{1}_{t,i-1}) \cdot (2 - \mathbb{1}_{t,i}) \cdot SA^{m_p}$$

where $\mathbb{1}_{t,i}$ indicates whether there is a scheduled stop of train t at s_i. Second, considering all possible paths between operation points s_{i-1} and s_i, one gets that the size of $\mathcal{B}_{t,i}$ is

$$|\mathcal{B}_{t,i}| = \sum_{\substack{p\in \text{ paths} \\ s_{i-1}\rightarrow s_i}} (2 - \mathbb{1}_{t,i-1}) \cdot (2 - \mathbb{1}_{t,i}) \cdot SA^{m_p} \tag{4.30}$$

Summing up over the entire itinerary $(s_0, s_1, \ldots, s_{n_t})$ of train t and over all trains, assuming that each train uses a different rolling stock, the number of patterns in the database is

$$\begin{aligned} |database| &= \sum_{t\in\mathcal{T}} \sum_{i=0}^{n_t} |\mathcal{B}_{t,i}| \\ &= \sum_{t\in\mathcal{T}} \sum_{i=1}^{n_t} \sum_{\substack{p\in \text{ paths} \\ s_{i-1}\rightarrow s_i}} (2 - \mathbb{1}_{t,i-1}) \cdot (2 - \mathbb{1}_{t,i}) \cdot SA^{m_p} \end{aligned} \tag{4.31}$$

From formula (4.31) it is possible to draw some conclusions about the number of decision variables in the problem:

- The number of trains affects the number of decision variables super-linearly. Thus, limiting the rescheduling horizon length to cover a lower number of trains reduces the problem size.

- The number of sections (between consecutive operation points) of each train has a linear effect on the number of decision variables.

Thus, limiting the extension of the control area reduces the size of the problem.

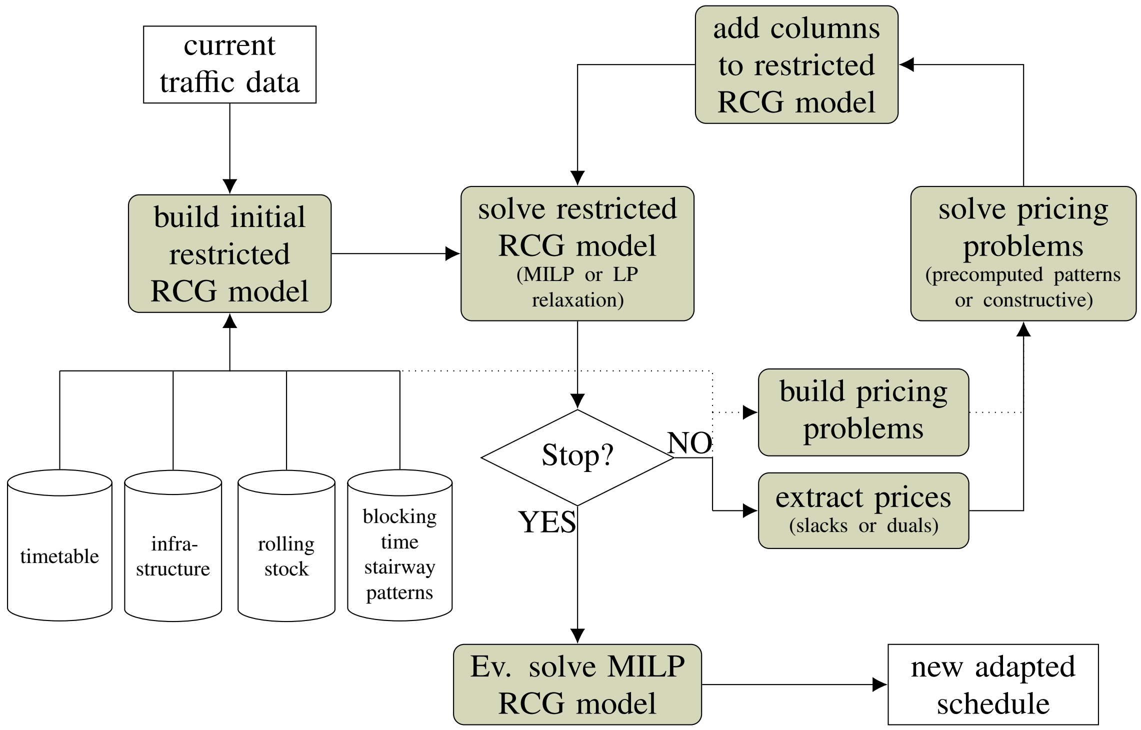
- The number of routing alternatives (paths) between consecutive operation points influences the problem size linearly. This was also highlighted by other research groups working on different models (e.g., Samà et al., 2016). Thus, an a-priori selection of paths reduces the problem size. This can be done with meta-heuristic methods (cfr. Tabu search for routing with alternative graphs by Corman et al., 2010b). Alternatively, as it will be shown in the following of this work, the path choice can be embedded in a column generation framework.

- The number of combinations of speed advices that can be passed at signal points strongly affects the problem size if either there is a large number of consecutive block sections (m_p) between consecutive operation points or many different speeds can be communicated at each signal (SA). Thus, a selection of speed profiles reduces the problem size (cfr. maximum allowed speed profiles for entering and leaving a busy central station area, Caimi, 2009; Fuchsberger, 2012). This, again, can be done using meta-heuristics or column-generation techniques.

4.4 Column generation for the RCG model

This section proposes a column generation approach to solve the RCG model. This approach is expected to give good quality solutions in much shorter computation times than the monolith, given the promising results obtained for other problems of similar size and structure (see Section 3.2.2). Figure 4.5 gives an overview of the data and processes involved. Given the infrastructure, rolling stock, timetable and a prediction of traffic, an initial restricted master problem is built (see Section 4.4.1). Then the problem is solved and the quality of the current best solution is checked (Section 4.4.2). If the requirements to exit the computation are met, the current solution is returned. Otherwise, a loop is started where the prices of the resources are gathered from the current solution (Section 4.4.3) and used within one or more pricing problems to generate new alternative blocking time stairways which may improve the solution of the restricted master (Section 4.4.4). Two different pricing techniques are proposed: one based on precomputed blocking time stairways patterns and a *constructive* one which is based on a shortest path algorithm on a

Figure 4.5: Column generation work-flow (Stop criteria explained in Section 4.4.2).



time expansion of the infrastructure graph. For each technique, an *arc pricing* and a *path pricing* version are presented. Each arc pricing problem looks for suitable blocking time stairways between a pair of operation points, while each path pricing problem tries to generate compatible blocking time stairways for the entire itinerary of each train. Then, these new alternatives are added to the restricted master problem (Section 4.4.5). Given that constraints (4.9)-(4.15) in the rescheduling problem defined in Section 4.2 depend on the decision variables, they need to be recomputed/updated at each loop iteration. Finally, the restricted master problem is solved again and the stop criterion checked (Section 4.4.2). If the conditions to exit the loop are satisfied, the current best schedule is returned, otherwise another iteration is executed.

Note that the LP relaxation of the restricted master problem can be solved in step 2 instead. This provides two advantages. First, it decreases the computational effort of step 2 because solving an LP is usually quite cheap. Second, it might speed up the convergence, as the duals of an LP measure the potential improvement of the objective if the constraints' bounds are enlarged by one unit and thus are usually good estimates of the resources' prices in constrained optimisation problems. However, if the LP relaxation is solved in step 2, it might be necessary to solve the resulting restricted master problem as a MILP after the stop criterion is met to get integral values, which may have a significant "gap" with respect to the optimal objective value of the LP relaxation.

4.4.1 The initial restricted master problem

Given the infrastructure, the timetable, and a prediction of traffic, the RCG model (4.1)-(4.17) is built exploiting the algorithms by Caimi (2009) and Fuchsberger (2012) to define the maximal conflict cliques and the maximal incompatible sets, respectively. In this initial restricted master problem, the blocking time stairways used as decision variables correspond to a limited number of time shifts of the predicted trajectories. This way, the initial restricted master problem corresponds to a sort of "conflict detection" step. The following notation is used to keep track of variables and constrains and to allow model updates when adding new columns. Let $\mathcal{M}(\mathcal{B}) = (x, r, h, c, AS, A, D, C)$ denote the restricted master problem associated with the collection of blocking time stairways $\mathcal{B}$, where

- $x = (x_b)_{b \in \mathcal{B}}$ is the vector of decisions about blocking time stairways in set $\mathcal{B}$;
- $r = (r_{t,i})_{t \in \mathcal{T}, i=1,2,\ldots,n_t}$ is the vector of decisions about scheduling train runs in the current horizon for all trains and sections;
- $h = (h_{t,i})_{t \in \mathcal{T}_p, i=1,2,\ldots,n_t}$ is the vector of run cancellation decisions for all passenger trains and sections;
- $c = (c_{s,t,t'})_{(s,t,t') \in \mathcal{CN}}$ is the set of wait-depart decisions for all scheduled connections;
- $AS = (AS_r)_{r \in \mathcal{R} \setminus \bigcup_{s \in \mathcal{S}} P_s}$ denotes the collection of allocation schemas of all resources;
- $D = (D_{t,p})_{p \in \bigcup_{s \in \mathcal{S}} P_s}$ is the collection of departure lists for all tracks of operation points;
- $A = (A_{t,p})_{p \in \bigcup_{s \in \mathcal{S}}}$ is the collection of arrival lists for all tracks of operation points;
- $C = (c_{indep}, c_{bind}, c_{cliques}, c_{OPocc}, c_{OPtrack}, c_{self-conn}, c_{conn})$ is the set of model constraints where: c_{indep} are all constraints (4.5-4.7), which do not depend on $\mathcal{B}$; c_{bind} are model constraints (4.8) binding x and r variables; $c_{cliques}$ are conflict clique constraints (4.9); c_{OPocc} are operation point tracks conflict constraints (4.10);

$c_{OPtrack}$ denote track consistency at operation points constraints (4.12); $c_{self-conn}$ are self-connection constraints (4.13-4.14); c_{conn} are connection constraints (4.15).

4.4.2 Solve the restricted master problem and stop criteria

The restricted master problem is solved using the commercial solver CPLEX (IBM, 2015). If the solution contains a blocking time stairway for each scheduled run and the delays of all trains is under a given threshold (one minute for ”sharp punctuality” or three minutes for ”soft punctuality”), it is considered of sufficient quality and the algorithm generates the corresponding route setting and departure commands, the speed requests, and customer information. Note that in some cases there is no schedule satisfying the aforementioned quality criteria. Using the LP-relaxed model, these cases are identified when there is no columns with negative reduced cost to be added to the restricted master problem. Unfortunately, this check is not possible with the original MILP formulation. Thus, to prevent an infinite loop, a maximum number of loop iterations is given, such that a solution is provided even if it is not of the quality desired.

4.4.3 Compute the prices of the resources

The goal of the pricing programs is to generate new alternatives which potentially improve the quality of the current restricted master solution. In order for an alternative to be selected by the master, it should not overlap with the blocking time stairways of the other trains. Thus, the pricing problems charge the use of infrastructure resources according to the allocations of the current restricted master solution. This can be done in several ways.

The naivest way to price the resources in RCG is to use the slacks of the conflict constraints (4.9)-(4.10). Remember that conflict constraints (4.9) and (4.10) and proper platform utilisation constraints (4.12-4.14) are associated with resource release and platform occupation times. Thus, given the values of the slack variables of constraints (4.9-4.14) from the last restricted master problem, the price of a candidate blocking time stairway can be approximated with the number of conflict constraints at their limits that a blocking time stairway contributes to.

A most proper way to price resources’ utilisation is to consider the values associated with the dual problem. Given that the concept of duality does not extend to MILP, it is necessary to consider the LP-relaxation of the restricted master problem. Analogously as with the previously mentioned prices, given the values of the duals associated with constraints (4.9-4.14) from last LP-relaxed restricted master problem, the price of a candidate blocking time stairway can be approximated with the dual prices of the conflict constraints that a blocking time stairway contributes to.

4.4.4 Pricing problems

New alternatives to be added to the master problem can either be selected from an existing set of blocking time stairways or constructed from scratch. In the first case, the set may be generated off-line by the simulation based approach of Section 4.2.3. Then, the pricing problem consists of evaluating the precomputed blocking time stairways according to the given resource prices and select the most convenient one(s). In the latter case, train dynamics and route lock and resource release settings must be modelled explicitly in each pricing problem to generate new feasible blocking time stairways with minimal resource cost.

Both pricing techniques can be applied either directly to the master problem (4.1)-(4.17) for generating new x_b variables, or to the following version which contains only the constraints for interdependences of different trains.

$$\min f(y, c) \tag{4.32}$$

$s.t.$

$$\sum_{g \in \mathcal{G}_t} y_g = 1, \quad \forall t \in \mathcal{T} \tag{4.33}$$

$$\sum_{g \ni b, b \in C} y_g \le 1 \quad \forall C \in C^r, r \in \mathcal{R} \setminus \bigcup_{s \in \mathcal{S}} P_s \tag{4.34}$$

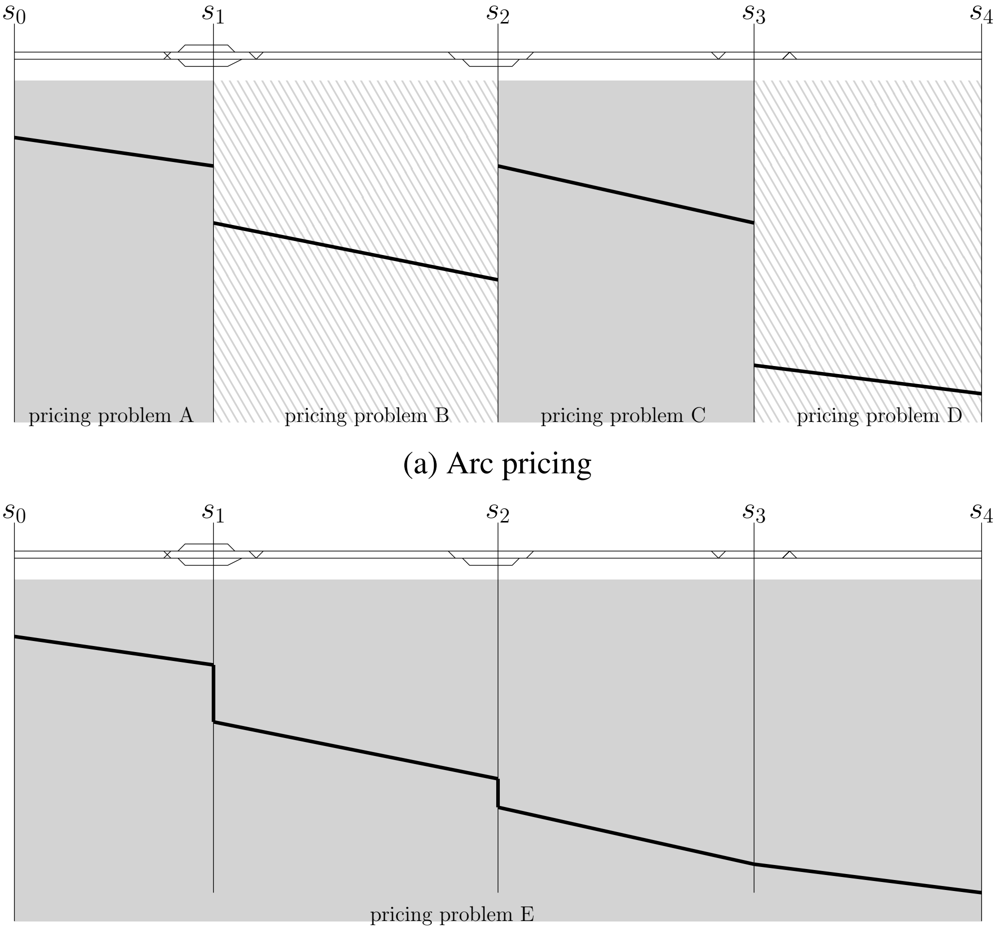
$$0 \le \sum_{\underline{g(p)} \le \alpha} y_g - \sum_{\overline{g(p)} < \alpha} y_g \le 1 \quad \forall \alpha \in \{\underline{g(p)}, g \in \mathcal{G}^p\}, p \in P \tag{4.35}$$

$$\sum_{\underline{g(p)} = \alpha} y_g \le 1 \quad \forall \alpha \in \{\underline{g(p)}, g \in \mathcal{G}^p\}, p \in P \tag{4.36}$$

$$c_{s,t,t'} + \sum_{y \ni b, b \in U} y_g + \sum_{g \ni b, b \in V} y_g \le 2 \quad \begin{array}{l}(U, V) \in \Omega^{t,t'}_{p,p'}, p \in P_{s,t}, \\ p' \in P_{s,t'}, (s, t, t') \in \mathcal{CN}.\end{array} \tag{4.37}$$

$$y_g \in \{0, 1\}, \quad \forall g \in \bigcup_{t \in \mathcal{T}} \mathcal{G}_t \tag{4.38}$$

Figure 4.6: Arc pricing and path pricing



(a) Arc pricing

(b) Path pricing

where $g \in \mathcal{G}_t = \{0,1\}^{n_t} \times \{0,1\}^{n_t} \times \prod_{i=1}^{n_t} \mathcal{B}_{pricing,t,i}$ is an alternative itinerary of a train t through all its scheduled sections; the first n_t dimensions describe whether a train run is scheduled in each section of its itinerary in the current horizon (i.e., the $r_{t,i}$ variables of the original model), the following n_t dimension describe whether the train is cancelled at any section (i.e., the $h_{t,i}$ variables), and the remaining n_t report the sequence of candidate blocking time stairways for the sections (i.e., the x_b variables), y_g is a binary decision variable indicating whether g is selected to be part of the new schedule, $f(y, c)$ denotes a linear combination of y and c variables representing the objective, $\underline{g(p)}$ and $\overline{g(p)}$ are the arrival and departure times of g at platform p, given that g runs through p. Constraints (4.33) impose that exactly one g is chosen for each train. Conflict prevention is enforced by (4.34) for line resources and by (4.354.36) resources at operation points. Constraints (4.37) model connections.

Hereinafter, approaches for generating columns for the original model (4.1)-(4.17) will be referred to as *arc pricing* techniques, while the approaches generating columns for the modified problem (4.32)-(4.37) will be referred to as *path pricing* techniques. Figure 4.6 exemplifies the difference between arc pricing and path pricing.

4.4.4.1 Arc pricing problems with Precomputed blocking time stairways (cgApP)

Let $\mathcal{B}_{pricing}$ be a set of candidates blocking time stairways for the run of a train t between two operation points of its itinerary. This set can be generated from the patterns produced by the simulation-based approach from Section 4.2.3. Promising blocking time stairways for the restricted master problem should not overlap with the ones chosen by the previous iteration. In addition, the new blocking time stairways should be compatible with the ones chosen for the adjacent sections. Applying the slack-variables based resource prices described in Section 4.4.3, the cost of a precomputed blocking time stairway b is:

$$
\begin{aligned}
\Upsilon(b) = & f(b) && (4.39)\\
& + \sum_{r \in b} \sum_{C \in C^r} w_0 \cdot (1 - \lambda_C) \cdot \mathbb{1}_{\tau(C) \in b(r)} && \text{conflict cliques}\\
& + \sum_{C \in C^{pd}} w_1 \cdot (1 - \lambda_C) \cdot \mathbb{1}_{|\tau(C) - \delta(b)| < H} && \text{conflict at departure platform}\\
& + \sum_{C \in C^{pa}} w_2 \cdot (1 - \lambda_C) \cdot \mathbb{1}_{|\tau(C) - \alpha(b)| < H} && \text{conflict at arrival platform}\\
& + \sum_{(U,V) \in \Omega^{t,t}_{pd,pd}} w_3 \cdot \left(1 - \lambda_{(U,V)}\right) \cdot \mathbb{1}_{\delta(b) < \overline{V}} && \text{self-connection at departure platform}\\
& + \sum_{(U,V) \in \Omega^{t,t}_{pa,pa}} w_4 \cdot \left(1 - \lambda_{(U,V)}\right) \cdot \mathbb{1}_{\alpha(b) > \underline{U}} && \text{self-connection at arrival platform}\\
& + \sum_{(U,V) \in \Omega i^t_{pd}} w_5 \cdot \left(1 - \lambda_{(U,V)}\right) \cdot \mathbb{1}_{\delta(b) > \underline{V}} && \text{i.self-connection at departure platform}\\
& + \sum_{(U,V) \in \Omega i^t_{pa}} w_6 \cdot \left(1 - \lambda_{(U,V)}\right) \cdot \mathbb{1}_{\alpha(b) < \overline{U}} && \text{i.self-connection at arrival platform}\\
& + \sum_{(U,V) \in \Omega v^t_{pd}} w_7 \cdot \left(1 - \lambda_{(U,V)}\right) \cdot \mathbb{1}_{v\delta(b) > \underline{V}} && \text{v.self-connection at departure platform}\\
& + \sum_{(U,V) \in \Omega v^t_{pa}} w_8 \cdot \left(1 - \lambda_{(U,V)}\right) \cdot \mathbb{1}_{v\alpha(b) < \overline{U}} && \text{v.self-connection at arrival platform}
\end{aligned}
$$

$$+ \sum_{(U,V)\in\Omega vi^t_{pd}} w_9 \cdot \left(1 - \lambda_{(U,V)}\right) \cdot \mathbb{1}_{v\delta(b)<\overline{V}} \qquad \text{v.self-connection at departure platform}$$

$$+ \sum_{(U,V)\in\Omega_v ipa^t} w_{10} \cdot \left(1 - \lambda_{(U,V)}\right) \cdot \mathbb{1}_{v\alpha(b)>\underline{U}} \qquad \text{v.self-connection at arrival platform}$$

$$- w_{11}\lambda_{(t,pd)} \qquad \text{same platform}$$

$$\forall b \in \mathcal{B}_{pricing}$$

where $f(b)$ denotes the contribution of b to the objective function of the restricted master, pd and pa denote the departure and arrival platforms of b, $C^{p\cdot}$ are the set of constraints of type (4.10), $\tau(C)$ is the resource release time associated with the conflict clique or the reservation time for platform occupation constraint C, λ_C denotes the slack of C, H denotes the minimum headway on platforms, $\lambda_{(U,V)}$ denotes the slack of the constraint associated with the pair of incompatible sets (U, V), and $\lambda_{t,pd}$ is the slack of constraint (4.12). The weights w_{0-11} are usually equal one, but they can be defined in order to prioritise some constraints over the others. The remaining notation is borrowed from Section 4.2. If the LP relaxation of the restricted master is solved in step 2, then the $w_\cdot \cdot (1 - \lambda_\cdot)$ terms can be substituted with the duals (see Section 4.4.3).

After computing the prices of all blocking time stairways, the pricing problem consists into sorting the candidates according to their prices Υ and return the one with the lowest price. After a candidate is added to the master problem, it is removed from the pricing problem. In the following, column generation with Arc pricing using Precomputed blocking time stairways will be referred to as *cgApP*.

4.4.4.2 Path pricing problems with Precomputed blocking time stairways (cgPpP)

Applying Lagrangian multipliers to constraints (4.9), (4.10), (4.11) and (4.15) the objective of the original problem (4.1)-(4.17) becomes the minimisation of function:

$$\mathcal{L}(x, r, h, c, \lambda) = f(x, r, h, c) \tag{4.40}$$

$$- \lambda_{\mathbf{0}} \cdot \left[1 - \sum_{b\in C_0} x_b\right]_{C\in C^r, r\in\mathcal{R}\setminus\bigcup_{s\in\mathcal{S}} P_s} \qquad \text{conflict cliques}$$

$$- \lambda_{\mathbf{1}} \cdot \left[1 - \sum_{\underline{b(p)}\leq\alpha} x_b + \sum_{\overline{b(p)}<\alpha} x_b\right]_{\alpha\in\{\underline{b(p)}, b\in\mathcal{B}^p\}, p\in P_s, s} \qquad \text{conflicts at arrival track}$$

$$- \lambda_{\mathbf{2}} \cdot \left[1 - \sum_{\underline{b(p)}=\alpha} x_b \right]_{\alpha \in \{\underline{b(p)}, b \in \mathcal{B}^p\}, p \in P_s, s} \qquad \text{conflicts at arrival track}$$

$$- \lambda_{\mathbf{3}} \cdot \left[\sum_{\underline{b(p)} \leq \alpha} x_b - \sum_{\overline{b(p)} < \alpha} x_b \right]_{\alpha \in \{\underline{b(p)}, b \in \mathcal{B}^p\}, p \in P_s, s} \qquad \text{conflicts at departure track}$$

$$- \lambda_{\mathbf{4}} \cdot \left[2 - c_{s,t,t'} - \sum_{b \in U} x_b - \sum_{b \in V} x_b \right]_{\substack{(U,V) \in \Omega^{t,t'}_{p,p'}, p \in P_{s,t}, \\ p' \in P_{s,t'}, (s,t,t') \in \mathcal{CN}}} \qquad \text{connection management}$$

where $\lambda_{\mathbf{0,1,2,3,4}}$ denote the Lagrangian multipliers. Assuming that the decision variables $c_{s,t,t'}$ are fixed to given values, it is possible to compute the contribution of each train $t \in \mathcal{T}$ and section $i \in \{1, 2, \ldots, n_t\}$ to $\mathcal{L}$ independently using equation (4.40). In the remaining of this section, this is denoted as $\mathcal{L}([x_b]_{b \in \mathcal{B}_{t,i}}, r_{t,i}, h_{t,i}, \lambda)$.

Setting the Lagrangian multipliers equal to the complement of the slacks multiplied by some weight one obtains the path pricing problem for each course $t \in \mathcal{T}$:

$$\min \sum_{i=1}^{n_t} \mathcal{L}([x_b]_{b \in \mathcal{B}_{pricing,t,i}}, r_{t,i}, h_{t,i}, \lambda) \tag{4.41}$$

s.t.

$$r_{t,i} \leq r_{t,i-1} \qquad \forall i = 2, \ldots, n_t \tag{4.42}$$

$$r_{t,i} + h_{t,i} \leq 1 \qquad \forall i = 1, \ldots, n_t \tag{4.43}$$

$$h_{t,i} \geq h_{t,i-1} \qquad \forall i = 2, \ldots, n_t \tag{4.44}$$

$$r_{t,i} = \sum_{b \in \mathcal{B}_{pricing,t,i}} x_b \qquad \forall i = 2, \ldots, n_t \tag{4.45}$$

$$\sum_{b \in A_{t,p}} x_b - \sum_{b \in D_{t,p}} x_b \leq 0 \qquad \forall p \in P_{\cdot,t}, t \in \mathcal{T} \tag{4.46}$$

$$\sum_{b \in U} x_b + \sum_{b \in V} x_b \leq 1 \qquad \forall (U, V) \in \Omega^{t,t}_{p,p}, p \in P_{s,t} \tag{4.47}$$

$$\sum_{b \in U} x_b + \sum_{b \in V} x_b \leq 1 \qquad \forall (U, V) \in \Omega v^t_p, p \in P_{s,t} \tag{4.48}$$

$$x_b \in \{0, 1\} \qquad \forall b \in \mathcal{B}_{pricing} \tag{4.49}$$

$$r_{t,i}, h_{t,i} \in \{0, 1\} \qquad \forall i = 1, \ldots, n_t \tag{4.50}$$

where $\mathcal{B}_{pricing}$ denotes the set of blocking time stairways that can be chosen during the pricing phase and $\mathcal{B}_{pricing,t,i}$ the ones corresponding to train t and section i. To prevent trajectories to be generated more than once, for each new generated trajectory y^* a new row of the following form is added to the pricing problem (4.41)-(4.50).

$$\sum_{\substack{i\in\{1,\ldots,n_t\\ |r_{t,i}^{y^*}=0\}}} r_{t,i} + \sum_{\substack{i\in\{1,\ldots,n_t\\ |h_{t,i}^{cum,y^*}=1\}}} (1-h_{t,i}) + \sum_{\substack{b\in\mathcal{B}_{pricing}\\ |x_b^{y^*}=1\}}} (1-x_b) \geq 1 \tag{4.51}$$

where $r_{t,i}^{y^*}$ indicates whether the i^{th} run of train t is scheduled by trajectory y^* and $h_{t,i}^{cum,y^*}$ whether it is cancelled, while $x_b^{y^*}$ indicates whether blocking time stairway b was part of y^*. Similarly as for the arc pricing case, the formulation can be slightly modified to apply duals based prices as defined in Section 4.4.3.

In the following, column generation with Path pricing using Precomputed blocking time stairways will be referred to as *cgPpP*.

4.4.4.3 Arc pricing problems Constructing blocking time stairways (cgApC)

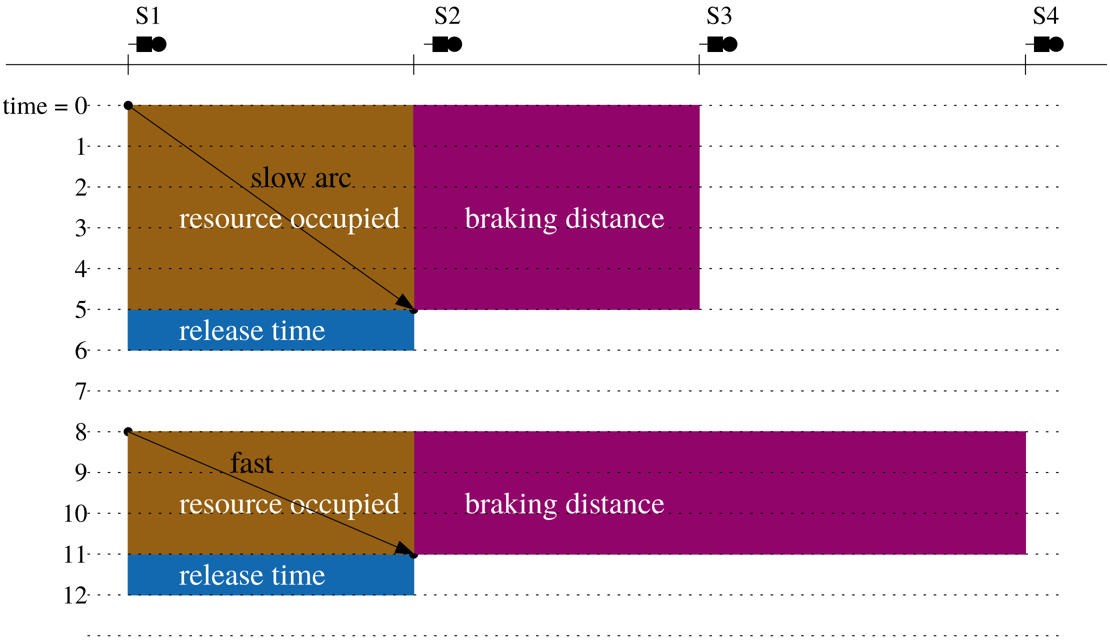
The constructive arc pricing is formulated as a shortest path problem on a directed graph. The graph is a time-expansion of the infrastructure between two consecutive operation points on the train's itinerary. The nodes are the joining points of adjacent resources. Two space-time nodes are linked if their space coordinates correspond to the end points of the same resource and the difference between their time coordinates is at least the minimum running time of the considered rolling stock on that resource. The direction of each arc reflects the travel direction. The length of the arcs is given by the cumulative price of all resources blocked by the corresponding train movements, as shown in Figure 4.7:

- the resource corresponding to the infrastructure between the space coordinates of the end nodes during the time span between the time coordinates (e.g., the brown areas in Figure 4.7, they spans from space coordinate $S1$ to $S2$ and from time coordinates 0 (slow arc) and 8 (fast arc) to 5 and 11 (respectively));

- and for an additional release time needed by the interlocking system (e.g., blue area, space between $S1$ and $S2$, time between 5 and 6 for slow arc and between 11 and 12 for fast arc);

- the resources on the braking path given by the implicitly defined speed (e.g., purple area spanning over next block for slow arc and

Figure 4.7: Elements contributing to arc lengths



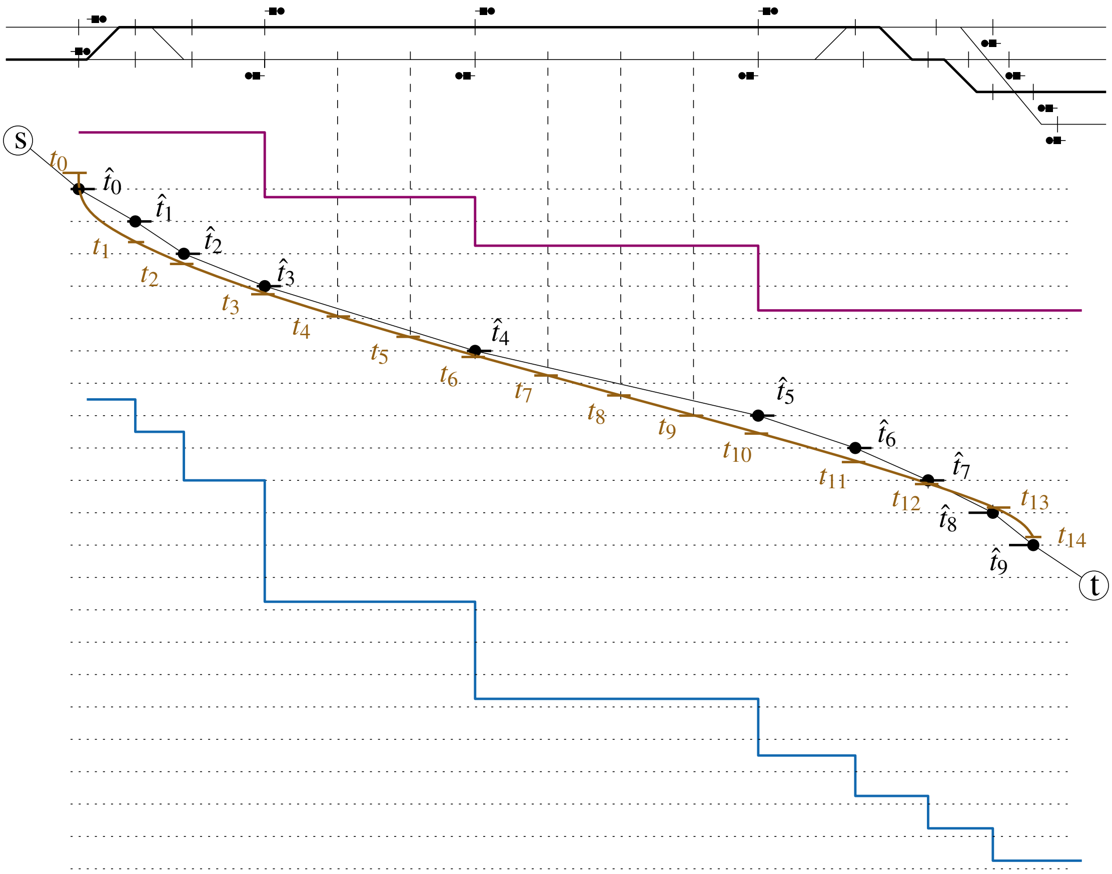
over next two blocks for fast arc); if several paths are possible, the average of all path costs is used.

A source node indicates the entrance point of the train into the considered section. If the train has already entered the section, the source is linked to all time expansions of starting point of the next resource on its path. Otherwise it is linked to all starting points of the routes leaving a track of the operation point. In the first case, the lengths of these arcs are obtained by the prices of the resources currently occupied. In the latter case, they are given by the prices of the resource of the operation point track they represent. A target node indicates the exit point of the train from the considered section. This node is connected to all time expansions of the last infrastructure traversed before leaving the section. The lengths of these arcs are the sum of the resource price of the arrival track and the arrival delay at the successive operation point.

A new promising alternative is built from the shortest source-target path. Note that train dynamics should be ensured, in particular maximal acceleration and braking ratios. Thus, a feasible trajectory is defined by the following system which aims at minimising the difference between the resulting trajectory and the space-time coordinates of the shortest path.

Let $(\hat{s}_0, \hat{t}_0), (\hat{s}_1, \hat{t}_1), \ldots, (\hat{s}_n, \hat{t}_n)$ be the nodes of the shortest path and $s_0, s_1, \ldots, s_N$ a discretisation of the route defined by the shortest source-target path. These points are defined such that the infrastructure between s_i and s_{i+1} is part of same resource and is shorter than a given parameter Δs_{max}. The parameter Δs_{max} is used to split the infrastructure along the

Figure 4.8: Post-processing for constructive pricing: the shortest source-target path (black nodes and arcs) is transformed into a feasible trajectory (brown) using additional discretisation points and minimising the time difference (vertical) at the shortest path nodes, then the corresponding blocking time stairway is generated (reservation in purple, release in blue).



train movement into pieces that are small enough to assume a uniform acceleration. A function $\sigma : \{0, 1, \ldots, n\} \mapsto \{0, 1, \ldots, N\}, i \mapsto \sigma(i)$ maps the indexes of the nodes of the shortest paths to the corresponding discretisation indexes (i.e., $\hat{s}_i = s_{\sigma(i)}$). Then, the following system defines a promising feasible trajectory

$$\min \quad \textstyle\sum_{i=0}^{n} \left(t_{\sigma(i)} - \hat{t}_i\right)^2 \tag{4.52}$$

$$s.t \quad t_i - t_{i-1} \geq rt_{min}(s_{i-1} \rightarrow s_i) \qquad i = 1, 2, \ldots, N \tag{4.53}$$

$$-d_{max} \cdot (t_i - t_{i-1}) \leq v_i - v_{i_1} \qquad i = 1, 2, \ldots, N \tag{4.54}$$

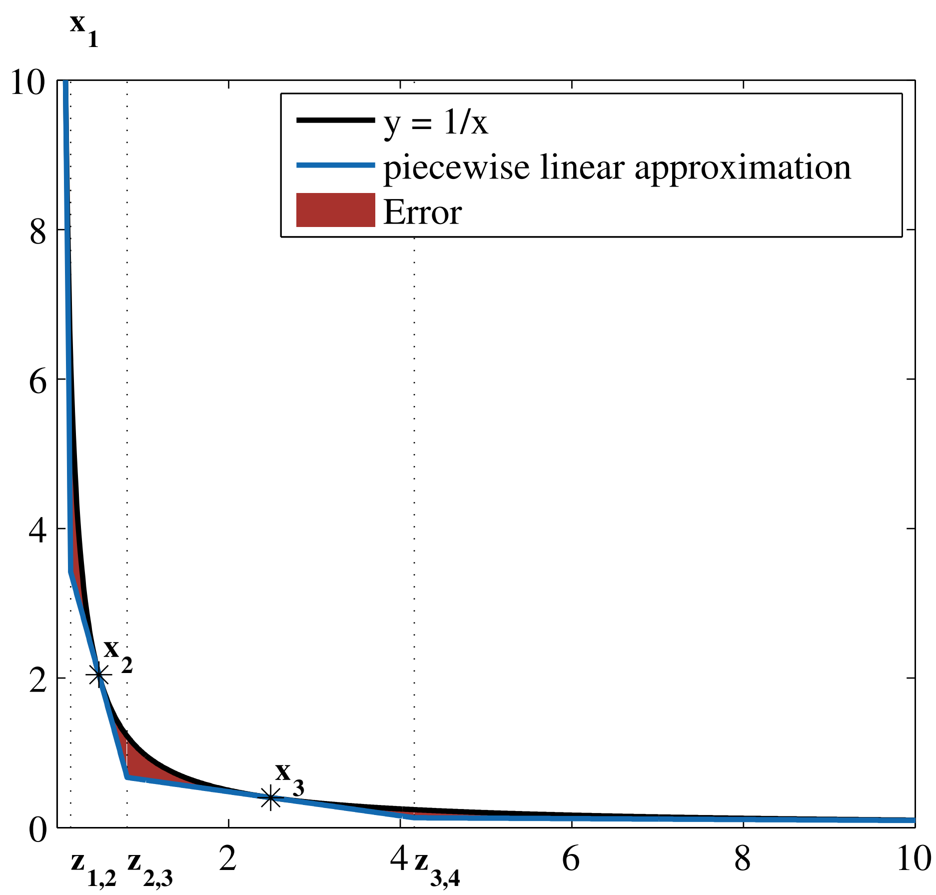
$$v_i - v_{i-1} \leq a_{max} \cdot (t_i - t_{i-1}) \qquad i = 1, 2, \ldots, N \tag{4.55}$$

$$\frac{v_i - v_{i-1}}{2} = \frac{s_i - s_{i-1}}{t_i - t_{i-1}} \approx (s_i - s_{i-1}) \cdot PLA(t_i - t_{i-1}) \qquad i = 1, 2, \ldots, N \tag{4.56}$$

$$t_i, v_i \in \mathbb{R}_{\geq 0} \qquad i = 0, 1, \ldots, N \tag{4.57}$$

where (4.52) aims at minimising the overall square error between the final times and the ones given by the shortest path; (4.53) ensure that the minimum running time from s_i to s_{i+1} ($rt_{min}(s_i \rightarrow s_{i+1})$) is respected; (4.54) limit the speed reduction in each partition according to the maximal deceleration ratio of the rolling stock (d_{max}); (4.55) limit the speed increase in each partition according to the maximal acceleration ratio of the rolling stock (a_{max}); and (4.56) link time and speed variables using the average speed formula. Note that the latter constraint can be linearised with a piecewise linear approximation of function $\frac{1}{\Delta t}$. There are several methods to approximate a function with a piecewise linear formula (see, e.g., Lin et al., 2013). Here, the function is approximated with a set of tangents which minimises the area between the hyperbole and its approximation as shown in Figure 4.9. This choice is motivated by the fact that fixing the number of segments, the lower and the upper points of the area of interest $[x_1, x_N]$, the breakpoints $z_{k,k+1}$ and the slopes a_k are defined analytically

Figure 4.9: Piecewise linear approximation of hyperbole using 6 tangents. The breakpoints $z_{k,k+1}$ and the hyperbole points defining the tangents' slopes x_k^* are chosen to minimise the area between the two curves.



by the following formulae.

$$x_k^* = x_N^{\frac{k-1}{N-1}} \cdot x_0^{\frac{N-k}{N-1}} \tag{4.58}$$

$$a_k = -\frac{1}{x_k^{*2}} \tag{4.59}$$

$$z_{k,k+1} = 2 \cdot \frac{x_k^* \cdot x_{k+1}^*}{x_k^* + x_{k+1}^*} \tag{4.60}$$

Finally, the blocking time stairway for the resulting trajectory can be computed according to the time points t_i, the speeds v_i, the required braking distance at each point s_i, the train length and additional time supplements for route setting and release (see Algorithm 4.1). Then, the new blocking time stairway can be associated with a decision variable to be inserted into the restricted master problem. In the following, column generation with Arc pricing Constructing new blocking time stairways will be referred to as *cgApC*.

4.4.4.4 Path pricing problems Constructing blocking time stairways (cgPpC)

The constructive path pricing is very similar to its arc counterpart, but here the time-expanded graph of the infrastructure spans the entire train itinerary. Again, the nodes are the joining points of adjacent resources and arcs reflect possible train movements, ensuring minimum running times and minimum dwell times. The only difference is that the contribution to the objective function for intermediate operation points has to be added to the arcs corresponding to the operation point's tracks instead of the source and target arcs. In the following, column generation with Path pricing Constructing new blocking time stairways will be referred to as *cgPpC*.

4.4.5 Add new columns to a RCG model

New columns can be added to the restricted master problem exploiting the techniques outlined by Caimi (2009) and Fuchsberger (2012) for building the problem. These techniques were developed to time-efficiently compute maximal conflict cliques and incompatible sets. In the first case, the events of each allocation schema are sorted in ascending order and the conflict cliques are defined at resource release times that are immediately preceded by at least one resource reservation item (Algorithm 2 by Caimi, 2009). In the second case, the events of all operation point departure and arrival lists are sorted and incompatible sets are defined iteratively

Algorithm 4.1 Compute blocking time stairway for given trajectory

Input: trajectory $(s_i, t_i, v_i)_{i=0,\dots,n}$, routes $(r_j)_{j=1,\dots,m}$, maximum deceleration ratio a_{dec}, reservation and release time supplements τ_{res}, τ_{rel}

Output: blocking time stairway $b = (resource, [reservation, release])_{resource \in \cup_j r_j}$

for $j = 1, \dots, m$ **do**
  set all reservation times of resources in r_j to ∞
  set all release times of resources in r_j to -1
end for
$routesOnBrakingPath := \emptyset$
$routeLength := (0.0)$
for $i = n - 1, \dots, 1$ **do**
  $distToStop := \frac{v_i^2}{2 * a_{dec}}$
  $routeLength[0] := routeLength[0] + s_{i+1} - s_i$
  $current_{reservation} := t_i - \tau_{res}$
  r_{it} :=First route in $routesOnBrakingPath$
  while $r_{it} < routesOnBrakingPath.end()$ and $distToStop > 0$ **do**
    set reservation time of all resources in route r_{it} to $current_{reservation}$
    $distToStop := distToStop - routeLength[r_{it}]$
    r_{it} :=Next route in $routesOnBrakingPath$
  end while
  Find minimal index $currentRelease$ such that $s_{currentRelease-1} \geq s_i + trainLength$
  set reservation time of the resource in s_i-s_{i+1} to $t_{currentRelease} + \tau_{rel}$

  if s_i is beginning of a new route rr **then**
    $routesOnBrakingPath := \begin{pmatrix} rr \\ routesOnBrakingPath \end{pmatrix}$
    $routeLength := \begin{pmatrix} 0.0 \\ routeLength \end{pmatrix}$
  end if
end for
return b

by checking the compatibility of pairs of event from the lists (Algorithm 3.1 by Fuchsberger, 2012). Here, new columns are added to the problem by inserting the respective block reservation and release times in the allocation schemas and the departure and arrival times in the lists of the corresponding resources. Then, either the existing constraints are updated or new ones are defined.

The conflict clique constraints (4.9) can be updated using the following algorithm 4.2. The idea is to insert the reservation and release times of each resource in the corresponding allocation schema, update all conflict cliques associated with the times between the newly inserted allocations, and eventually define new conflict cliques. Figure 4.10 shows two examples.

Similarly, the departure and arrival times linked with the new column are used by algorithms 4.3-4.4 to update the conflicts at operation points constraints (4.10) ans the same platform constraints (4.12). The idea is to find the position of the time event in the corresponding allocation schema, update all constraints associated with later times, and define a new constraint in the case of an arrival event. Figure 4.11 shows two examples. Note that in the allocation schemas, if items are associated to the exact same time, the reservation is set before the release and the arrival before the departure.

The incompatible sets generating constraints (4.13)-(4.15) can be updated using the following Algorithms 4.5-4.6. The idea is exactly the same as of Fuchsberger (2012) but when a new maximal set is found which contains a previously defined one the old one is changed into the new one. Figure 4.12 shows an example for a connection constraint.

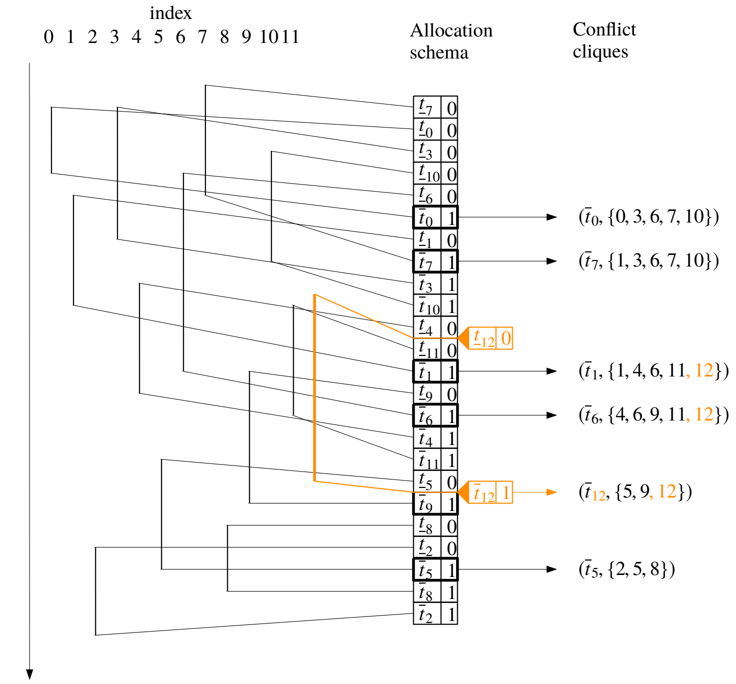
Algorithm 4.2 Update conflict cliques for new blocking time interval

Input: blocking time interval $b_r = \left[\underline{b}_r, \overline{b}_r\right]$ of resource r; allocation schema AS_r of resource r; conflict clique constraints $c_{cliques}$; boolean variable x_b

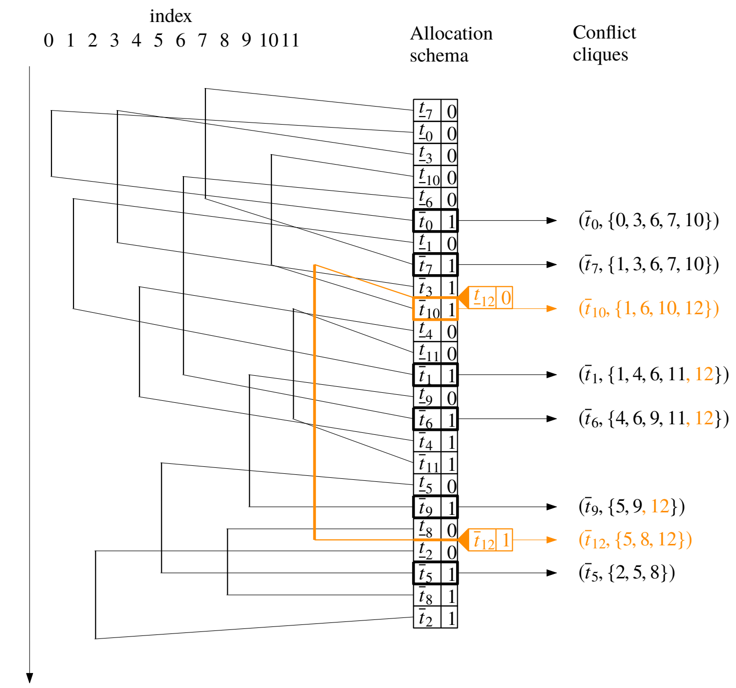
Output: updated conflict cliques $c_{cliques}$

$i := 0$
while $i \neq$ **length**(AS_r) **and** $AS_r[i] < \underline{b}_r$ **do**
  $i := i + 1$
end while
Insert $\underline{b}_r$ into AS_r at position i
if ($i = 0$ **or** $AS_r[i-1]$ is release) **and** $AS_r[i+1]$ is release **then**
  define conflict clique constraint for release element $AS_r[i+1]$
end if
while $i \neq$ **length**(AS_r) **and** $AS_r[i] \leq \overline{b}_r$ **do**
  if $\exists \tilde{c} \in c_{cliques}$ defined by $AS_r[i]$ **then**
    $\tilde{c} := \tilde{c} + x_b$
  end if
  $i := i + 1$
end while
Insert $\overline{b}_r$ into AS_r at position i
if $i = 0$ **or** $AS_r[i-1]$ is release **then**
  do nothing
else if $AS_r[i+1]$ is release of constraint $\tilde{c} \in c_{cliques}$ **then**
  $\tilde{c} := \tilde{c} + x_b$
else
  define conflict clique constraint for release element $AS_r[i]$
end if
return $c_{cliques}$.

Figure 4.10: Updating conflict cliques. On the left hand side the blocking time of a resource by different trajectories, in the centre the allocation schema of the resource, and on the right the conglict clique constraints. The orange parts correspond to a newly inserted column



(a) Conflict cliques with times between $\underline{t}_{12}$ and $\bar{t}_{12}$ are updated



(b) In addition, two new conflict clique constraints are defined

Algorithm 4.3 Update operation point track constraints for new departure

Input: departure time δ; departure track p; allocation schema AS_p of p; operation point track conflict constraints c_{OPocc}; track-consistency constraint $c_{OPtrack}(p)$ for track p; boolean variable x_b

Output: updated operation point track conflict constraints c_{OPocc} and track-consistency constraint $c_{OPtrack}(p)$

$c_{OPtrack}(p) := c_{OPtrack}(p) - x_b$
$i := 0$
while $i \neq$ **length**(AS_p) **and** $AS_p[i] <= \delta$ **do**
  $i := i + 1$
end while
Insert δ into AS_p at position i
while $i \neq$ **length**(AS_p) **do**
  if $\exists \tilde{c} \in c_{OPocc}$ defined by $AS_p[i]$ **then**
    $\tilde{c} := \tilde{c} - x_b$
  end if
  $i := i + 1$
end while
return $c_{OPocc}, c_{OPtrack}(p)$.

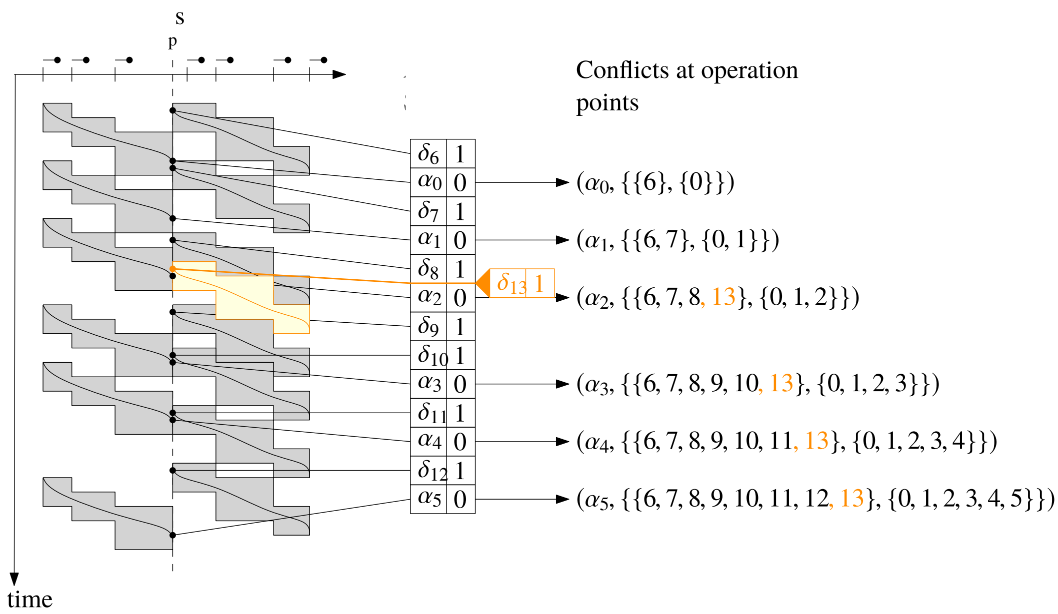
Algorithm 4.4 Update operation point track constraints for new arrival

Input: arrival time α; arrival track p; allocation schema AS_p of p; operation point track conflict constraints c_{OPocc}; track-consistency constraints $c_{OPtrack}(p)$ for track p; boolean variable x_b

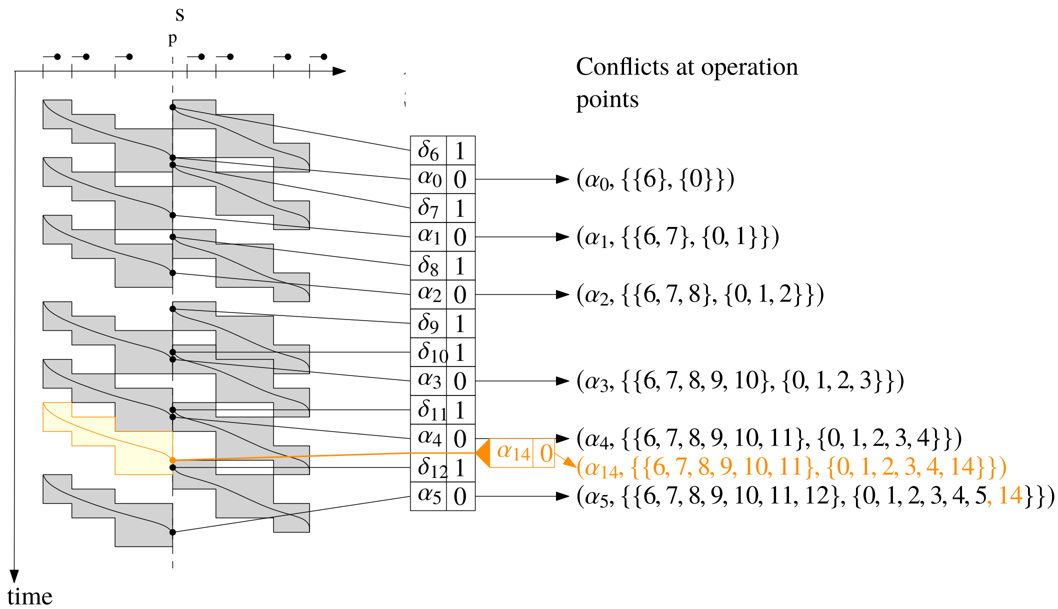
Output: updated operation point track conflict constraints c_{OPocc} and track-consistency constraints $c_{OPtrack}(p)$

$c_{OPtrack}(p) := c_{OPtrack}(p) - x_b$
$i := 0$
while $i \neq$ **length**(AS_p) **and** $AS_p[i] < \alpha$ **do**
  $i := i + 1$
end while
Insert α into AS_p at position i
define operation point track conflict constraint for element $AS_p[i]$
while $i \neq$ **length**(AS_p) **do**
  if $\exists \tilde{c} \in c_{OPocc}$ defined by $AS_p[i]$ **then**
    $\tilde{c} := \tilde{c} + x_b$
  end if
  $i := i + 1$
end while
return $c_{OPocc}, c_{OPtrack}(p)$.

Figure 4.11: Updating conflict at operation points constraints. On the left hand side the blocking time stairways arriving at and departing from a resource, in the centre the allocation schema of the resource, and on the right the sets of indices for conflicting allocations. The orange parts correspond to a newly inserted column.



(a) Departure event



(b) Arrival event (defines new constraint)

Algorithm 4.5 Update maximal incompatible sets for departure

Input: departure time δ; arrival list $A_{t,p}$ of feeder train at given platform; departure list $D_{t',p'}$ of current train from given platform; maximal incompatible sets $\Omega^{t,t'}_{p,p'}$; minimal connecting time m; index b of current column

Output: updated maximal incompatible sets $\Omega^{t,t'}_{p,p'}$

$i := 0$
$lastSetIndex := 0$
while $i < \textbf{length}(D_{p',t'})$ **and** $D_{t',p'}[i] < \delta$ **do**
    if $D_{t',p'}[i]$ defines maximal incompatible set **then**
        $lastSetIndex := lastSetIndex + 1$
    end if
    $i := i + 1$
end while
Insert δ into $D_{t',p'}$ at position i
if $\alpha(lastSetIndex) + m > \delta$ **then**
    $\delta(lastSetIndex) := \delta$
    $V(lastSetIndex) := V(lastSetIndex) \cup \{b\}$
else
    $j :=$ index of $\alpha(lastSetIndex)$ in $A_{t,p}$
    $j := j + 1$
    while $j <$ index of $\alpha(lastSetIndex + 1)$ in $A_{t,p}$ **do**
        if $A_{t,p}[j] + m > \delta)$ **then**
            Define new incompatible set for index $(A_{t,p}[j], \delta)$ and place it at position $lastSetIndex$
            break
        end if
        $j := j + 1$
    end while
end if
$lastSetIndex := lastSetIndex + 1$
while $lastSetIndex < \textbf{length}(\Omega^{t,t'}_{p,p'})$ **do**
    $V(lastSetIndex) := V(lastSetIndex) \cup \{b\}$
    $lastSetIndex := lastSetIndex + 1$
end while
return $\Omega^{t,t'}_{p,p'}$.

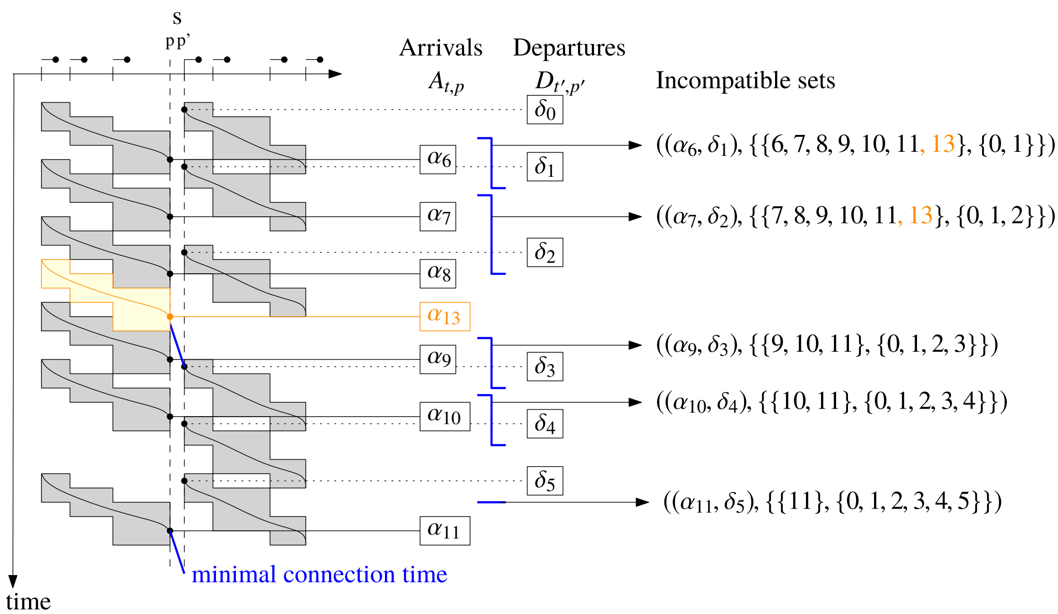
Algorithm 4.6 Update maximal incompatible sets for arrival

Input: arrival time α; arrival list $A_{t,p}$ of current time at given platform; departure list $D_{t',p'}$ of destination train from given platform; maximal incompatible sets $\Omega^{t,t'}_{p,p'}$; minimal connecting time m; index b of current column

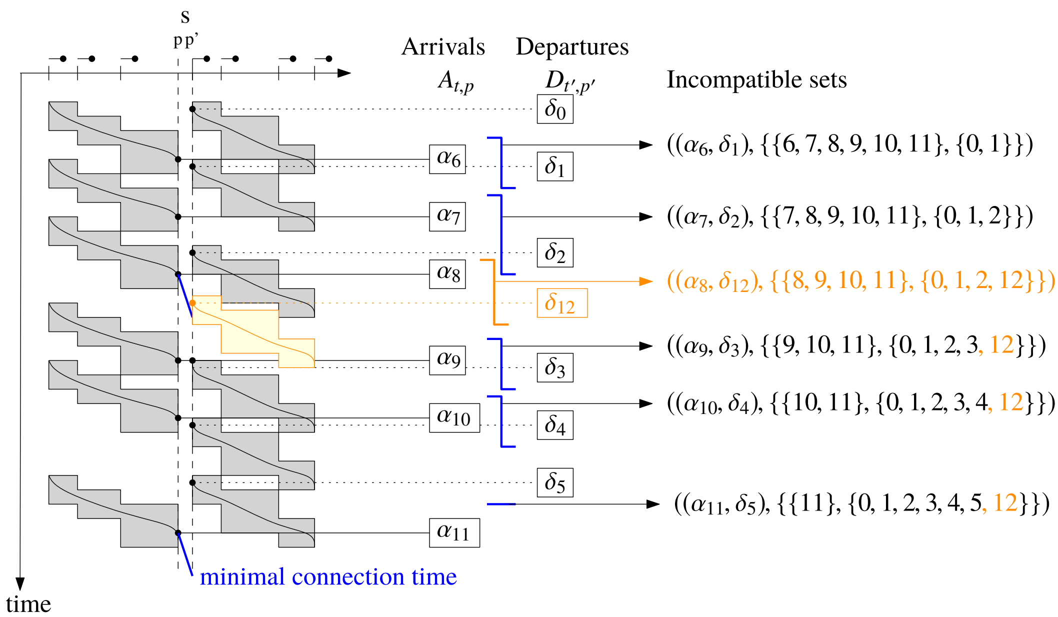
Output: updated maximal incompatible sets $\Omega^{t,t'}_{p,p'}$

$i :=$ **length**$(A_{p,t})$
$lastSetIndex :=$ **length**$(\Omega^{t,t'}_{p,p'})$
while $i > 0$ **and** $A_{t,p}[i-1] > \alpha$ **do**
    $i := i - 1$
    if $A_{t,p}[i]$ defines maximal incompatible set **then**
        $lastSetIndex := lastSetIndex - 1$
    end if
end while
Insert α into $A_{t,p}$ at position i
if $\alpha + m > \delta(lastSetIndex)$ **then**
    $\alpha(lastSetIndex) := \alpha$
    $U(lastSetIndex) := U(lastSetIndex) \cup \{b\}$
else
    $j :=$ index of $\delta(lastSetIndex)$ in $D_{t',p'}$
    $j := j - 1$
    while $j >$ index of $\delta(lastSetIndex - 1)$ in $D_{t',p'}$ **do**
        if $\alpha + m > D_{t',p'}[j])$ **then**
            Define new incompatible set for index $(\alpha, D_{t',p'}[j])$ and place it at $lastSetIndex$
            break
        end if
        $j := j - 1$
    end while
end if
while $lastSetIndex > 0$ **do**
    $lastSetIndex := lastSetIndex - 1$
    $U(lastSetIndex) := U(lastSetIndex) \cup \{b\}$
end while
return $\Omega^{t,t'}_{p,p'}$.

Figure 4.12: Updating connection constraints at station s. On the left hand side the blocking time stairways arriving at and departing from two tracks in s, in the centre the departure and arrivals lists of the tracks, and on the right the sets of maximal incompatible sets. The orange parts correspond to a newly inserted column.



(a) Departure event



(b) Arrival event (defines new constraint)

The following Algorithm 4.7 summarises how to insert a new column into the restricted master problem.

Algorithm 4.7 Insert a new column into restricted master problem

Input: blocking time stairway b for train t section s, restricted master problem $\mathcal{M}(\mathcal{B})$
Output: updated restricted master problem $\mathcal{M}(\mathcal{B} \cup \{b\})$

x_b := **new** boolean variable
$x := x \cup \{x_b\}$
$c_{bind}(t, s) := c_{bind}(t, s) + x_b$
for all blocking time intervals $b_r \in b$ **do**
    Update $c_{cliques}$ with Algorithm 4.2
end for
Update $c_{OPtrack}$ and c_{OPocc} with Algorithms 4.3-4.4
Update $c_{self-conn}$ and c_{conn} with Algorithms 4.5-4.6
$\mathcal{M}(\mathcal{B} \cup \{b\}) = (x, r, h, c, AS, A, D, C)$
return $\mathcal{M}(\mathcal{B} \cup \{b\})$.

4.5 Non-hierarchical coordination framework

This section proposes a non-hierarchical framework for distributed railway traffic rescheduling with the RCG model (4.1)-(4.17). In fact, even assuming that the column generation framework presented in the previous section improves the resolution of the RCG formulation considerably, the computational effort of network-wide railway traffic rescheduling with a single RCG formulation will still be way above the capability of state-of-the-art computers (and probably even of future ones). Thus, as suggested by the literature review in Section 3.3, the geographic extension and the amount of traffic considered by a rescheduling model have to be limited to get problems of tractable sizes. Assuming that the size of the rescheduling horizon and the number of trains to be considered by a RCG model are inherited by the yearly timetable, which is fixed, it is sensible to partition the rescheduling problem geographically in condensation and compensation zones as described in Section 4.1.

The notation from Table 4.1 is used hereinafter. The problem of rescheduling traffic in a network $\mathcal{N}$ with a set of operation points $\mathcal{S}$ is denoted as

$$\Phi = \min_{(x,r,h,c)\in X} f(x, r, h, c) \tag{4.61}$$

where X denotes the set of values of (x, r, h, c) satisfying constraints

(4.1)-(4.17). Three formulations for partitioning (4.61) into a set of local rescheduling problems $\{\phi^i\}_i$ satisfying Definition 4.2 are proposed. Each ϕ^i is based on a RCG model for the sub-network $\mathcal{N}_i$.

The central topic of this section is the coordination of the local rescheduling problems to get a feasible global schedule. In fact, when local rescheduling of adjacent areas is executed separately, consistent track, time, and speed assignments at portals are not guaranteed. Even if a common global timetable is used as reference by all zones, explicit coordination is often required (see, e.g., the results in Section 5.8). Coordination can be achieved in different ways: in non-hierarchical approaches adjacent areas exchange values at boundaries directly with each other to ensure coordination; while in multilevel approaches an upper level enforces coordination at portals by fixing common time, speed and track values. This section focuses on non-hierarchical approaches. Despite the relevance of multilevel coordination approaches, they are out of the scope of the current work. Thus, they will not be further investigated.

Section 4.5.1 proposes a formulation allowing trains to be stored at the portals between neighbouring zones, while Section 4.5.2 and 4.5.3 present simpler formulations requiring that trains travel through portals without stopping.

4.5.1 Subgradient method formulation with wait possibilities at portals

The formulation presented hereinafter allows every operation point in the network to be used as a boundary between zones. Thus, it enables a large number of different problem partitioning approaches.

Definition 4.4 *A* valid network partitioning *of a network $\mathcal{N}$ consists of a set $\mathcal{N}_0$, $\mathcal{N}_1$, …, $\mathcal{N}_m$ of zones of $\mathcal{N}$ that satisfy (R1-R2) of Definition 4.1 such that*

(R3) each sub-network $\mathcal{N}_i$ is either a condensation or a compensation zone with either simple or complex topology,

(R4) no portal is set at operation points where connections are scheduled, i.e., $\forall s \in \mathcal{S}_{portals} \nexists t, t' \in \mathcal{T}\ s.t. (t, t', s') \in C\mathcal{N}$.

Note that each train run from an operation point to the next one can be mapped to exactly one sub-network of a valid network partitioning. In addition, requiring that no connection is scheduled at portals (R4), each connection can be also mapped to exactly one sub-network of a valid network partitioning. It follows that each decision variable of the RCG model (4.1)-(4.17) can be linked to exactly one sub-network $\mathcal{N}_{.}$ of a valid network partitioning. Then, the railway traffic rescheduling problem (4.61) can be reformulated as

$$\Phi = \min_{(x,r,h,c)\in X} \sum_{i=0}^{m} f^i(x^i, r^i, h^i, c^i) \tag{4.62}$$

where x^i, r^i, h^i, c^i denote the decision variables linked to each sub-network $\mathcal{N}_i$, and $f^i(.)$ denotes the part of the objective function depending on them.

For a problem similar to (4.62) Terelius et al. (2011) propose a "decentralized Multi-Agent Optimisation via Dual Decomposition". Their main idea is to distribute the problem among $m + 1$ agents. They add a copy of each decision variable for each agent and impose all copies of each decision variable to be equal to each other. These equality constraints are dualised using Lagrangian multipliers. The copies of the decision variables and the terms of the objective function based on them are assigned to different agents that iteratively solve their own optimisation problem, exchange the values of the decision variables with the neighbours, and update the values of the Lagrangian multipliers using the values obtained by the previous computation and the ones received from the neighbours.

Here, given that all decision variables and most of the constraints can be linked to exactly one sub-network, it seems unduly to add copies of all decision variables and Lagrangian multipliers for the respective equality constraints. In addition, the RCG model already includes several constraints that ensure consistency and conflict freedom at every operation point, including portals. Hereinafter, these constraints are referred to as *complicating constraints*. The idea is to apply a Lagrangian relaxation to the complicating constraints and get

$$\begin{aligned} \min_{(x,r,h,c)\in X'} & \sum_{i=0}^{m} f^i(x^i, r^i, h^i, c^i) \\ & + \sum_{s\in\mathcal{S}_{portals}} \Bigg(\lambda_A^s A^s r + \lambda_C^s C^s h + \lambda_F^s \left[\begin{pmatrix} F^s \\ -F^s \end{pmatrix} x - \begin{pmatrix} \mathbb{1} \\ \mathbb{0} \end{pmatrix} \right] \\ & \qquad + \lambda_G^s \left[G^s x - \mathbb{1} \right] + \lambda_H^s H^s x \\ & \qquad + \lambda_J^s \left[J^s x - \mathbb{1} \right] + \lambda_K^s \left[K^s x - \mathbb{1} \right] \Bigg) \\ & \lambda_{:} \geq 0 \end{aligned} \tag{4.63}$$

where X' denotes the set defined by (4.1)-(4.17) without the relaxed constraints, A^s denotes the coefficients of variables r in constraints of type (4.5) related to portal $s \in \mathcal{S}_{portal}$, C^s denotes the coefficients of variables h in constraints of type (4.7) related to portal s, F^s denotes the coefficients of variables x in conflict constraints of type (4.10) related to portal s, G^s denotes the coefficients of variables x in conflict constraints of type (4.11) related to portal s, H^s denotes the coefficients of variables x in consistency constraints of type (4.12) related to portal s, J^s denotes the coefficients of variables x in consistency constraints of type (4.13) related to portal s, and K^s denotes the coefficients of variables x in consistency constraints of type (4.14) related to portal s.

Given the linearity of the complicating constraints, the terms of (4.63) can be rearranged. Note that X' is the product of $(m+1)$ disjoint sets X_0, X_1,…,X_m that represent the local RCG models (4.1)-(4.17) in the sub-networks $\mathcal{N}_0$,$\mathcal{N}_1$, …, $\mathcal{N}_m$. Then, the rescheduling problem (4.62) is equivalent to finding a solution to

$$\Phi = \max_{\lambda} \sum_{i=0}^{m} \phi^i(\lambda) \tag{4.64}$$

where $\phi^i(\lambda)$ denotes the local rescheduling problem in zone $\mathcal{N}_i$ defined as

$$\begin{aligned}
\phi^i(\lambda) = \min_{(x^i,r^i,h^i,c^i)\in X_i} & \Big\{ f^i\ (x^i, r^i, h^i, c^i) \\
& + \sum_{s\in\partial\mathcal{N}_i} \Big(\lambda_A^s A^{si} r^i + \lambda_C^s C^{si} h^i \\
& \qquad + \lambda_F^s \left[\begin{pmatrix} F^{si} \\ -F^{si} \end{pmatrix} x^i - \begin{pmatrix} \mathbb{1} \\ \mathbb{0} \end{pmatrix} \right] \\
& \qquad + \lambda_G^s \left[G^{si} x^i - \mathbb{1} \right] + \lambda_H^s H^{si} x^i \\
& \qquad + \lambda_J^s \left[J^{si} x^i - \mathbb{1} \right] + \lambda_K^s \left[K^{si} x^i - \mathbb{1} \right] \Big) \Big\}
\end{aligned} \tag{4.65}$$

where A^{si}, C^{si}, F^{si}, G^{si}, H^{si}, J^{si}, K^{si} denote the columns of A^s, C^s, F^s, G^s, H^s, J^s, K^s corresponding to the variables linked to zone $\mathcal{N}_i$.

Lagrangian dual problems such as (4.64) are usually solved using the subgradient method (see, e.g., Terelius et al., 2011). The subgradient method is an iterative approach that starts with some initial estimate of the Lagrangian multipliers $\lambda(0)$ and updates it at each iteration with the current solutions to local problems $\phi^{\cdot}$ according to

$$\lambda(k+1) = \lambda(k) + w \cdot q(k) \tag{4.66}$$

where $\lambda(k)$ denotes the value of λ used at the k^{th} iteration to solve the local problems ϕ^i, $w(k)$ indicates the step size (see, e.g., Nedic and Ozdaglar, 2009), and $g(k)$ indicates a subgradient of the function $Q(\lambda) = \sum_{i=0}^{m} \phi^i(\lambda)$ at $\lambda(k)$. The subgradient method with constant step size is guaranteed to converge to some value very close to the optimum (see e.g, Boyd et al., 2003).

Differentiating (4.63) with respect to each component of λ, one gets the following refinements of (4.66):

$$\lambda_A^s(k+1) = \lambda_A^s(k) + wA^s r(k) \tag{4.67}$$
$$\lambda_C^s(k+1) = \lambda_C^s(k) + wC^s h(k) \tag{4.68}$$
$$\lambda_F^s(k+1) = \lambda_F^s(k) + w\left[\begin{pmatrix} F^s \\ -F^s \end{pmatrix} x(k) - \begin{pmatrix} \mathbb{1} \\ \mathbb{0} \end{pmatrix}\right] \tag{4.69}$$
$$\lambda_G^s(k+1) = \lambda_G^s(k) + w\left[G^s x(k) - \mathbb{1}\right] \tag{4.70}$$
$$\lambda_H^s(k+1) = \lambda_H^s(k) + wH^s x(k) \tag{4.71}$$
$$\lambda_J^s(k+1) = \lambda_J^s(k) + w\left[J^s x(k) - \mathbb{1}\right] \tag{4.72}$$
$$\lambda_K^s(k+1) = \lambda_K^s(k) + w\left[K^s x(k) - \mathbb{1}\right] \tag{4.73}$$

where $(x(k), r(k), h(k))$ is the solution to the local problems $\phi^{\cdot}(\lambda(k))$.

4.5.2 Subgradient method no-wait formulation

The formulation presented in the previous section allows each operation point to be used as a portal. However, if large densely utilised stations are used as portals, the number of complicating constraints, i.e., involving variables from different zones, is huge. Consequently, a huge number of dual variables $\lambda_{\cdot}$ is needed, which results in a large number of operations to update the coefficients of the local rescheduling problems $\phi^{\cdot}$ and to compute the new dual estimates after each iteration. In addition, looking more closely at the reference SBB network one discovers that a number of operation points represent junctions and switch regions where it is not sensible to stop. Moreover, many of these operation points are situated right outside main station areas (see, e.g., operation points BGN, BGOS, BGS, and BGWE in the case study in Section 5.8). This observation motivates the formulation proposed in this section.

Definition 4.5 *A* no-wait portal *is an operation point between zones where*

- *no train is planned to stop and*

- *the tracks are so short that if a conflict between two train movements occur, the blocking time intervals of at least one commonly used line resource adjacent to the track overlap.*

Definition 4.6 *A* valid no-wait network partitioning *of a network $\mathcal{N}$ consists of a set $\mathcal{N}_0$, $\mathcal{N}_1$, ..., $\mathcal{N}_m$ of zones of $\mathcal{N}$ that satisfy the requirements (R1-R2) of Definition 4.1, (R3-R4) of Definition 4.4, and*

(R5) the boundaries of the zones are no wait portals, i.e., $\partial\mathcal{N}_i \subset \{s \in \mathcal{S} | s \text{ no-wait portal}\}$, $\forall 0 \leq i \leq m$.

Consider problem (4.62) and let $\mathcal{N}_0, \mathcal{N}_1, \ldots, \mathcal{N}_m$ be a valid no-wait network partitioning. The conflict-freedom constraints (4.10)-(4.11) linked to the portals of the valid no-wait network partitioning can be removed from the definition of the domain X, while the other complicating constraints for all portals $s \in \mathcal{S}_{portals}$ become

$$(4.5) \;\rightarrow\; r_{t,\rightarrow s} = r_{t,s\rightarrow} \quad \forall t \in \mathcal{T} \tag{4.74}$$

$$(4.7) \;\rightarrow\; h_{t,\rightarrow s} = r_{h,s\rightarrow} \quad \forall t \in \mathcal{T} \tag{4.75}$$

$$(4.12) \;\rightarrow\; \sum_{b\in A_{t,p}} x_b = \sum_{b\in D_{t,p}} x_b \quad \forall p \in P_s, t \in \mathcal{T} \tag{4.76}$$

$$(4.13) \;\rightarrow\; \sum_{b\in A_{t,p}} \alpha(b) x_b = \sum_{b\in D_{t,p}} \delta(b) x_b \quad \forall p \in P_s, t \in \mathcal{T} \tag{4.77}$$

$$(4.14) \;\rightarrow\; \sum_{b\in A_{t,p}} v\alpha(b) x_b = \sum_{b\in D_{t,p}} v\delta(b) x_b \quad \forall p \in P_s, t \in \mathcal{T} \tag{4.78}$$

where $\rightarrow s$ indicates the section of a train itinerary with arrival at portal s, and $s \rightarrow$ the section departing from it.

Let $\mathcal{T}^i_{\rightarrow s}$ denote the set of trains in zone $\mathcal{N}_i$ leaving from portal $s \in \partial\mathcal{N}_i$ and $\mathcal{T}^i_{s\rightarrow}$ the ones entering it. In addition, given that each train is supposed to pass only once through each portal, one can uniquely define the variables of one zone linked to the runs from and to a portal s using the index s instead of the run index. The application of a Lagrangian relaxation to the complicating constraints (4.74)-(4.78) of problem (4.61) gives

$$\min_{(x,r,h,c)\in X'} \sum_{i=0}^{m} \Bigg\{ f^i(x^i, r^i, h^i, c^i)$$

$$+ \sum_{j\neq i} \sum_{s\in\mathcal{N}_i\cap\mathcal{N}_j} \left[\sum_{t\in\mathcal{T}^i_{\rightarrow s}\cup\mathcal{T}^i_{s\rightarrow}} \left\{ \lambda^{s,t}_{A,ij} \left[r^i_{t,s} - r^j_{t,s} \right) + \lambda^{s,t}_{C,ij} \left[h^i_{t,s} - h^j_{t,s} \right] \right\} \right.$$

$$+ \sum_{p\in P_s} \Bigg(\sum_{t\in\mathcal{T}^i_{\rightarrow s}} \Bigg\{ \lambda^{s,t,p}_{H,ij} \Bigg[\sum_{b\in A_{t,p}} x^i_b - \sum_{b\in D_{t,p}} x^j_b \Bigg]$$

$$
\begin{aligned}
&\qquad + \lambda^{s,t,p}_{J,ij}\left[\sum_{b\in A_{t,p}} \alpha(b)x^i_b - \sum_{b\in D_{t,p}} \delta(b)x^j_b\right] \\
&\qquad +\lambda^{s,t,p}_{K,ij}\left[\sum_{b\in A_{t,p}} v\alpha(b)x^i_b - \sum_{b\in D_{t,p}} v\delta(b)x^j_b\right]\Bigg\} \qquad (4.79) \\
&+ \sum_{t\in\mathcal{T}^i_{s\to}} \Bigg\{\lambda^{s,t,p}_{H,ij}\left[\sum_{b\in D_{t,p}} x^i_b - \sum_{b\in A_{t,p}} x^j_b\right] \\
&\qquad + \lambda^{s,t,p}_{J,ij}\left[\sum_{b\in D_{t,p}} \delta(b)x^i_b - \sum_{b\in A_{t,p}} \alpha(b)x^j_b\right] \\
&\qquad +\lambda^{s,t,p}_{K,ij}\left[\sum_{b\in D_{t,p}} v\delta(b)x^i_b - \sum_{b\in A_{t,p}} v\alpha(b)x^j_b\right]\Bigg\}\Bigg)\Bigg]\Bigg\} \\
&\lambda^{\cdot}_{\cdot} \geq 0
\end{aligned}
$$

Then, by rearranging the terms and using the sets $X_0, X_1, \ldots, X_m$ as in the previous section one can write the rescheduling problem (4.61) as (4.64) where $\phi^i(\lambda)$ denotes the local rescheduling problem in zone $\mathcal{N}_i$ for a given $\lambda_{\cdot} \geq 0$ and it is defined as

$$
\begin{aligned}
\phi^i(\lambda) = \min_{(x^i,r^i,h^i,c^i)\in X_i} &\Bigg\{ f^i(x^i, r^i, h^i, c^i) \\
&+\sum_{j\neq i} \sum_{s\in\mathcal{N}_i\cap\mathcal{N}_j} \Bigg[\sum_{\substack{t\in\mathcal{T},\\ t\text{passes } s}} \left[r^i_{t,s}\left(\lambda^{s,t}_{A,ij} - \lambda^{s,t}_{A,ji}\right) + h^i_{t,s}\left(\lambda^{s,t}_{C,ij} - \lambda^{s,t}_{C,ji}\right)\right] \\
&+\sum_{p\in P_s}\Bigg(\sum_{t\in\mathcal{T}^i_{\to s}} \sum_{b\in A_{t,p}} x^i_b \Big[\lambda^{s,t,p}_{H,ij} - \lambda^{s,t,p}_{H,ji} \\
&\qquad + \alpha(b)\left(\lambda^{s,t,p}_{J,ij} - \lambda^{s,t,p}_{J,ji}\right) \qquad (4.80) \\
&\qquad +v\alpha(b)\left(\lambda^{s,t,p}_{K,ij} - \lambda^{s,t,p}_{K,ji}\right)\Big] \\
&+ \sum_{t\in\mathcal{T}^i_{s\to}} \sum_{b\in D_{t,p}} x^i_b \Big[\lambda^{s,t,p}_{H,ij} - \lambda^{s,t,p}_{H,ji} \\
&\qquad + \delta(b)\left(\lambda^{s,t,p}_{J,ij} - \lambda^{s,t,p}_{J,ji}\right) \\
&\qquad +v\delta(b)\left(\lambda^{s,t,p}_{K,ij} - \lambda^{s,t,p}_{K,ji}\right)\Big]\Bigg)\Bigg]\Bigg\}
\end{aligned}
$$

As mentioned in the previous section Lagrangian dual problems such as (4.63) are usually solved using the subgradient method in which the local problems (4.80) are iteratively solved with given λ estimates that are updated at each iteration according to (4.66). In this case, for each pair of adjacent zones ($\mathcal{N}_i$, $\mathcal{N}_j$) and each operation point $s \in \mathcal{N}_i \cap \mathcal{N}_j$ the updates can be written as

$$\lambda^{s,t}_{A,ij}(k+1) = \lambda^{s,t}_{A,ij}(k) + w \cdot \left(r^i_{t,s}(k) - r^j_{t,s}(k)\right) \quad \forall t \in \mathcal{T}, \text{ passes } s \tag{4.81}$$

$$\lambda^{s,t}_{C,ij}(k+1) = \lambda^{s,t}_{C,ij}(k) + w \cdot \left(h^i_{t,s}(k) - h^j_{t,s}(k)\right) \quad \forall t \in \mathcal{T}, \text{ passes } s \tag{4.82}$$

$$\begin{aligned} \lambda^{s,t,p}_{H,ij}(k+1) &= \lambda^{s,t,p}_{H,ij}(k) + w \cdot \left(\sum_{b \in A_{t,p}} x^i_b(k) - \sum_{b \in D_{t,p}} x^j_b(k) \right) \\ &= \lambda^{s,t,p}_{H,ij}(k) + w \cdot \left(\mathbb{1}_{\{s\alpha^j_{t,s}(k)=p\}} - \mathbb{1}_{\{s\delta^j_{t,s}(k)=p\}} \right) \quad \forall p \in P_s, t \in \mathcal{T}^i_{\to s} \end{aligned} \tag{4.83}$$

$$\begin{aligned} \lambda^{s,t,p}_{J,ij}(k+1) &= \lambda^{s,t,p}_{J,ij}(k) + w \cdot \left(\sum_{b \in A_{t,p}} \alpha(b) x^i_b(k) - \sum_{b \in D_{t,p}} \delta(b) x^j_b(k) \right) \\ &= \lambda^{s,t,p}_{J,ij}(k) + w \cdot \left(\alpha^i_{t,s,p}(k) - \delta^j_{t,s,p}(k) \right) \quad \forall p \in P_s, t \in \mathcal{T}^i_{\to s} \end{aligned} \tag{4.84}$$

$$\begin{aligned} \lambda^{s,t,p}_{K,ij}(k+1) &= \lambda^{s,t,p}_{K,ij}(k) + w \cdot \left(\sum_{b \in A_{t,p}} v\alpha(b) x^i_b(k) - \sum_{b \in D_{t,p}} v\delta(b) x^j_b(k) \right) \\ &= \lambda^{s,t,p}_{K,ij}(k) + w \cdot \left(v\alpha^i_{t,s,p}(k) - v\delta^j_{t,s,p}(k) \right) \quad \forall p \in P_s, t \in \mathcal{T}^i_{\to s} \end{aligned} \tag{4.85}$$

$$\begin{aligned} \lambda^{s,t,p}_{H,ij}(k+1) &= \lambda^{s,t,p}_{H,ij}(k) + w \cdot \left(\sum_{b \in D_{t,p}} x^i_b(k) - \sum_{b \in A_{t,p}} x^j_b(k) \right) \\ &= \lambda^{s,t,p}_{H,ij}(k) + w \cdot \left(\mathbb{1}_{\{s\delta^i_{t,s}(k)=p\}} - \mathbb{1}_{\{s\alpha^j_{t,s}(k)=p\}} \right) \quad \forall p \in P_s, t \in \mathcal{T}^i_{s\to} \end{aligned} \tag{4.86}$$

$$\begin{aligned} \lambda^{s,t,p}_{J,ij}(k+1) &= \lambda^{s,t,p}_{J,ij}(k) + w \cdot \left(\sum_{b \in D_{t,p}} \delta(b) x^i_b(k) - \sum_{b \in A_{t,p}} \alpha(b) x^j_b(k) \right) \\ &= \lambda^{s,t,p}_{J,ij}(k) + w \cdot \left(\delta^i_{t,s,p}(k) - \alpha^j_{t,s,p}(k) \right) \quad \forall p \in P_s, t \in \mathcal{T}^i_{s\to} \end{aligned} \tag{4.87}$$

$$\begin{aligned} \lambda^{s,t,p}_{K,ij}(k+1) &= \lambda^{s,t,p}_{K,ij}(k) + w \cdot \left(\sum_{b \in D_{t,p}} v\delta(b) x^i_b(k) - \sum_{b \in A_{t,p}} v\alpha(b) x^j_b(k) \right) \\ &= \lambda^{s,t,p}_{K,ij}(k) + w \cdot \left(v\delta^i_{t,s,p}(k) - v\alpha^j_{t,s,p}(k) \right) \quad \forall p \in P_s, t \in \mathcal{T}^i_{s\to} \end{aligned} \tag{4.88}$$

Then, with a constant step size w and an initial estimates of the duals $\lambda(0) = \mathbb{0}$ each local rescheduling problem can be formulated as:

$$\phi^i(\lambda(K+1)) = \min_{(x^i, r^i, h^i, c^i) \in X_i} \left\{ f^i(x^i, r^i, h^i, c^i) \right. \tag{4.89}$$

$$+ \sum_{j \neq i} \sum_{s \in \mathcal{N}_i \cap \mathcal{N}_j} \left[\sum_{t \in \mathcal{T}^i_{\to s} \cup \mathcal{T}^i_{s \to s}} \left\{ \tilde{\lambda}^{i,s}_{t,r}(K) r^i_{t,s} + \tilde{\lambda}^{i,s}_{t,h}(K) h^i_{t,s} \right\} \right.$$

$$+ \sum_{p \in P_s} \left(\sum_{t \in \mathcal{T}^i_{\to s}} \sum_{b \in A_{t,p}} \tilde{\lambda}^{ij}_{s,t,p}(b, K) x^i_b \right.$$

$$\left. + \sum_{t \in \mathcal{T}^i_{s\to}} \sum_{b \in D_{t,p}} \tilde{\lambda}^{ij}_{s,t,p}(b, K) x^i_b \right) \Bigg] \Bigg\}$$

$$\tilde{\lambda}^{i,s}_{t,r}(K) = 2w \sum_{k=0}^{K} \left[r^i_{s,t}(k) - r^j_{s,t}(k) \right] \tag{4.90}$$

$$\tilde{\lambda}^{i,s}_{t,h}(K) = 2w \sum_{k=0}^{K} \left[h^i_{s,t}(k) - h^j_{s,t}(k) \right] \tag{4.91}$$

$$\tilde{\lambda}^{i,\to s}_{t,p}(b, K) = 2w \sum_{k=0}^{K} \Big[\mathbb{1}_{\{s\alpha^i_{s,t}(k)=p\}} - \mathbb{1}_{\{s\delta^j_{s,t,p}(k)=p\}} \tag{4.92}$$

$$+\alpha(b) \left(\alpha^i_{s,t,p}(k) - \delta^j_{s,t,p}(k) \right) + v\alpha(b) \left(v\alpha^i_{s,t,p}(k) - v\delta^j_{s,t,p}(k) \right) \Big]$$

$$\tilde{\lambda}^{i,s\to}_{t,p}(b, K) = 2w \sum_{k=1}^{K} \Big[\mathbb{1}_{\{s\delta^i_{s,t}(k)=p\}} - \mathbb{1}_{\{s\alpha^j_{s,t}(k)=p} \tag{4.93}$$

$$+\delta(b) \left(\delta^i_{s,t,p}(k) - \alpha^i_{s,t,p}(k) \right) + v\delta(b) \left(v\delta^i_{s,t,p}(k) - v\alpha^j_{s,t,p}(k) \right) \Big]$$

This can be interpreted as follows: in each iteration, the local rescheduling problem ϕ^i receives suggestions based on the results of the previous iteration on how to change the values of the variables at the portals to get a consistent global schedule. The signs of the differences indicate whether the value should grow or fall and their absolute values scaled according to $2w$ suggest the entity of the change needed.

4.5.3 Negotiation based no-wait formulation

The approximate duals $\tilde{\lambda}^{\cdot}_{\cdot}$ (4.90-4.93) are computed using the local schedule information at the portals from all previous iterations. This section presents an alternative to the subgradient method no-wait formulation that uses only the local schedules from the last iteration.

As before, at each iteration, the local rescheduling problems ϕ^i (4.89) are solved with given approximate duals. The difference from the subgradient no-wait formulation is the computation of the approximate duals. For each portal an "intermediate schedule" is computed by combining the current local schedules with some weights that reflect the relative importance of exchanged information. Then, the values $\tilde{\lambda}^{\cdot}_{\cdot}$ are defined to quantify the distances of the decision alternatives from the intermediate schedule. The relative importance of exchanged information can depend on:

(a) direction: The most natural way to coordinate the movement of trains along different areas is by considering their travel directions. Xie et al. (2014) suggest that communicating outflows to neighbouring intersections within a rolling horizon provides an implicit coordination. Direction based precedences are the base of the coordination by Corman et al. (2012b). The strength of this strategy is that decisions that are to be applied in a nearer future get higher priority than the following ones. A drawback of this strategy is that trains might be forced into an already congested area even if they could stay a little longer in a previous uncongested area without creating any inconvenience.

(b) hierarchical: Analogous to the microscopic timetabling work-flow with condensation and compensation areas proposed by Caimi (2009). The local schedules of condensation zones are imposed to the adjacent compensation zones. Of course this strategy can be applied to different hierarchies than condensation zones before compensation zones (e.g., model based Kersbergen et al., 2016). A drawback of this approach is that the rescheduling problem in a low-rank area can result infeasible due to the boundary conditions imposed by higher-ranked neighbours.

(c) negotiation: A combination of the previous two strategies can be achieved with a negotiation between the areas. Adjacent zones negotiate the passing track, time and speed of each train at the common portals aiming at having a common result. The weight of each local schedule for computing the intermediate schedule can depend on the travel direction of the train or/and the type of area (condensation or compensation). Note that this strategy also allows the definition of overlapping areas.

Strategies a) and b) are special cases of strategy c), in which the weight of the schedule information of either one direction or area is one and the other zero. Table 4.2 exemplifies the three strategies for the situation depicted in Figure 4.13 and assuming zone $\mathcal{N}_i$ is a condensation zone and zone $\mathcal{N}_j$ a compensation zone. Strategy c) also enables the assignment of different weights to each of the components track, time and speed.

Table 4.2: Weights of exchanged information for rescheduling in adjacent zones: Examples for Figure 4.13.

	$\mathcal{N}_i$		$\mathcal{N}_j$	
Travel direction / Approach	inbound	outwards	inbound	outwards
(a) direction	0	1	0	1
(b) hierarchical	1	1	0	0
(c) negotiation	[0, 1]	[0, 1]	[0, 1]	[0, 1]

For given negotiation weights, the approximate duals $\tilde{\lambda}_{\cdot}^{\cdot}$ can be updated as follows:

$$\tilde{\lambda}^{i,s}_{t,r}(k+1) = \varphi(t)\left[\mathbb{1}_{\{w^{i,r}_{t,s}\geq w^{j,r}_{t,s}\wedge r^{i}_{s,t}(k)=0\}} + \mathbb{1}_{\{w^{i,r}_{t,s}<w^{j,r}_{t,s}\wedge r^{j}_{s,t}(k)=0\}}\right] \tag{4.94}$$

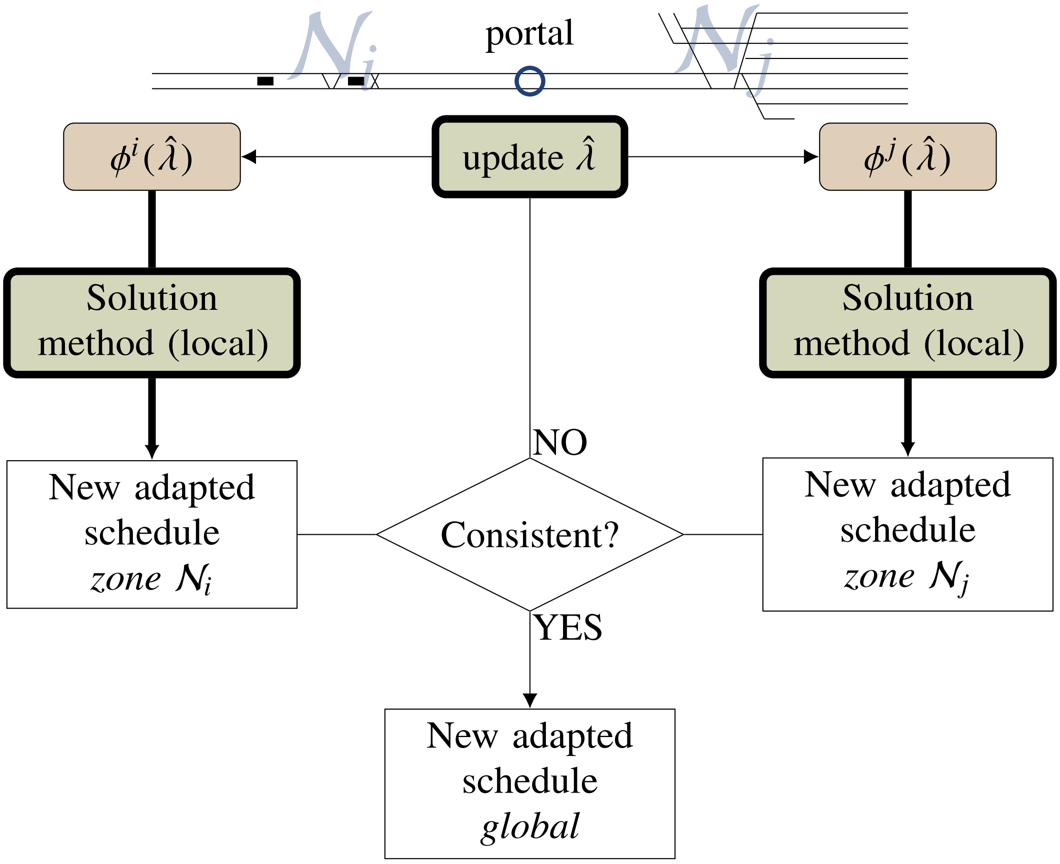
$$\tilde{\lambda}^{i,s}_{t,h}(k+1) = \varphi(t)\left[\mathbb{1}_{\{w^{i,h}_{t,s}\geq w^{j,h}_{t,s}\wedge h^{i}_{s,t}(k)=0\}} + \mathbb{1}_{\{w^{i,h}_{t,s}<w^{j,h}_{t,s}\wedge h^{j}_{s,t}(k)=0\}}\right] \tag{4.95}$$

$$\begin{aligned}\tilde{\lambda}^{i,s}_{t,p}(b,k+1) = \varphi(t)\Big[&\mathbb{1}_{\{w^{i,track}_{t,s}\geq w^{j,track}_{t,s}\wedge s\alpha^{i}_{s,t}(k)\neq p\}} + \mathbb{1}_{\{w^{i,track}_{t,s}<w^{j,track}_{t,s}\wedge s\delta^{j}_{s,t,p}(k)\neq p\}} \\ &+ \left|\alpha(b) - \left(w^{i,time}_{t,s}\alpha^{i}_{s,t}(k) + w^{j,time}_{t,s}\delta^{j}_{s,t}(k)\right)\right| \\ &+ \left|v\alpha(b) - \left(w^{i,speed}_{t,s}v\alpha^{i}_{s,t}(k) + w^{j,speed}_{t,s}v\delta^{j}_{s,t}(k)\right)\right|\Big]\end{aligned} \tag{4.96}$$

$$\begin{aligned}\tilde{\lambda}^{i,s}_{t,p}(b,k+1) = \varphi(t)\Big[&\mathbb{1}_{\{w^{i,track}_{t,s}\geq w^{j,track}_{t,s}\wedge s\delta^{i}_{s,t}(k)\neq p\}} + \mathbb{1}_{\{w^{i,track}_{t,s}<w^{j,track}_{t,s}\wedge s\alpha^{j}_{s,t,p}(k)\neq p\}} \\ &+ \left|\delta(b) - \left(w^{i,time}_{t,s}\delta^{i}_{s,t}(k) + w^{j,time}_{t,s}\alpha^{i}_{s,t}(k)\right)\right| \\ &\left|v\delta(b) - \left(w^{i,speed}_{t,s}v\delta^{i}_{s,t}(k) + w^{j,speed}_{t,s}v\alpha^{j}_{s,t}(k)\right)\right|\Big]\end{aligned} \tag{4.97}$$

where $\varphi(t)$ is a function of the available time for decisions about train t; $w^{\cdot,\cdot}_{t,s}$ are the weights of the decisions about train t at portal s in the zones $\mathcal{N}_i$ and $\mathcal{N}_j$ concerning the components time, speed and track, and of decisions on deferring it to the next horizon or cancelling it; $r^{\cdot}_{s,t}(k)$ is the decision taken in the k^{th} iteration about deferring the run of train t on the section adjacent to portal s; $h^{\cdot}_{s,t}(k)$ is the decision selected in the k^{th} iteration about cancelling the run of train t on the section adjacent to portal s; $\alpha^{\cdot}_{s,t}(k)$ and $\delta^{\cdot}_{s,t}(k)$ are the arrival and departure times of train t at the portal s found within the k^{th} iteration; $s\alpha^{\cdot}_{s,t}(k)$ and $s\delta^{\cdot}_{s,t}(k)$ are the arrival and departure tracks; and $v\alpha^{\cdot}_{s,t}(k)$ and $v\delta^{\cdot}_{s,t}(k)$ the arrival and departure speeds. Figure 4.13 shows the the distributed non-hierarchical

Figure 4.13: Coordinated rescheduling of adjacent zones using the negotiation-based no-wait formulation.



framework for railway traffic rescheduling using the negotiation-based no-wait formulation.

As a train approaches a portal, agreeing on a track, a time and a speed becomes more important, i.e., the coordination costs increase with approaching application time of the decision. Thus, $\varphi()$ is a monotonically decreasing function such as

$$\varphi(t) = \frac{M}{\min(\alpha_{t,i}, \alpha_{t,j}) - h_{begin}} \tag{4.98}$$

where h_{begin} is the beginning of the current rescheduling horizon and M is a large constant.

There are often operational rules that fix the direction of travel of the tracks at portals. Consider the examples in Figure 4.14. On single lines there is only one possible track assignment for all movements. In multiple-track sections the maximum capacity is attained when all tracks are used for one direction (and train type). Thus, in many situations it is

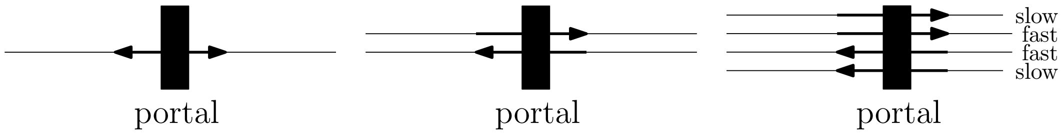


Figure 4.14: Tracks assignments at portals

possible to fix the passing tracks at portals and the equations 4.96-4.97 are simplified as

$$\tilde{\lambda}^{i,s}_{t,p}(b,k+1) = \varphi(t)\Big[\mathbb{1}_{\{w^{i,track}_{t,s} \geq w^{j,track}_{t,s} \wedge s\alpha^{i}_{s,t}(k) \neq p\}} + \mathbb{1}_{\{w^{i,track}_{t,s} < w^{j,track}_{t,s} \wedge s\delta^{j}_{s,t,p}(k) \neq p\}}$$
$$+ \left|\alpha(b) - \left(w^{i,time}_{t,s}\alpha^{i}_{s,t}(k) + w^{j,time}_{t,s}\delta^{j}_{s,t}(k)\right)\right| \tag{4.99}$$
$$+ \left|v\alpha(b) - \left(w^{i,speed}_{t,s}v\alpha^{i}_{s,t}(k) + w^{j,speed}_{t,s}v\delta^{j}_{s,t}(k)\right)\right|\Big]$$

$$\tilde{\lambda}^{i,s}_{t,p}(b,k+1) = \varphi(t)\Big[\mathbb{1}_{\{w^{i,track}_{t,s} \geq w^{j,track}_{t,s} \wedge s\delta^{i}_{s,t}(k) \neq p\}} + \mathbb{1}_{\{w^{i,track}_{t,s} < w^{j,track}_{t,s} \wedge s\alpha^{j}_{s,t,p}(k) \neq p\}}$$
$$+ \left|\delta(b) - \left(w^{i,time}_{t,s}\delta^{i}_{s,t}(k) + w^{j,time}_{t,s}\alpha^{i}_{s,t}(k)\right)\right| \tag{4.100}$$
$$\left|v\delta(b) - \left(w^{i,speed}_{t,s}v\delta^{i}_{s,t}(k) + w^{j,speed}_{t,s}v\alpha^{j}_{s,t}(k)\right)\right|\Big]$$

Differently from the subgradient method, there is no guarantee that this method converges. Thus, the following rule is applied. If no convergence has been reached after a given number of iterations, the values at the portals are fixed to the ones found in the last iteration according to the weights of the components. The number of iteration should be determined according to the available time for computation and the available compute server.

Chapter 5

Computational experiments

This chapter presents a proof-of-concept of the methodology for coordinated railway traffic rescheduling presented in the previous chapter. The models and algorithms have been translated into executables that have been run on a number of railway traffic scenarios to test the validity of the proposed methodology and its limitations. Tests have been made at every step of the development of the methodology (see Table 5.1).

The tests have been executed in simulated environments based on three networks that have been selected basing on the requirements of the different stages of the algorithm development and on data availability. The toy network of the ETH railway operations laboratory (EBL) is used to test the validity of the mathematical representation of the problem, the different rescheduling objectives, and the computational implementation of the algorithms. The small number of stations and tracks enables a full overview of the network, which is ideal to identify implementation errors and blind spots of the model with a relatively low computational effort (see Section 5.2.1 for further details). The line Berne-Worblaufen of the Regionalverkehr Bern-Solothurn commuter network (RBS) is used to investigate the solution quality per computation time performance of the monolith and column generation approaches for homogeneous but extremely dense traffic in a simple topology (see Section 5.2.2 for further details). In fact, in peak hours, this 3.8 km double track section is travelled by 42 trains per hour. The area around Brugg managed by the Swiss Federal Railways (SBB) is used to test the scalability of column generation approaches and the validity of the non-hierarchical coordination approach for rescheduling in adjacent zones from Section 4.5. In Brugg a north-south corridor with very heterogeneous traffic and many freight trains meets an east-west corridor mostly used by long distance

Table 5.1: Overview of numerical experiments.

Section	to prove	scenario	theory in Section
5.3	validity of RCG model extension to consider multiple operation points and energy efficiency objectives	EBL	4.2.1, 4.2.2
5.4	suitability of RCG model to represent passenger traffic objectives	EBL	4.2.2
5.5	validity of multi-objective rescheduling framework	part of SBB	4.2.2
5.6	effectiveness of column generation approaches for a small condensation zone with simple topology	RBS	4.4
5.7	effectiveness of column generation approaches in a large network around a complex condensation zone	SBB	4.4
5.8	validity of coordination strategies	SBB	4.5

and regional passenger trains. The size of the area enables the limits of local rescheduling algorithms to be tested and provides different network partitioning possibilities (see Section 5.2.3 for further details).

Section 5.1 summarises the main features of the railway simulator used for the experiments and the interaction between the simulated railway operations and the rescheduling routines. Section 5.2 presents the three networks in detail. Sections 5.3-5.8 describe the test cases and the computational results. The experiments that have been described in detail in previous publications are summarised and references to the original papers are given; while unpublished results are described in detail. The tests are presented following the chronological order of the algorithm development. First, the experiments aiming at validating the mathematical representation of the railway traffic rescheduling problem are reported. Second, the experiments investigating the performance of the solution methods for local rescheduling are described. Finally, the experiments addressing the validity of the framework to coordinate rescheduling in adjacent zones are presented. To guide the reader through the steps of the algorithm development, the lessons learned from each experiment are highlighted in boxes at the end of each section. Table 5.1 shows an overview of the experiments.

5.1 Test environment

Models and algorithms should provide feasible schedules that consider the current state of the resources (infrastructure and vehicles) and are compatible with signalling and interlocking systems. The selected train routes must have correspondents into the interlocking system. All routing, speed, and departure commands must be displayable by the signalling system and must reach the trains some time before their execution[1]. Given the high risks linked with a failure, it is not possible to validate models and algorithms from this research directly on real world railway traffic. To overcome this issue, there are several simulation tools that are able to replicate railway operations with a high degree of precision, including train dynamics and interlocking.

5.1.1 Setup

Many researchers have developed *ad hoc* tools to validate their own research results (e.g., Fuchsberger, 2012; Rao, 2015). This praxis has the disadvantage that comparisons with other works are difficult. In addition, it requires a considerable effort from the researcher to implement and verify the simulator before being able to exploit it to validate the main elements of the own research. Several academic and commercial railway simulation software applications exist (e.g., OpenTrack, Hürlimann, 2002; EGtrain, Quaglietta, 2011; RailSys, rmcon, 2015), and many of them are currently widely accepted and used by both the research community and practitioners. These software applications replicate the chains of interconnected events taking place during railway operations and usually include modules for stochastic components such as delays or disruptions.

To validate the models and algorithms of the present thesis, a simulator must be able to replicate railway operations on a microscopic track topology with realistic train dynamics, interlocking, signalling, and information exchanges between traffic control, infrastructure, and trains. OpenTrack is a simulator of railway operations that has been validated on a large number of different railway networks and timetables and is also utilised by practitioners. Given a microscopic track topology model, a train catalogue and a timetable, it replicates railway operations including train dynamics, interlocking and signalling with a track circuit precision. A joint project of OpenTrack, IBM, and the University of Applied Sciences of Winterthur ZHAW has developed an Application Programming

[1] This time depends on the signalling system and on whether the train operation is controlled by a human driver or automated.

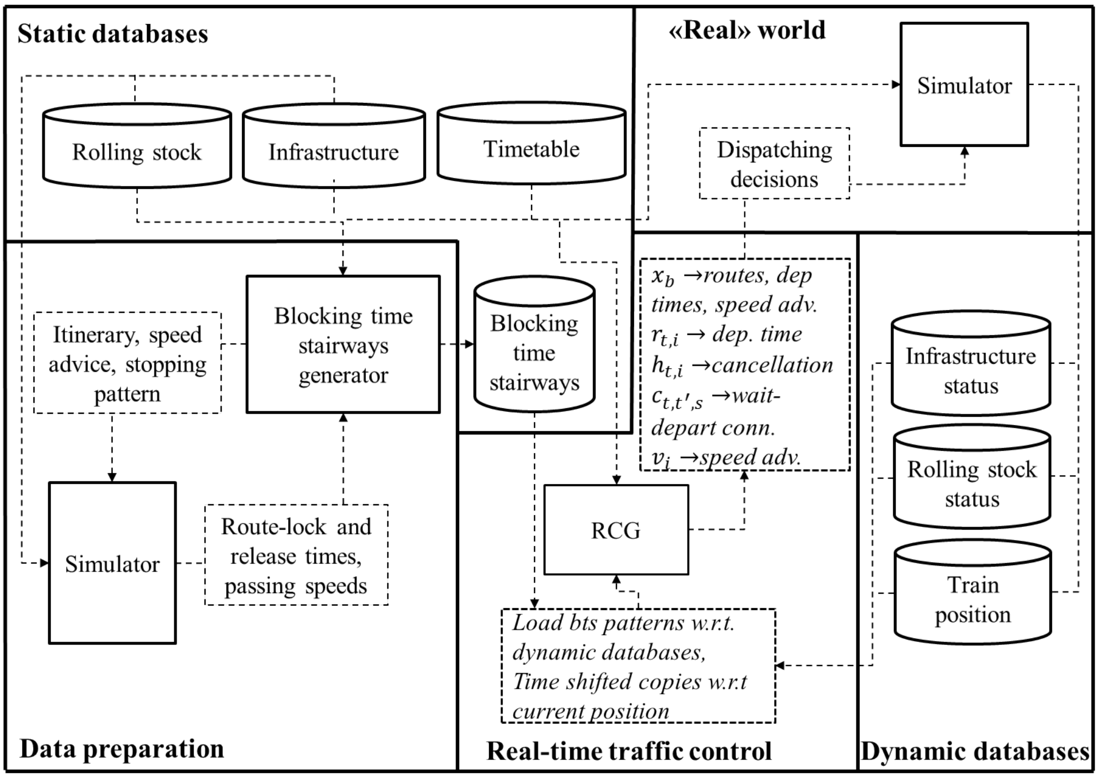


Figure 5.1: Test environment for computational experiments. Variables x_b, $r_{t,i}$, $h_{t,i}$, $c_{t,t',s}$, v_i as defined in previous chapter.

Interface (API) that enables message exchanges between an OpenTrack simulation environment and external programs. The API enables OpenTrack to send notifications of train positions, timetable events (departure, passing, arrival), and interlocking actions (reserve, enter, exit, release routes/sections of routes/infrastructure resource). Conversely, OpenTrack receives dispositions affecting trains (create, cancel, request position report), the timetable (change departure, passing, and arrival times, set or remove stops/connections), and the interlocking system (allow, disallow, reserve routes). These features make OpenTrack the most suitable tool for testing both the simulation-based blocking time stairways generator from Section 4.2.3 and the different rescheduling models and algorithms from Chapter 4. Figure 5.1 shows the test environment.

The blocking time stairway generator is an executable from C++ code that sends to the simulator the sets of trains, paths, and speed parameters basing on the (static) infrastructure, rolling stock, and timetable data. OpenTrack simulates the train runs according to the received information and notifies route-lock and resource-release times to the generator, which builds the blocking time stairways and saves them into a static database[2].

[2]This program has been used to both generate the database of blocking time stairways patterns for the rescheduling algorithms and predict train movements according to the published timetable.

Local and coordinated rescheduling programs are executables implemented in C++. In this case, the role of the simulator is to reproduce railway operations as they could occur in the real world. OpenTrack takes the railway infrastructure, the timetable, and the rolling stock information and simulates the traffic evolution with or without perturbations. During each simulation run, the current traffic situation is reflected into dynamic infrastructure, rolling stock and train positions databases, which can be accessed by the rescheduling program. Train position notifications can have different granularities: one could consider an infrastructure-side positioning in which trains are located when they travel through insulated joints and axle counters, or a train-side positioning in which trains continuously communicate their position to the traffic control centre (e.g., via GSM-R in case of ETCS Level 2). Hereinafter, in order to faithfully reflect the current reality of the considered test scenarios, infrastructure-side train positioning is applied. The rescheduling program is fed with both static and dynamic rolling stock, infrastructure, and timetable data. With these pieces of information it computes the new schedule and sends decisions about routings, timings and connections to OpenTrack, which then continues the simulation of railway operations for validation of the new schedule.

5.1.2 A reminder of notation

Hereinafter, the nomenclature and abbreviations defined in the previous chapter are used:

monolith denotes the RCG model from Section 4.2 built with all precomputed blocking time stairway patterns in a given database and all time-shifts fitting into the considered rescheduling horizon, the time-shift interval is given. This model is solved by CPLEX (IBM, 2015).

cgApP denotes the column generation framework for the RCG model with arc pricing problems using precomputed blocking time stairway patterns described in Section 4.4.4.1.

cgApC denotes the column generation framework for the RCG model with arc pricing problems constructing new alternative blocking time stairways described in Section 4.4.4.3.

cgPpP denotes the column generation framework for the RCG model with path pricing problems using precomputed blocking time stairways described in Section 4.4.4.2.

cgPpC denotes the column generation framework for the RCG model with path pricing problems constructing new blocking time stairways described in Section 4.4.4.4.

5.2 Test scenarios

This section presents the three test networks that have been selected for the experiments (cfr. plan in Table 5.1). Some easily controllable scenarios were needed to test the mathematical model and the first implementations of the algorithms; while real-world scenarios with both simple and complex topologies were needed to validate the algorithms. To satisfy these requirements three networks have been chosen according to the data availability. The small toy network of the railway operations lab of the Institute for Transport Planning and Systems (IVT) at ETH Zurich is described in Section 5.2.1. A double-track section with extremely dense homogeneous passenger traffic managed by RBS is described in Section 5.2.2. A complex node with highly heterogeneous traffic managed by SBB is described in Section 5.2.3.

5.2.1 Railway operations laboratory (EBL)

The first tests of the simulation-based blocking time stairway generator and of the RCG rescheduling model are run using the infrastructure of the railway operations laboratory of IVT shown in Figure 5.2. The network is equipped with signals, track circuits, and axle counters that replicate precisely the interlocking and line-side signalling systems currently used in Switzerland. Rolling stock reflects the vehicles currently circulating on the SBB network. Several timetables with different traffic density and heterogenity are developed for testing the first implementations of the blocking time stairways generator (manual configuration and data export from OpenTrack) and rescheduler (Java executable and manual import of result in OpenTrack), see, e.g., Toletti et al. (2015a, 2016).

5.2.2 Berne-Worblaufen (RBS)

The double track line between Berne and Worblaufen managed by RBS is used as an example of an extremely congested condensation zone with simple topology. RBS is a local public transport company active in the Swiss cantons of Berne and Solothurn. RBS manages four commuter train lines and several local bus lines. Only the rail services are considered

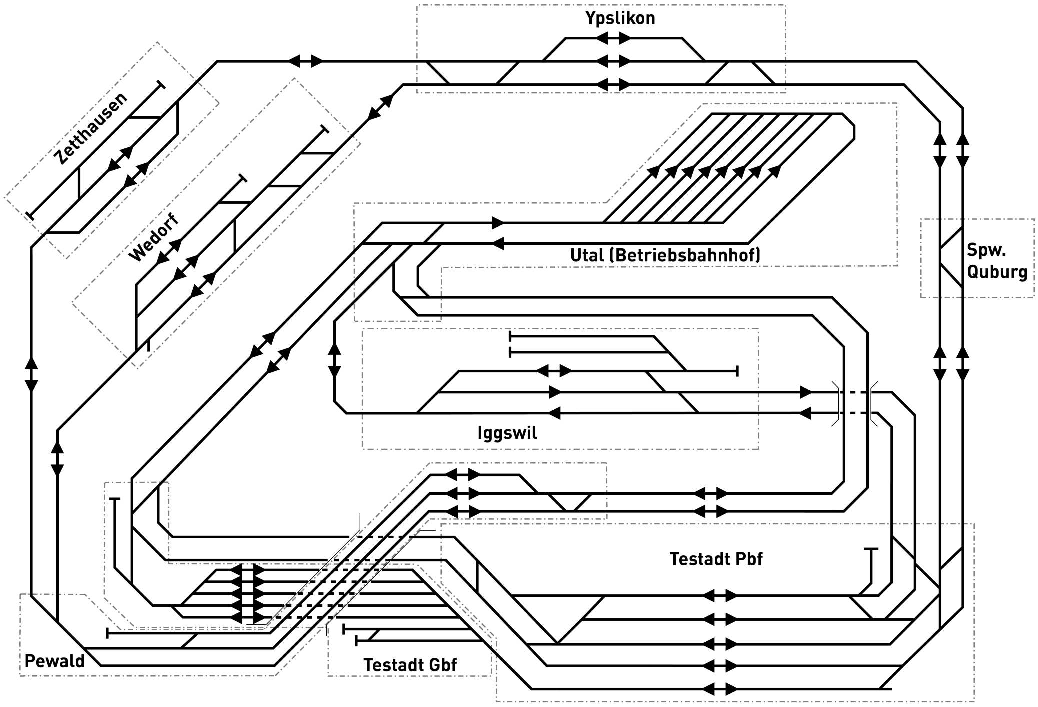


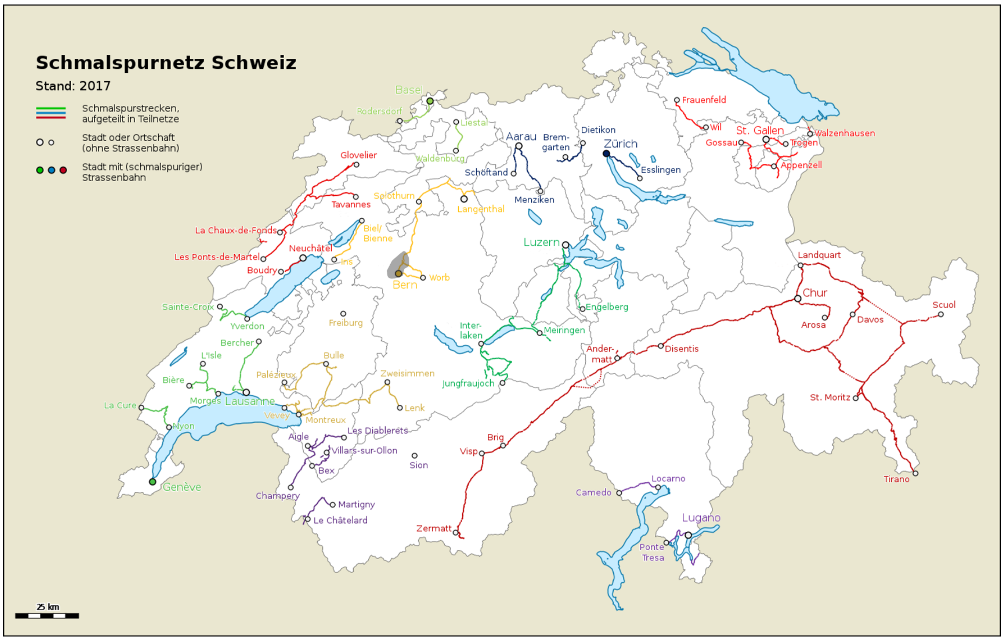
Figure 5.2: EBL test network (IVT archive)

by the following experiments. The railway network consists of a double-track section from main station Berne (BN) to junction station Worblaufen (WBL) and three (partially single-track) branches from Worblaufen towards Worb, Unterzollikofen, and Jegenstorf-Solothurn (Figure 5.3). The network is equipped with line-side signalling, and track circuits are used to position trains and release infrastructure resources.

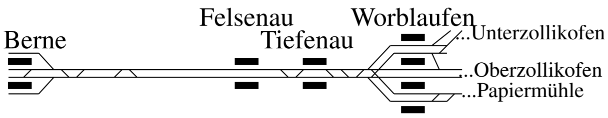
The entire network topology at track circuit precision including maximum speed indications had been implemented in OpenTrack by PSI Software AG and made available by RBS for the current study. RBS has provided the timetable in graphical form, while rolling stock has been assigned to train runs according to the information given by RBS on its website.

During the peak hour 7:00-7:59 a.m., the 3.8-km long section Berne-Worblaufen is travelled by a total of 42 trains, which results in an average of 90 seconds between train movements at Berne station. Therefore, there is not shunting possibility at terminus station Berne, and all services must depart form the same track where the previous service ended. Not only the line is at its capacity limit but also the Berne station area which was built for about 16’000 passengers per day and is currently used by about 60’000 passengers per day. Thus, doors to manage pedestrian flows at

Figure 5.3: RBS test network



(a) Global view of Swiss narrow-gauge railways: test environment in grey (adapted from Plutowiki, 2017).



(b) Local view of selected tracks .

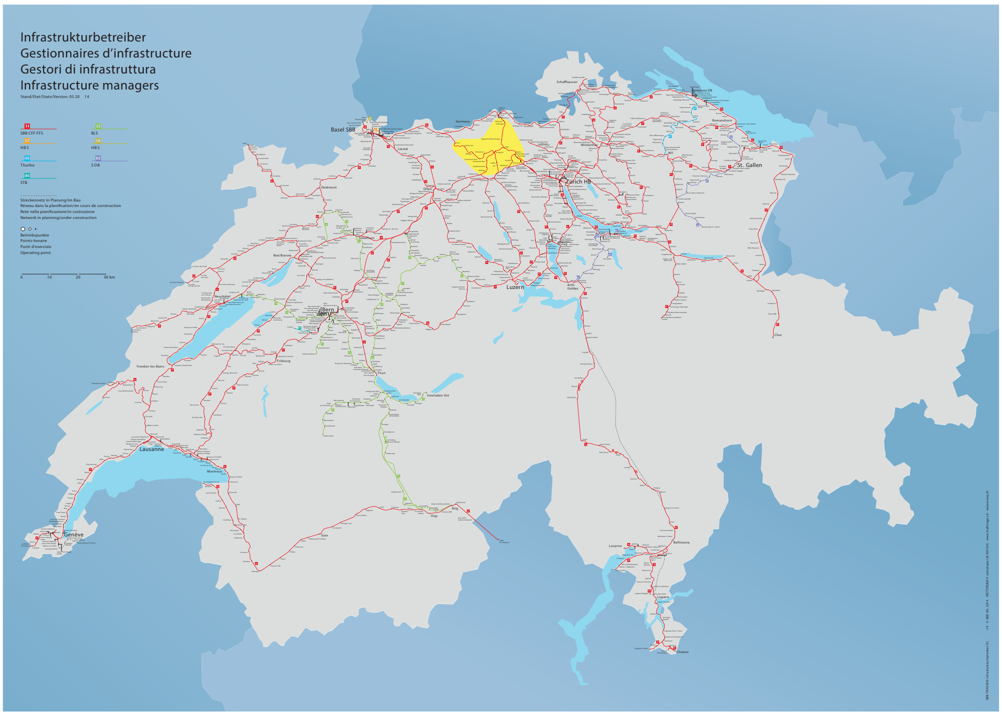
the station had to be installed, which imposes longer turn around times to allow passengers to alight and board.

5.2.3 Brugg (SBB)

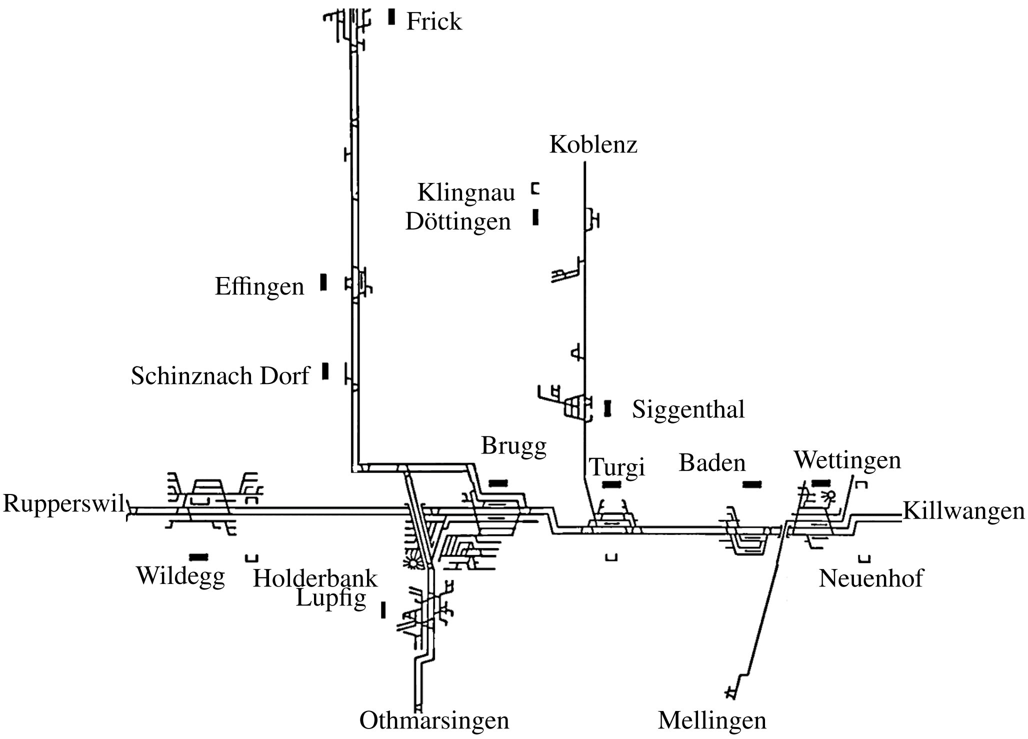
The region around Brugg, Switzerland, is used as large real world test environment. Brugg is a middle size station situated at the meeting point of the east-west corridor connecting Berne (the political capital) to Zurich (the main economic pole) and the north-south corridor connecting Germany to Italy. Traffic is mixed on both corridors and there are some trains travelling from east to north and south and the other ways round. Passenger traffic is dominant on the east-west corridor, and there are several smaller stations used by many passengers, as they are situated near to residential areas. Freight traffic is dominant on the north-south axis, which mostly consists of a double track line with some minor stations serving very few passengers and a few switches where trains can change track. Figure 5.4 shows the considered area.

The portals (i.e., the borders of the area) are defined following sugges-

Figure 5.4: SBB test network



(a) Global view of Swiss railways: test environment in yellow (adapted from SBB AG, 2014a).



(b) Local view of selected tracks.

tions from practitioners. The Killwangen junction is currently managed by an automatic train reordering and speed advisory system (RCS-HOT, Oettich and Caspar, 2015). Thus, assuming that the system can manage any traffic configuration leaving the test area, it is reasonable to set a portal just before the junction. Due to the similar track topology, it is assumed that the same system will be applied at the Grüemet junction (right outside station Mellingen), in Othmarsingen, Rupperswil and Koblenz. The last portal, in Frick, is suggested from to the fact that the suburban lines of city Basel end there. Thus, the portal enables a clean separation with Basel suburban traffic.

A mix of open and SBB proprietary data has been used to build the case studies. The (microscopic) track topology, signalling and interlocking descriptions have been made available by SBB. Currently (2018), this area is equipped with a fixed block safety system that detects trains when they pass insulated joints and axle counters with line-side signalling. Appendices B.1-2 show the list of documents used. Gradients and allowed speed profiles for all braking classes have been read from the collection of Swiss standards R I-30131 (2016). The official timetables are openly accessible on the internet (Official Timetable, 2017). Appendix B.3 shows the files used. These timetables include scheduled passenger and freight trains as well as scheduled operational runs such as empty trains and locomotives travelling to the starting point of their next commercial run. Train numbers have been mapped to a train type according to Swiss standards R I-30111 complement (2015). The rolling stock of long distance passenger traffic has been assigned according to Reisezüge.ch (2005-2011), a private non-commercial website listing the vehicles used by long distance passenger trains through Switzerland. All passenger trains of each line have been assumed to use the same type of rolling stock.

Furthermore, SBB has provided one week (20-27.03.2015) of railway operations records at station Brugg. These data have been used to identify rolling stock compositions of freight trains and to estimate the probability distributions of the disturbances occurring in this region. Appendix C highlights the main findings.

5.3 RCG model extension and energy efficiency objectives in a small compensation zone with simple topology

This section analyses the test run on an artificial mixed-traffic scenario based on the infrastructure of EBL and published by Toletti et al. (2016).

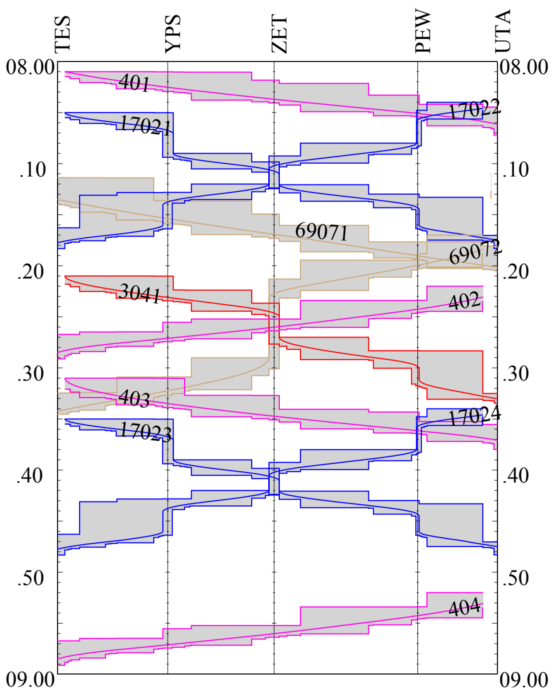


Figure 5.5: Mixed traffic scenario on EBL infrastructure (adapted from Toletti et al., 2016).

The goal is to attest the validity of the RCG model extension to consider multiple operation points (Section 4.2.1) and its suitability to represent energy related targets (Section 4.2.2). In addition, an analysis of the methods used in that first paper is performed to explain the methodological decisions made for the current thesis.

Toletti et al. (2016) define a compensation zone including EBL operation points Testadt Pbf (TES), Ypslikon (YPS), Zetthausen (ZET), Pewald (PEW) and Utal (UTA). The timetable is shown in Figure 5.5 and includes four intercity trains (magenta), a regional train (red), four suburban trains (blue), and two freight trains (brown). One of the freight trains makes a non-commercial stop half-way to allow crossing and overtaking. The delay scenario consists in one freight train entering the compensation zone with a delay of three minutes and causing follow-up delays and unplanned stops.

A preliminary version of the rescheduling program has been built in Java according to the specifications given in Section 4.2.1. The prgram exploits the algorithms developed by Caimi (2009) and Fuchsberger (2012) for finding the maximal cliques of the conflict graphs and the maximal incompatible sets for time-consistency at operation points. The objective of railway traffic rescheduling is the minimisation of a linear combination of train delays and energy consumption. The blocking time stairways linked to the decision variables are time-shifted and time-extended copies of patterns generated with OpenTrack software. These patterns are built from the standard time-space-speed-power consumption output file of OpenTrack (i.e., without using the API). Each train is simulated separately with no speed restriction nor delay according to the green wave principle. Then, the time-space-speed-power consumption output file is cut at the operation points along the train itinerary, and the corresponding blocking time stairways are built by mapping the train position to the positions of signals and track circuits along the itinerary. Time supplements for route reservation and release are set according to the literature (Hansen and Pachl, 2014). An estimation of the energy consumption is obtained from the standard time-space-speed-power consumption file and attached to each blocking time stairway, which is then stored into a database. One additional energy efficient pattern is generated with the simulation-based framework by De Martinis et al. (2014). Different speed profiles are approximated by progressively extending the reservation and release times of the patterns and assuming a quadratic reduction of the energy consumed.

Four different instances (with/without the energy efficient trajectory, with/without running time extensions) are created and solved by IBM ILOG CPLEX Optimization Studio version 12.6 on a 64-bit laptop running Windows 7 Enterprise and equipped with a 2.90 GHz processor (Intel(R) Core(TM).7-3520M CPU @ 2.9GHz) and 8 GB RAM. All four new schedules are feasible and have lower objective values than the non-optimised case. The computation times for building and solving each RCG model range from 14.1 to 104.9 seconds. Thus, it can be concluded that the RCG model described in Section 4.2.1 is valid for rescheduling in compensation zones. Moreover, this experiment demonstrates that, given the energy consumption associated with each blocking time stairway in a database, the overall energy consumption of a schedule can be easily estimated (and minimised) by the RCG approach.

The study also highlights the high sensitivity of the solution to the weights assigned to the delays and energy consumptions of the different trains in the objective function (see, Toletti et al., 2016). However, the definition of the weights of the different objectives is out of the scope of the current work and, thus, it is not further investigated.

Lessons learned:

1. *The RCG model extension in Section 4.2.1 is valid for rescheduling in compensation zones;*

2. *The overall energy requirement of a new schedule can be estimated (and minimised) by the RCG model;*

3. *The computation times for a small compensation zone are satisfactory but suggest the need for improved implementation of the algorithms for larger problems;*

4. *The railway simulation software OpenTrack provides all pieces of information required for building blocking time stairway patterns to be used by the rescheduling algorithms, which motivates the implementation of the simulation-based approach using the API proposed in Section 4.2.3.*

5.4 Passenger-oriented objectives in a small condensation zone with simple topology

This section summarises an experiment performed on an artificial mixed-traffic scenario with a dense timetable based on the infrastructure of EBL and published by Toletti and Weidmann (2016). To test the flexibility of the RCG model extension to represent passenger-oriented objectives and its computation times in condensation zones, additional trains and some passenger connections are added to the timetable of the previous experiment. Additionally, the single-track from station Pewald (PEW) to Wedorf (WED) is considered. Figure 5.6 shows the time-space diagram of the schedule that includes two long-distance trains (magenta), two regional services (red), eight trains of suburban passenger lines (blue), and seven freight trains (brown). Again, one of the freight trains has a scheduled stop to enable crossing and overtaking. Connections are provided at Pewald, where the main line is joined by the branch from Wedorf. The initial delay is caused by a passenger train which departs from a scheduled stop two minutes later than planned.

The RCG model is built in C++ under a Linux enviromnet according to the specifications given in Section 4.2.1. Again, the algorithms developed by Caimi (2009) and Fuchsberger (2012) are exploited for finding the maximal cliques of the conflict graphs and the maximal incompatible sets for time-consistency at operation points. The blocking time

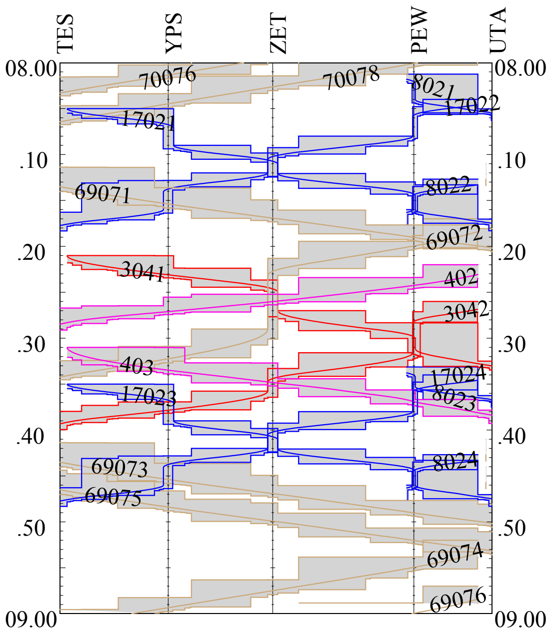


Figure 5.6: Mixed traffic scenario with four passenger connections on EBL infrastructure (adapted from Toletti and Weidmann, 2016).

stairways used as decision variables are time-shifted copies of patterns generated by a C++ implementation of the simulation-based approach presented in Section 4.2.3. The generator interacts with railway simulation software OpenTrack version 1.8.3 running on Windows through the API. OpenTrack is configured to receive routing information and speed targets, simulate each train separately, and send back route-locks, resource-releases, departure and arrival times and passing speeds at operation points. Then, these pieces of information are used to define the blocking time stairway patterns that are saved into a database.

Three different objectives are tested: (1) the minimisation of train delays (cancellations and missed connections are converted to delays according to the timetable period); (2) the minimisation of passenger delays (computed as weighted train delays according to the number of passengers using each run or connection); and (3) the minimisation of passenger inconvenience (also including weighted waiting times at stations, see Sato et al., 2013). The RCG models are solved by IBM ILOG CPLEX Optimization Studio version 12.6.3 on a 64-bit desktop PC running Linux Redhat enterprise 6 and equipped with four processors (Intel(R) Core(TM) i5-2400 CPU @ 3.1GHz) and 8 GB RAM. All resulting schedules are feasible and different from each other. Moreover, the schedules have lower objective values than the non-optimised case. Again, the results are very sensitive to the weights assigned in the objective functions.

Lessons learned:

1. *The RCG model extension in Section 4.2.1 is valid for rescheduling in condensation zones;*

2. *Different passenger-related targets can be represented by the RCG model;*

3. *The simulation-based blocking time stairway patterns generation approach proposed in Section 4.2.3 provides valid blocking time stairway patterns to be used by the RCG model.*

5.5 Multi-objective rescheduling framework[3]

Figure 5.7 shows a framework for multi-objective railway traffic rescheduling using the objectives and the ϵ-constraints method described in Section 4.2.2. The basic idea is to use the adaptive ϵ-constraints method by Laumanns et al. (2006) to investigate the solution space during a data preparation phase. Then, some promising partitions of the solution space are identified and used to define ϵ-constraints that are added to different copies of the RCG model from Section 4.2.2 that are solved concurrently to produce alternative optimal solutions with respect to the given objectives for real-time rescheduling.

The proposed multi-objective rescheduling framework is validated through numerical experiments using real infrastructure and timetable data about the eastern branch of SBB test network. The area consists of a main corridor and a side corridor (see Figure 5.10). Several fast passenger trains travel on the main corridor with a periodic timetable; regional trains circulate on both the main and the minor corridors according to a fully periodic timetable; freight traffic is present on both corridors and, although it is not properly periodic, the slots are allocated almost periodically due to the periodicity of passenger traffic which is dominant in the area. For this numerical study, the number of passengers alighting at each stop is fixed to 20 and no passenger transfer is considered.

The experiment has two phases:

[3]This section is adapted from Toletti et al. (2017)

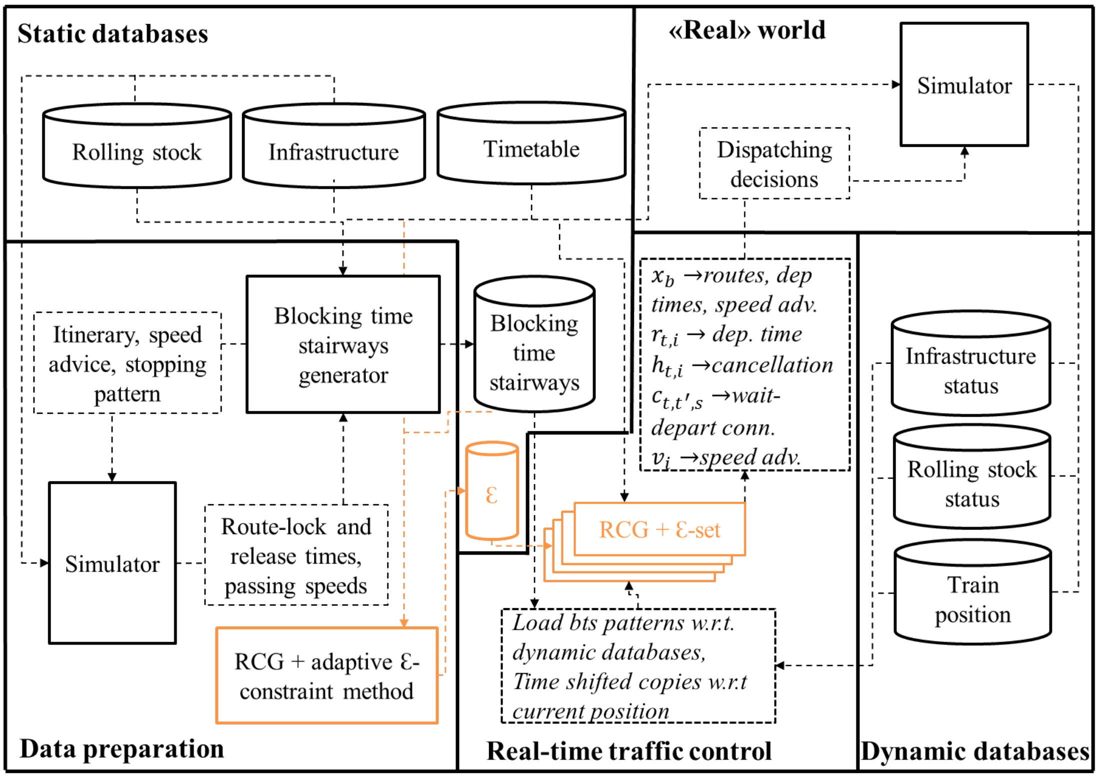


Figure 5.7: Framework for multi-objective railway traffic rescheduling (adaptation of Figure 5.1).

1. **Data preparation phase:** A multi-criteria RCG model is built and solved using an adaptive ϵ-constraint method. The result is the Pareto front of the multi-objective rescheduling problem for given train trajectory alternatives. The Pareto front is used to first gain an overview of the interdependencies of the different objectives and then to choose suitable ϵ-sets for being added to the copies of the RCG model in the next phase.

2. **Railway traffic rescheduling phase:** pools of solutions to three disturbed scenarios are generated by concurrently solving several copies RGC models including the different sets of ϵ-constraints computed in the data preparation phase.

The rescheduling horizon is fixed to 40 minutes, from 7:20 a.m. to 8:00 a.m, for both the data preparation phase and rescheduling. This includes 8 freight trains and 36 passenger trains. Given the periodic nature of traffic in the area, the results of data preparation do not depend on the rescheduling horizon and the rescheduling process can be launched at any time using the same ϵ-sets.

The blocking time stairways patterns to for the RCG model are generated using the simulation-based approach described in Section 4.2.3

and implemented in C++ for the previous experiment. A total of 1'669 blocking time stairway patterns are generated for the 44 trains. Each blocking time stairway corresponds to a station-to-station run in which each block's speed limit is set to either maximum track speed, 80 km/h, or 50 km/h. Speed limits are set at block level because currently, and until all trains are equipped with cab-signalling or driver advisory systems, signals are the only mean to transmit rescheduling decisions from the traffic management to the drivers.[4]. The Swiss railway network is being progressively equipped with type N signals which can display speed restrictions in multiples of 10 km/h. This and the willingness to consider essentially different speed profiles but limit the number of decision variables motivates the choice of fixing the blocks' speed limits to three different values: maximum track speed, 80 km/h, or 50 km/h.

5.5.1 Data preparation phase and Pareto front

The Pareto front is computed using an adaptive ϵ-constraint method as described above. The underlying RCG model is built using C++ and solved using CPLEX (version 12.6.3). Figure 5.8 shows the Pareto-optimal solutions obtained (for the given blocking time stairways). From this figure it is possible to recognise some interdependencies of the stated objective measures.

- Energy consumption and train delay can be both reduced by deferring some trains after the considered time horizon. This leads to large delays for passengers and freights. The blue and green points spreading out from the lower back corner of Figure 5.8 are linked to this phenomenon. However, the overall train delay and energy consumption are not completely proportional, as the figure also shows that solutions with similar overall train delay can have very different energy consumption and the other way round.

- The relation of train delay and customer delay is difficult to characterise. On the one hand, the overall train delay increases when passenger or freight trains are delayed. On the other hand, deferring trains can reduce the overall train delay within the considered horizon but increases customer delays.

- The relation between energy consumption and customer delay is more straightforward as energy consumption usually decreases with

[4]Consider, for instance, the automatic train sequencing system RCS-HOT that uses signals to provide speed recommendations to avoid conflicts at a very busy junction of the SBB network (Oettich and Caspar, 2015)

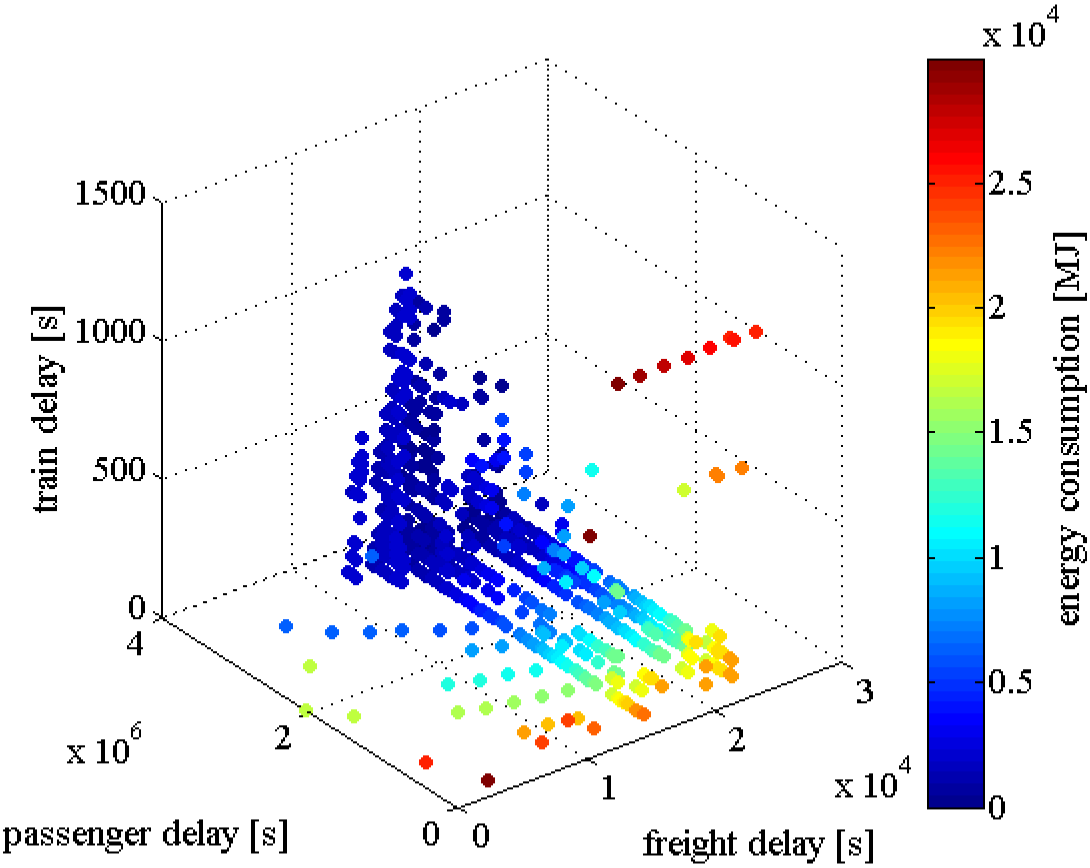
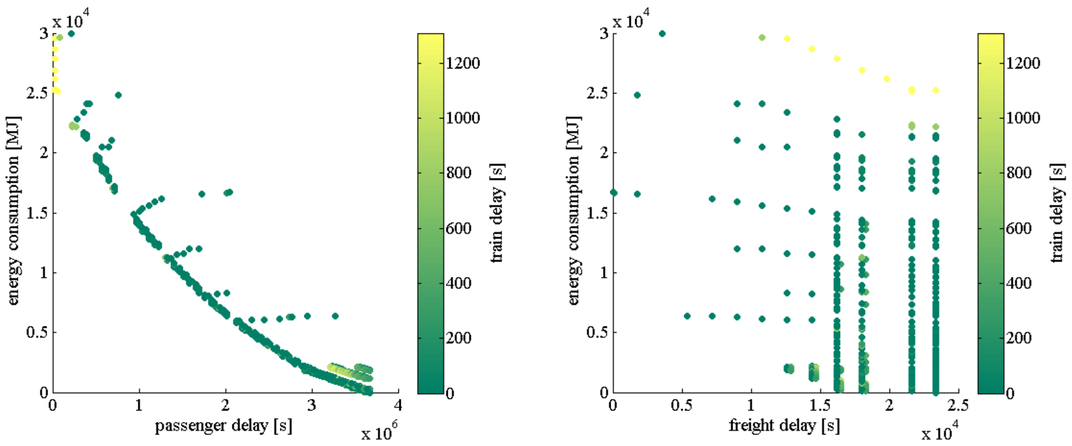


Figure 5.8: Pareto front, 40 minutes horizon (Toletti et al., 2017).

larger delays, which can result from either slower or deferred runs (Figure 5.9). Note that the projection with respect to passenger traffic is more complete than the freight one due to the prevalence of passenger trains in the test scenario (36 to 8). The almost horizontal lines inside the Pareto front in Figure 5.9(a) show that different combinations of freight and passenger trains can cause similar values of energy consumption.

- Another interesting relation appears between the two customer classes considered. In fact, if one wants to keep a given level of overall train delay and energy consumption (i.e., from operator's perspective) passenger delay can only be reduced at the costs of freights, and vice versa.

The main goal of multi-objective rescheduling is to provide dispatchers with alternative (Pareto-optimal) solutions. Thus, the ϵ-sets to be used during the rescheduling process are chosen in order to differentiate the solutions. For the current numerical experiment, the following sets are selected from the Pareto front in Figure 5.8:



(a) passenger trains (b) freight trains

Figure 5.9: 2D projections of the Pareto-optimal solutions to the numerical experiment.

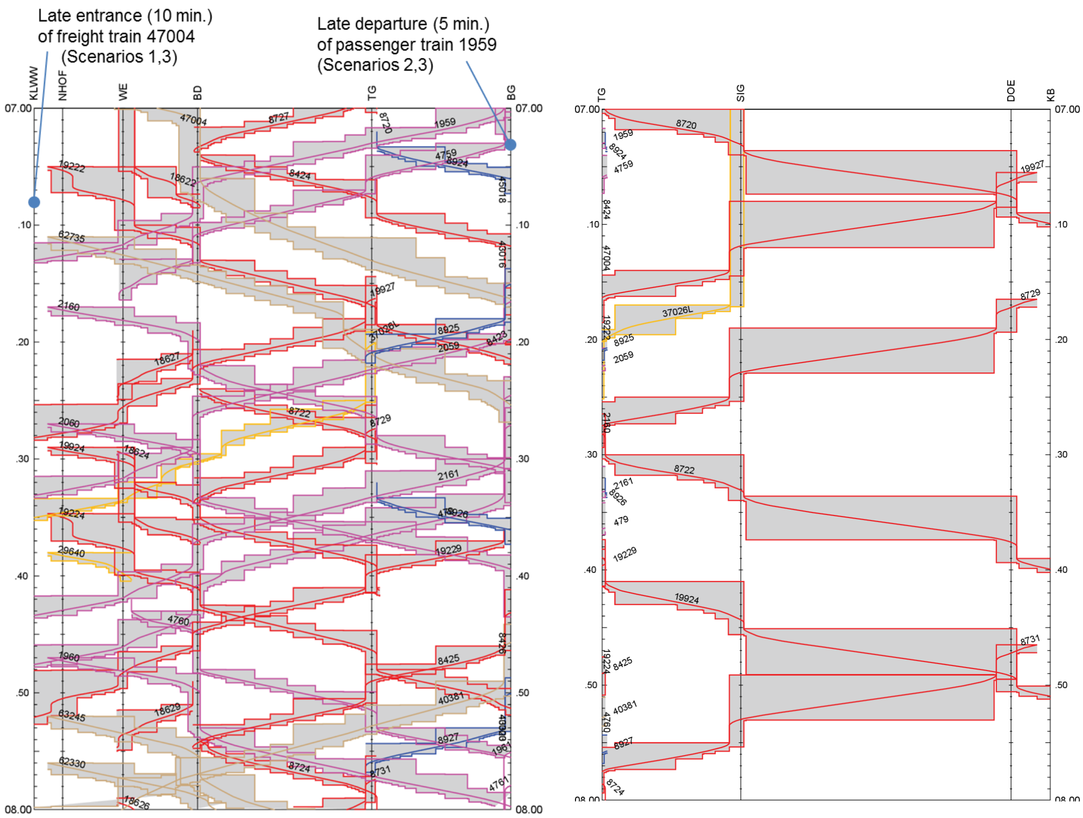
ϵ-set 1 High energy demand and low train delay: Deferring a large amount of trains to reduce delays and energy consumption in one time horizon is not really an option for traffic managers. Thus, a lower bound for energy consumption is needed to force the algorithm to schedule as many trains as possible. The delay is minimised by the objective function. This ϵ-set corresponds to the Pareto-points in the lower front corner in Figure 5.8 ($\underline{\epsilon_e} = 24821$, $\overline{\epsilon_e} = 29971$).

ϵ-set 2 Low passenger delay: Passenger trains often have higher priority in case of conflict than freight trains. The minimum passenger delay in Figure 5.8 corresponds to the eight points in the top right part. ($\overline{\epsilon_p} = 26200$).

ϵ-set 3 Low passenger delay and limited freight delay. Moving down (with respect to train delay) from the previous case, it is possible to reduce freight delays giving up only a small amount of passenger delay ($\overline{\epsilon_p} = 87620, \overline{\epsilon}_f = 10800$).

Since a large amount of the variety in the solutions is obtained by deferring some trains after the rescheduling horizon, but this strategy cannot be considered an improvement of railway traffic, two ϵ-sets trying to force all trains to be scheduled are also considered:

ϵ-set 4 Schedule all train runs in the considered horizon. Forcing freight and passenger delays under the cancellation penalty values, it is possible to find solutions without deferring runs after the horizon ($\overline{\epsilon_p} = 35999, \overline{\epsilon}_f = 1799$).



(a) main corridor (b) side corridor

Figure 5.10: Scenarios for multi-objective rescheduling (Toletti et al., 2017).

ϵ-set 5 Intensive train operations. Defining a range of energy that have to be consumed to move the scheduled trains, it may be possible to find good solutions with a low number of trains deferred and low train delay thanks to higher train speeds ($\underline{\epsilon_e} = 36749$, $\overline{\epsilon_e} = 44915$).

5.5.2 Railway traffic rescheduling phase

Rescheduling of the original timetable and of three disturbed scenarios is performed through concurrently solving five copies of the RGC models, each augmented by one of the previously defined ϵ-sets. The test scenarios are based on the yearly timetable for the main line between station Brugg and the Killwangen junction and the side corridor going north from station Turgi to Koblenz. Figure 5.10 shows the original timetable (Scenario 0) and the primary delays of the three disturbed scenarios:

- Scenario 1: Passenger train 2065 departs from BG with a delay of three minutes.

Table 5.2: Numerical experiment results

Scenario	ϵ-set	train delay [min] (objective value)	freight delay [min]	passenger delay [min]	energy required [MJ]	# runs deferred
0	1	0	180	30600	24907	80 (23 trains)
	2	0	300	0	33233	29 (5 trains)
	3	0	60	1200	37505	10 (2 trains)
	4	0	0	0	40832	0
	5	0	180	4200	37073	21 (7 train)
1	1	0	180	29400	25195	81 (21 trains)
	2	0	390	0	28374	39 (8 trains)
	3	0	90	1200	35738	12 (4 trains)
	4	Infeasible				
	5	0	150	3000 min	37741	20 (5 trains)
2	1	0	180	26400	25667	77 (21 trains)
	2	6.36667	390	127.333	28422	39 (8 trains)
	3	0	180	1200	30820	27 (6 trains)
	4	6.36667	0	127.333	40063	0
	5	0	90	7200	36751	28 (7 trains)
3	1	0	270	19800	24855	68 (19 trains)
	2	6.36667	390	127.333	28076	51 (8 trains)
	3	0	180	1200	32097	37 (5 trains)
	4	Infeasible				
	5	3.63333	60	4800	36839	16 (5 trains)

- Scenario 2: Freight train 50713 departs from WE with a delay of five minutes.
- Scenario 3: Cumulation of scenario 1 and 2.

For each scenario, the RCG models are built using an adaptation of the C++ program developed for the previous experiment and concurrently solved with CPLEX on one node of Euler, the computer cluster of ETH Zurich, equipped with two 12-core Intel Xeon E5-2697v2 processors (2.7 GHz nominal, 3.0–3.5 GHz peak) and 64 GB of memory clocked at 1866 MHz. Table 5.2 shows the results.

The rescheduling process tends to defer some train runs after the considered horizon to expand the solutions' pool. This practice is hardly applicable in real-world rescheduling, where the deferred runs would be moved after the end of the rescheduling horizon and probably interfere with the successive runs. To account for the expected consequences that the deferred movements might have on the successive traffic, a representation of terminal costs in the objective function is needed.

Table 5.3 shows the computation times of the four test scenarios. Building the model requires about 45 seconds and solving it about 6

Table 5.3: Numerical experiment computation times

Scenario	Setup [s]	Model building [s]	Model solving[s]
0	5.74	44.55	312.37
1	1.51	45.68	302.00
2	2.60	44.35	326.67
3	2.25	45.13	306.32

minutes, which gives an overall time of about 7 minutes for a 40 minutes long rescheduling horizon. These times might be reduced by shortening the horizon length as the next subsection shows.

5.5.3 The effect of terminal costs

Given the number of train runs deferred after the considered rescheduling horizon during the previous experiment, terminal costs are added to the objective function. The terminal costs are set equals 10’000’000 seconds for the next movement of each train and 10’000 for the successive ones. These values are set arbitrarily to check whether the introduction of terminal costs actually solves the problem of deferred train runs. A discussion about the appropriate size of terminal costs is presented at the beginning of the next experiment (Section 5.6). In addition, the size of the rescheduling horizon is halved, i.e., reduced to twenty minutes, to check whether shortening the horizon makes solution times compatible with real-time traffic management.

The new Pareto-front is shown in Figure 5.11. In this case, all train movements are scheduled within the current rescheduling horizon. As a consequence, there is a much smaller variation in terms of all objectives. In particular, there is one area characterised by highest energy consumption (bottom front points in the figure) and one with lower consumption, where one train class (passenger or freight) receives priority over the other. The ”angle” in the figure corresponds to the switching point from prioritising one category to the other. It is important to note, that the maximal energy saving is now about 0.4 %. This is due either to the lack of residual capacity in the main corridor to be used for energy saving, or to the small variation in energy consumption of the speed profiles used to generate the blocking time stairway patterns, or a combination of these elements.

Three ϵ-set are defined from the new Pareto-front. The first set aims at bounding the overall passenger delay ($\overline{\epsilon}_{passenger}$ = 500+initial delay of passengers [$seconds$], notation as in Section 4.2.2). The second aims at

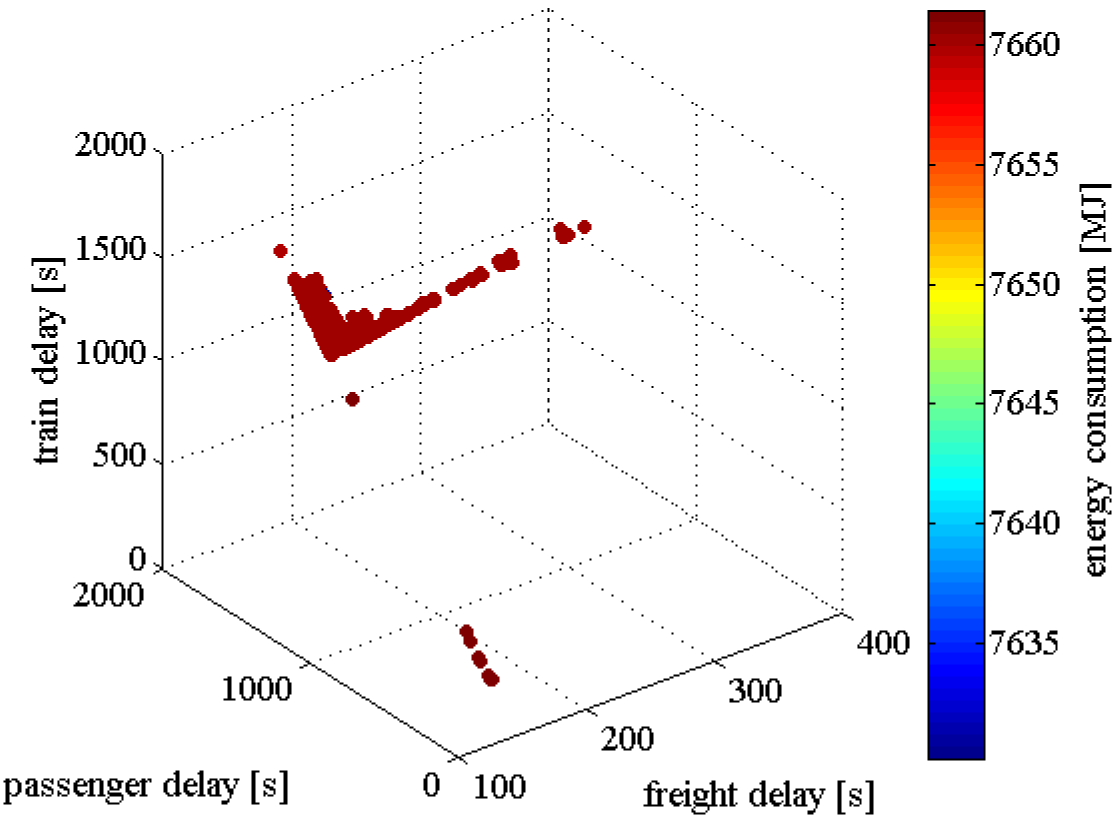


Figure 5.11: Pareto front for multi-objective rescheduling with terminal costs, 20 minutes horizon.

bounding the overall freight train delay ($\overline{\epsilon}_{freight}$ = 180+ initial delay of freight trains [$seconds$]). The third tries to reduce energy consumption ($\overline{\epsilon}_{energy}$ = 7635 [MJ]). As for the previous experiment, these sets are added to copies of the RCG model that are solved concurrently. The same test scenarios and compute server as the previous experiment are used. Table 5.4 shows the results.

Most of the computation times are now around ten seconds, which indicates a superlinear relation with the size of the rescheduling horizon, and thus with the number of decision variables in the RCG model. In general, much less variability than in the previous case can be observed and the results of the first two ϵ-sets are very similar to the ones with no ϵ constraint (ϵ-set 0).

Table 5.4: Results of multicriteria rescheduling in BG-KLWW partition of SBB case study.

Scenario	ϵ-set	train delay (objective value) [s]	freight delay [s]	passenger delay [s]	energy consumed [MJ]	computation time [s]
0	0	250 (-60389750)	137	113	9997.9	8.6
	1	250 (-60389750)	137	113	10471.1	6.7
	2	250 (-60389750)	137	113	9981.2	5.1
	3	318 (-60379682)	137	181	7540.2	16.9
1	0	250 (-60389750)	137	113	10169.7	18.7
	1	250 (-60389750)	137	113	10704.0	8.1
	2	250 (-60389750)	137	113	10341.7	7.2
	3	1374 (-60378626)	1214	160	7599.5	29.8
2	0	640 (-60399360)	371	269	10260.6	10.3
	1	640 (-60399360)	371	269	11079.1	13.3
	2	640 (-60399360)	371	269	10549.9	11.6
	3	2251 (-60377749)	1618	633	7023.9	87.5
3	0	640 (-60399360)	371	269	11148.0	14.1
	1	640 (-60399360)	371	269	10759.7	17.7
	2	640 (-60399360)	371	269	10981.5	12.7
	3	1946 (-60378054)	1617	329	7437.6	43.4

Lessons learned:

1. *The RCG model supports several different objectives including train delays, customer inconvenience and energy efficiency;*

2. *The objective function must include terminal costs for train movements that cannot be scheduled within the considered horizon to account for the expected consequences on railway traffic after the end of that horizon;*

3. *The implemented C++ rescheduling algorithm can concurrently handle different RCG models, which supports the development of algorithms running parallel processes (e.g., the problem decomposition and negotiation-based coordination approach proposed in Section 4.1);*

4. *The computation times for a large condensation zone are not satisfactory and highlight the need for improved model resolution algorithms.*

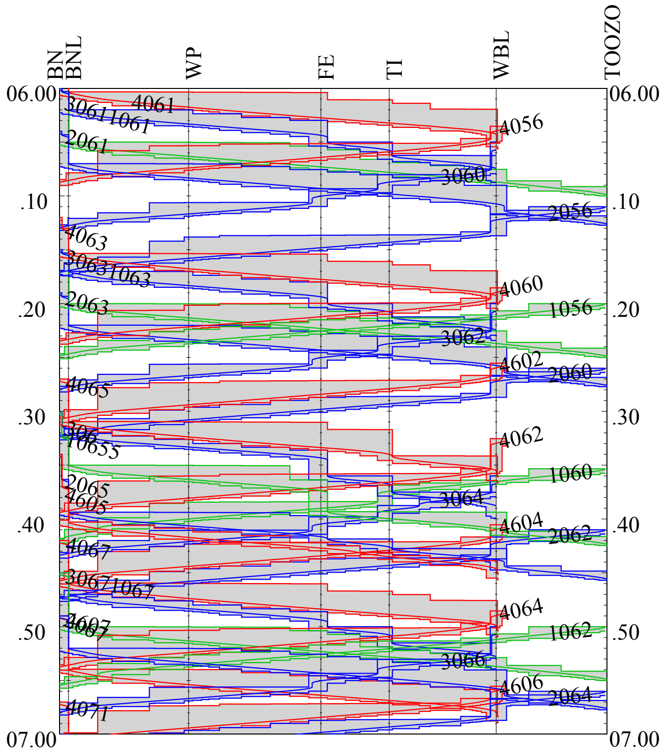


Figure 5.12: Regional traffic between Berne and Worblaufen (RBS).

5.6 Column generation approaches for RCG model in a real-world condensation zone with simple topology

In this section, the column generation approaches from Section 4.4 are tested on the RBS test network, using the 2017 timetable (Figure 5.12). The aim is to assess their effectiveness in producing good quality schedules in less computation time than the monolith for a real-world condensation zone with simple topology and extremely high traffic density.

For these experiments, the size of the rescheduling horizon is fixed to fifteen minutes, the timetable period. Without loss of generality, the current study considers only train delays and neglects customer delays, passenger inconvenience, and energy consumption. For all experiments in this section, the objective function is the sum of the overall train delays (equation 4.18) and the terminal costs of train runs that cannot be scheduled in the considered horizon, and thus have to depart in the next period. Terminal costs represent the expected additional overall delay that those train movements will cause in the next period. Given that the considered timetable is very dense, it is sensible to suppose that deferring a train run to the next period will cause another train to fall out from the following rescheduling horizon, which is equivalent to a delay of the size of the horizon. For this reason the terminal costs of each train movement is set equals to the length of the rescheduling horizon. In this case, it is equals to the timetable period, i.e., fifteen minutes. To prioritise the assignment of a blocking time stairway to trains that are already on the line between two operation points over movements happening in future, the terminal costs of these movements are ten times the rescheduling horizon size.

First, different algorithm's parameter settings are tested on a common scenario to assess their influence on the algorithm's computation time, the convergence behaviour, and the quality of the new schedule. Of major interest is the improvement with respect to the monolith. Then, the parameters are tuned accordingly, and further experiments on different delay scenarios are performed. The investigation focuses on the quality of the schedules generated by the resulting railway traffic rescheduling algorithm.

5.6.1 Algorithms and parameters testing

To gain an overview of the computation time, convergence behaviour, and yielded solution quality of the monolith and column generation algorithms, several simulations with a common delay scenario are performed on the RBS test case. All simulations start at 6:20 a.m. and end at 6:45 a.m., the rescheduling horizon is set from 6:30 a.m. to 6:45 a.m. The following parameter settings are investigated:

- Blocking time stairway patterns: A database of 1'961 blocking time stairway patterns for all train movements is available. This is generated a-priori using the C++ program of the simulation-based patterns generator described in Section 4.2.3 that has been implemented for the previous experiments. Three different speed commands can be passed at each signal: maximum track-speed, 50 km/h, 30 km/h.

- Monolith: the monolith is tested using six different time shift sizes for the blocking time stairways copies ($3s$, $6s$, $10s$, $15s$, $30s$, $60s$) and two databases of blocking time stairway patterns (rerouting + speed advisory, rerouting only).

- Initial restricted master RCG model for column generation: the initial restricted RCG model is built using different quantities of time-shifted copies (10, 20, 30) and six different time shift sizes ($3s$, $6s$, $10s$, $15s$, $30s$, $60s$).

- Pricing problems: both arc and path pricing problems are tested in both variants (with precomputed blocking time stairways and constructive), i.e. all cgApP, cgApC, cgPpP, and cgPpC. Six different time shift sizes are tested ($3s$, $6s$, $10s$, $15s$, $30s$, $60s$).
- Master-pricing iterations: at most ten master-pricing iterations are allowed for each experiment. Fewer iterations are performed if either no new column can be generated, or both the delay of each event is smaller than 60 seconds and all runs have an allocated blocking time stairway. Both LP relaxation settings (relaxed restricted master RCG with dual prices, MILP restricted master RCG with slack-based resource prices) are tested.
- Terminal costs: 15 minutes, i.e., equals the length of the rescheduling horizon and timetable period.

For these tests the rescheduling program implemented in C++ for the previous experiments is further developed and runs on a 64-bit linux server with eighty CPUs Intel(R) Xeon(R) CPU E7-4870 at 2.40GHz and 528 GB memory (RAM).

5.6.1.1 Tuning the parameters of column generation with arc pricing problems

In this section, the column generation approach with the different arc pricing problems to solve the RCG model (4.1-4.17) is analysed and compared with the monolith. Figure 5.13 shows the computation time versus the solution quality for different algorithms and parameters. The objective value per computation time of the monolith (black circles) is the reference value to evaluate the column generation approaches. In all cases, the times for generating the blocking time stairway patterns and building the pricing problems are not considered because these steps can be performed offline during a data preparation phase.

Both the objective value and the computation time of the monolith strongly depend on the time shift used to generate the blocking time stairway copies that are linked with the model variables: the smaller the time shift, the lower the objective value and the higher the computation time. The objective values and computation times of column generation approaches with constructive pricing problems (cgApC) are comparable with the monolith and are quite sensitive to the parameter settings. For

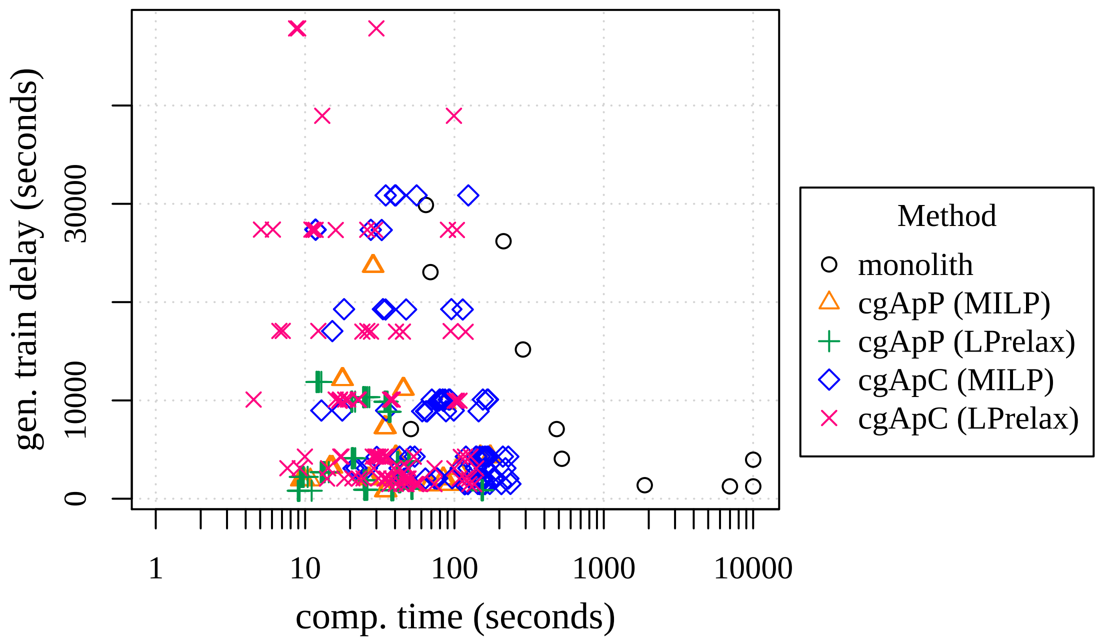


Figure 5.13: Objective value vs. computation time of the monolith and different column generation aproaches with arc pricing problems. Please note the logarithmic scala applied to the time (x-axis).

finer time grids the pricing graphs of cgApC could not be initialised due to memory issues (see also Figure 5.14). The column generation approaches with precomputed blocking time stairways (cgApP) show the best performance in terms of objective value per computation time. In particular the LP relaxed version, where most experiments converge to values quite close to the optimum (see also Figure 5.14).

Figure 5.14 shows the convergence of the four column generation approaches with arc pricing problems (precomputed bts./constructive, MILP/LP relaxation). In most cases without LP relaxation, very little improvement is achieved with later master-pricing iterations, although the optimum is still quite far. Thus, the LP relaxed versions should be preferred.

Figure 5.15 shows the time statistics of the pricing problems. The station sections (i.e., the ones for entering/leaving Berne and Worblaufen) show the highest average computation times. In all cases, the time needed by the pricing problems with precomputed blocking time stairways is much smaller than the one needed by the constructive approach. In the latter, the prices updating routine takes the largest share of the overall time. Finally, Figure 5.16 shows how the size of the pricing graph depends on the time grid applied. Both the numbers of arcs and nodes become quite large for finer time grids.

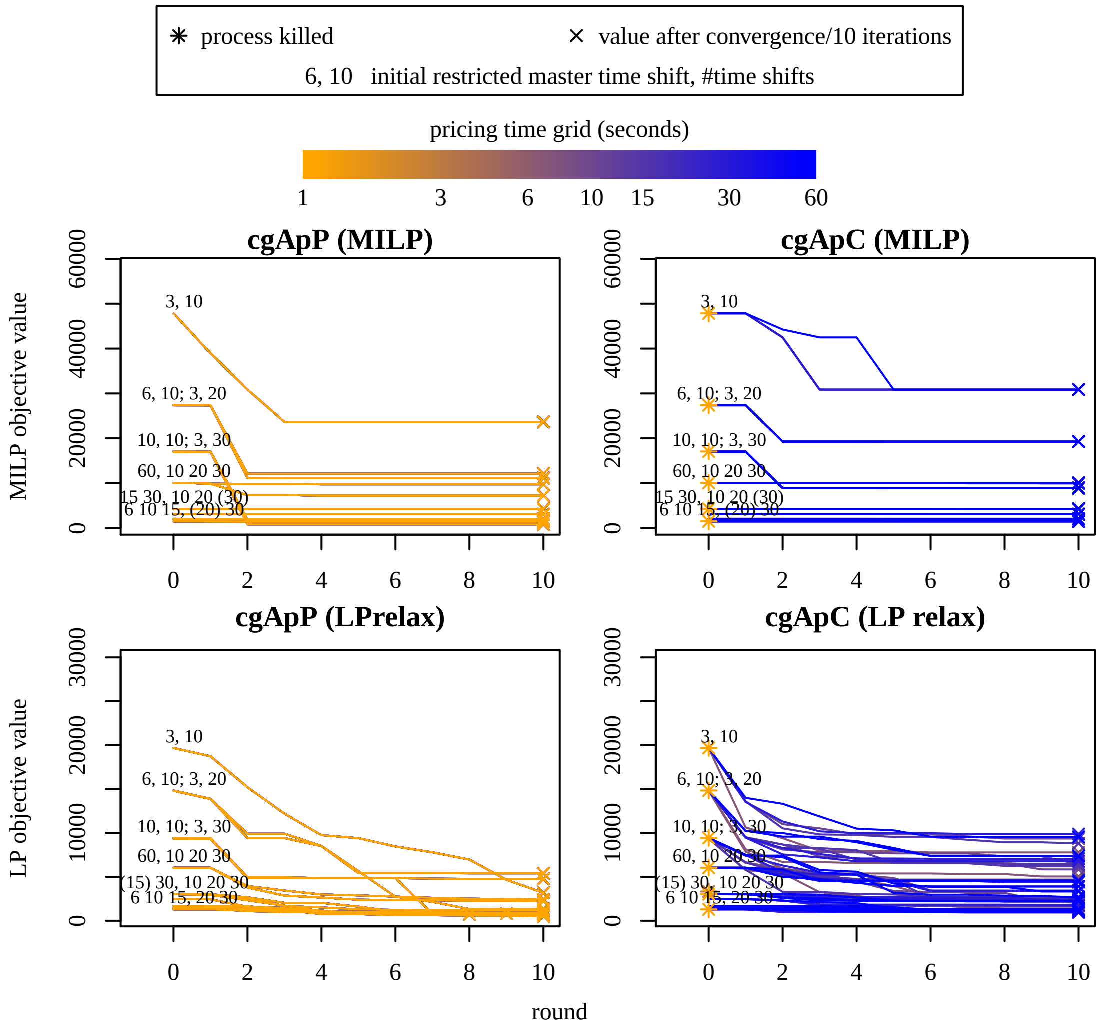


Figure 5.14: MILP and LP convergence of arc pricing approaches.

5.6.1.2 Tuning the parameters of column generation with path pricing problems

In this section, the different pricing approaches applied to the ”path” version of RCG model (4.32-4.38) are analysed. Figure 5.17 shows their objective values and computation times for the different algorithms and parameters. The monolith performance from the previous subsection (black circles) is used as reference. Here, the monolith often outperforms all current implementations of the path pricing approaches in terms of objective value per solution quality. Again, the approaches with precomputed blocking time stairways produce lower objective values than the constructive ones, which have been often killed before finishing due to extremely high RAM requirements.

Figure 5.18 shows the convergence of the four column generation approaches with path pricing problems (precomputed bts./constructive, MILP/LP relaxation). Again, in most cases, very little improvement is

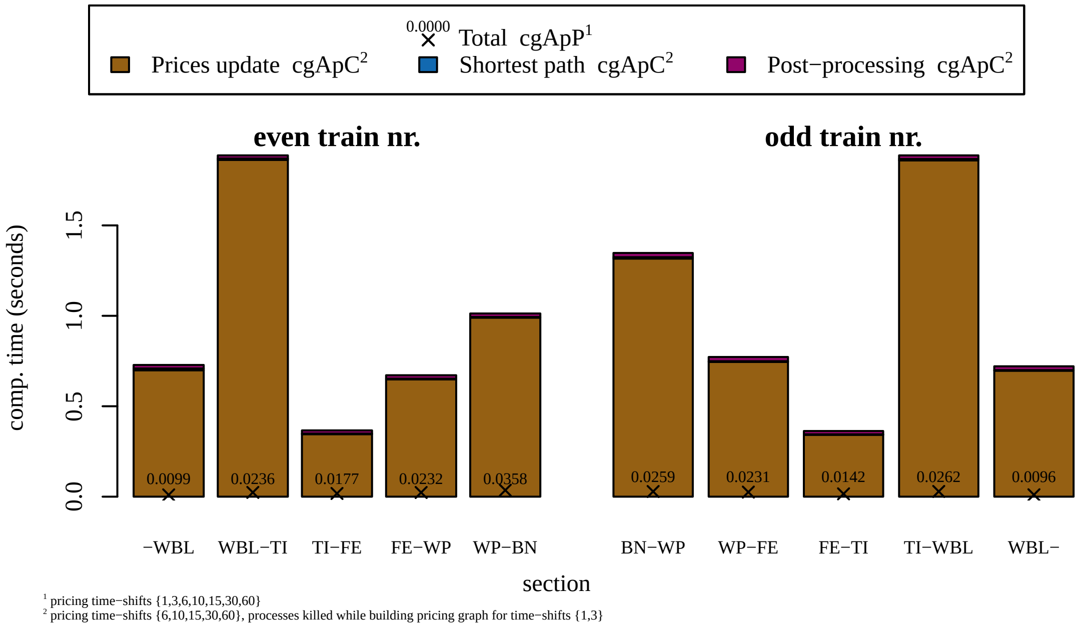


Figure 5.15: Average computation time for generating one new column to be added to the master.

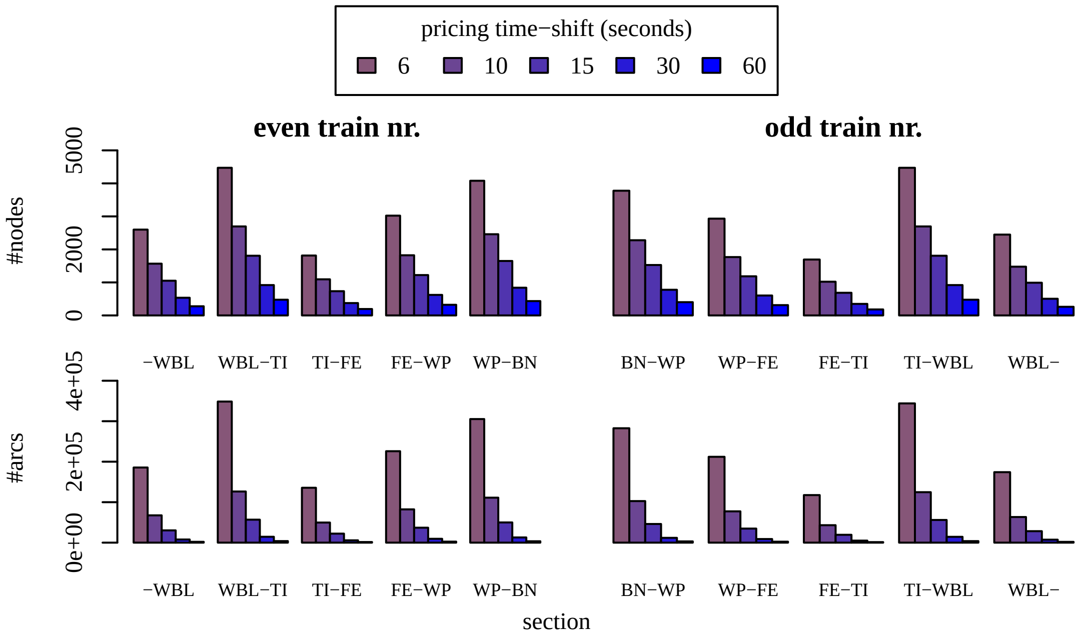


Figure 5.16: Size of pricing graphs for constructive arc pricing.

achieved at later master-pricing iterations. There is an early convergence for larger pricing time shifts (due to the fact that in areas with many short resources the rolling horizon is not sufficient to travel through the entire graph), while the process is always killed for smaller pricing time shifts. Figure 5.19 shows the time statistics of the pricing problems. Again, the average time needed by the pricing problems with precomputed blocking

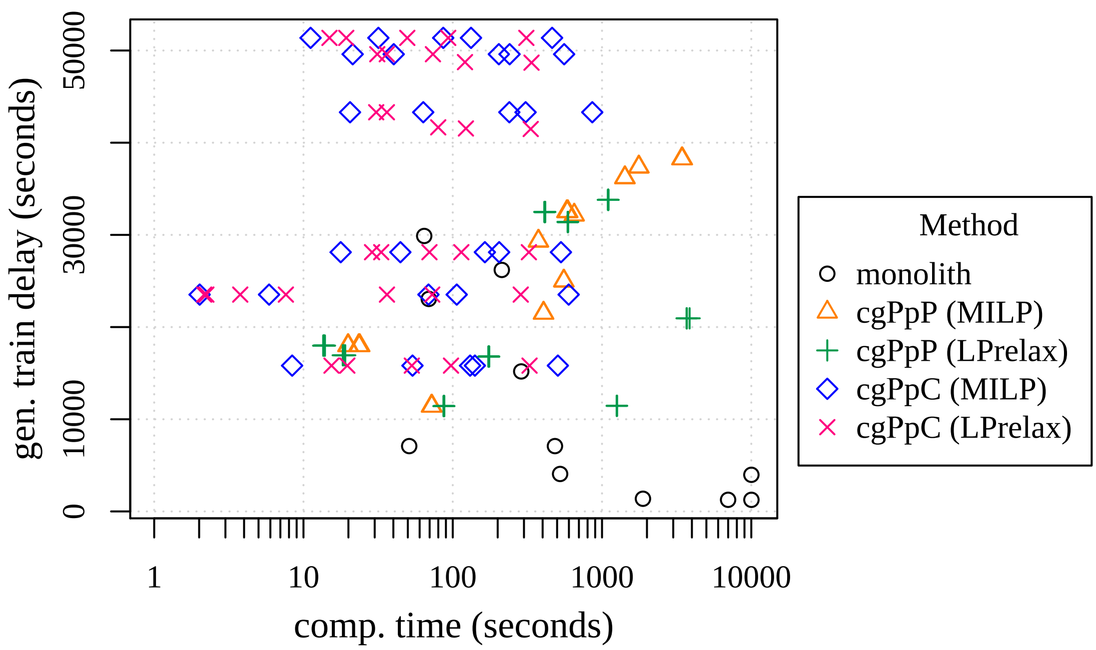


Figure 5.17: Objective value vs. computation time of the monolith and different column generation approaches with path pricing problems. Please note the logarithmic scala applied to the time (x-axis).

time stairways is much smaller than the one needed by the constructive approach. In the latter, the prices updating routine takes the largest share of the overall time.

Figure 5.20 shows how the size of the pricing graph depends on the time grid applied. Both the numbers of arcs and nodes become quite large for finer time grids.

5.6.1.3 Arc vs. path pricing problems

Figure 5.21 shows the correlations of the different algorithm's parameters, RCG model sizes, objectives, and computation times. In both arc and path pricing problems constructive approaches return higher (worse) objective values, and in the arc pricing case it also yields higher computation times than the reference approach (pricing problems with precomputed blocking time stairways and restricted master problem without LP relaxation). Figure 5.21(b) might suggest that constructive path pricing problems produce smaller models (fewer blocking time stairways and constraints) and a faster convergence than the reference. Although, this is not true. The negative correlation is explained by the fact that, in many cases, column generation approaches with constructive path pricing problems

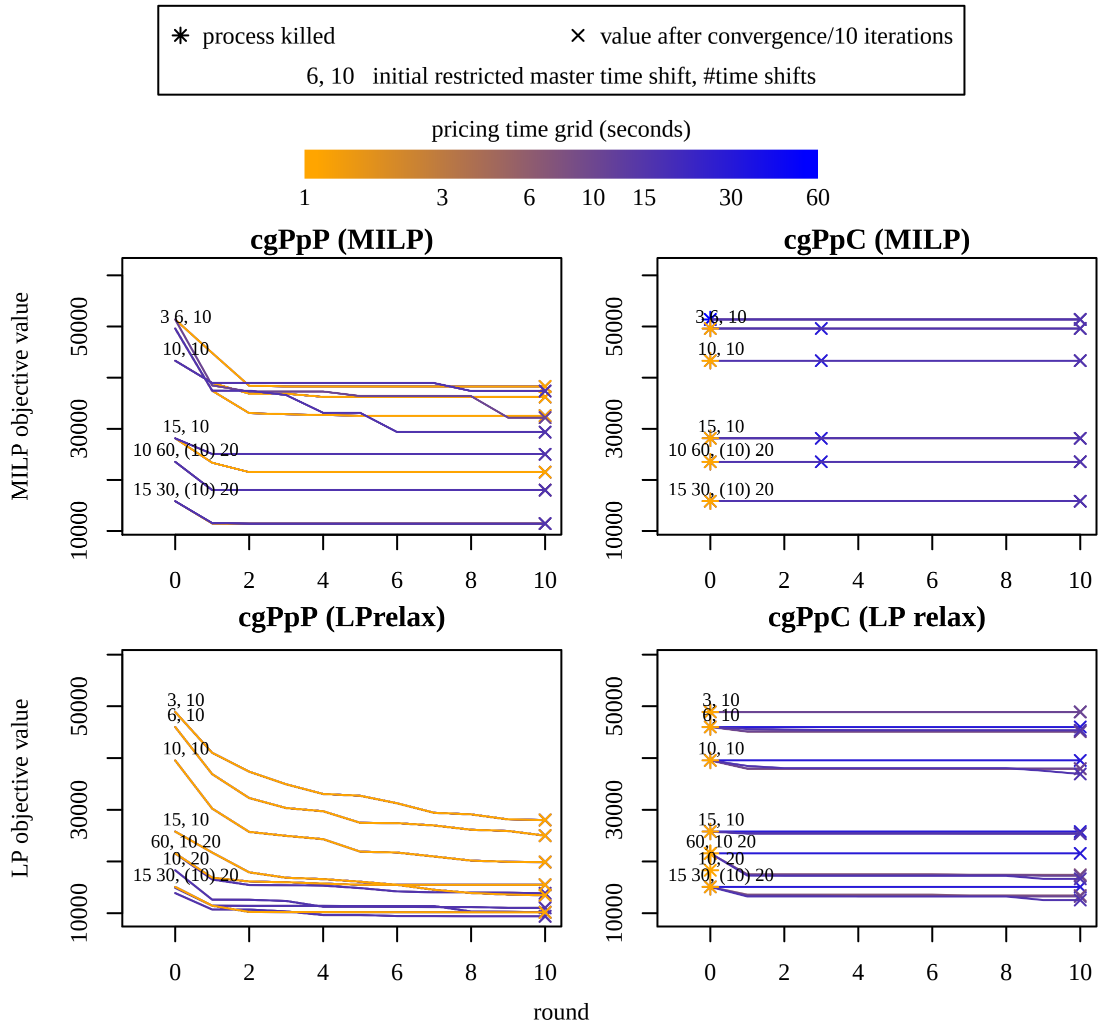


Figure 5.18: MILP and LP convergence of path pricing approaches.

have been killed while building the pricing graphs due to prohibitive memory requirements.

The most computational expensive phase of constructive pricing problems (both arc and path) is the computation of the prices for the graph's arcs (figures 5.15 and 5.19). Figure 5.22 shows how the computation time of the pricing problem depends on the size of the pricing graph. While Figure 5.23 suggests a much milder increase of the computation time of the pricing problem with growing numbers of candidate blocking time stairways for cgApP.

These experiments show clearly that the cgApP algorithm with LP relaxation of the restricted master is the most promising approach to compute new adapted schedules in short times. Thus, it is selected for further experiments.

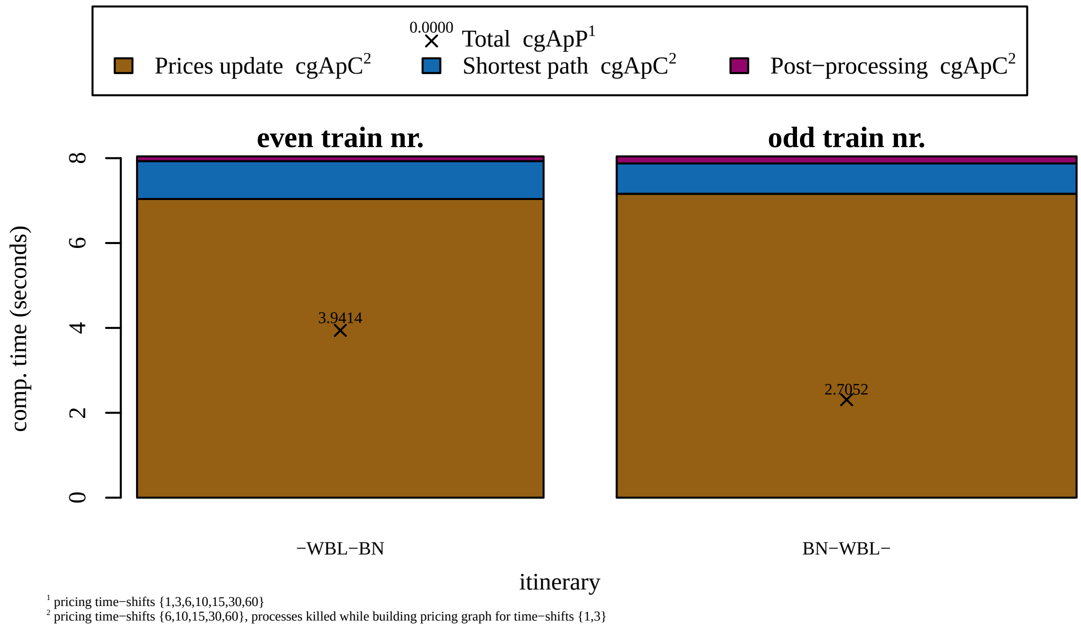


Figure 5.19: Average computation time for generating one column to be added to the master (path pricing).

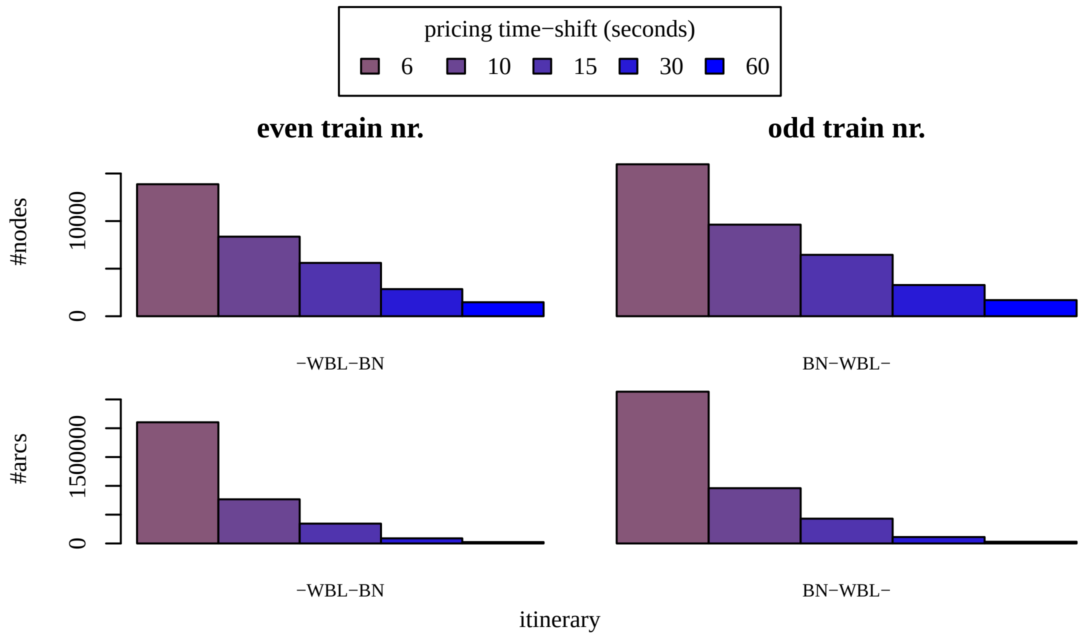
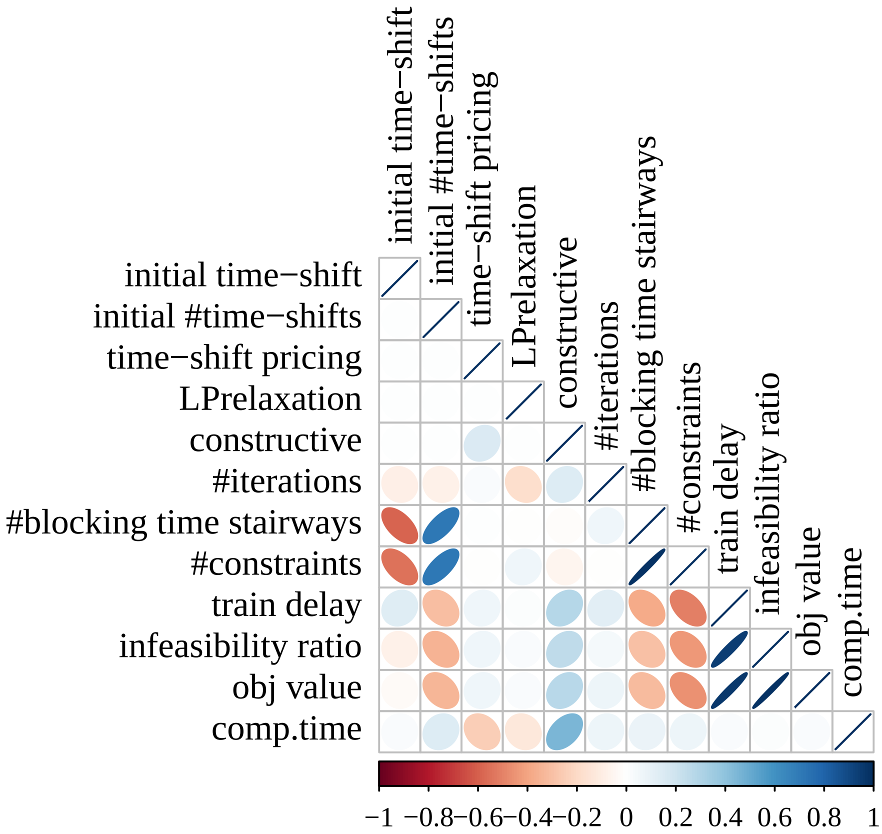
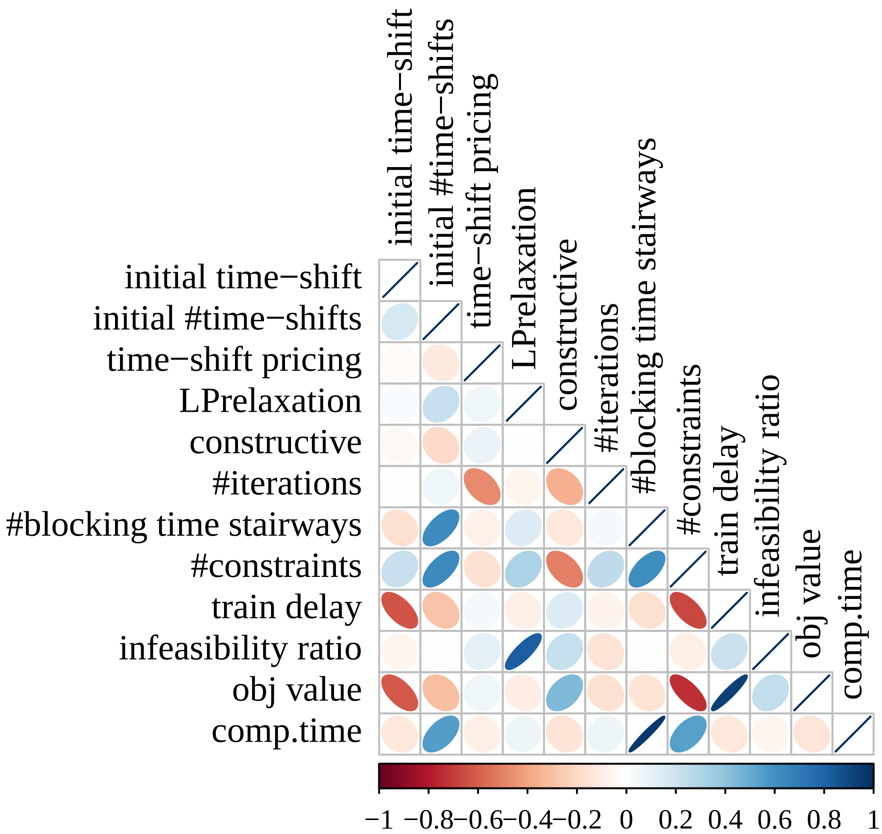


Figure 5.20: Size of pricing graphs for constructive arc pricing.

Figure 5.21: Correlations



(a) ”arc”



(b) ”path”

Figure 5.22: Size of pricing graphs vs computation time for constructive pricing.

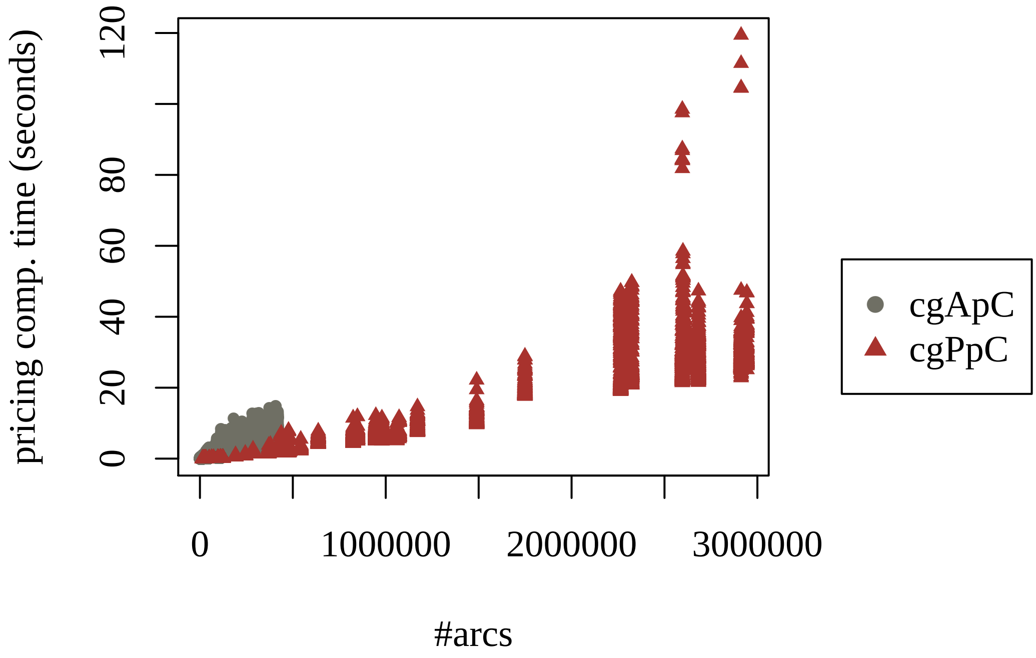
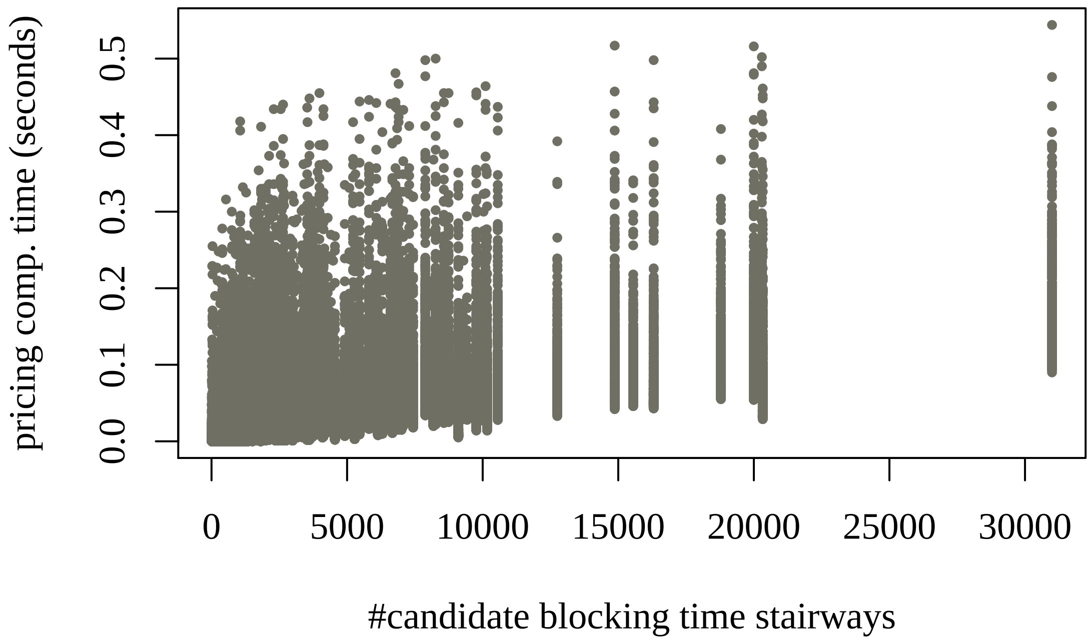


Figure 5.23: Number of candidates vs computation time for arc pricing with precomputed blocking time stairways.



5.6.2 Rescheduling different delayed scenarios

The most promising algorithm from the previous analysis is tested on different delay scenarios in this section:

- algorithm = cgApP with LP relaxation (best solution quality per computation time performance, Figure 5.13, and convergence behaviour, Figure 5.14);
- time-shift in initial restricted master = 10 s (infeasibility could be always resolved);
- number of time-shifts in initial restricted master = 10 (shorter computation times, Figure 5.21(a));
- time-shift in pricing problems = 3 s (higher precision without significant loss of computation time, Figure 5.23);

For these tests the rescheduling program (C++ implementation from previous experiments) runs on a 64-bit linux desktop computer with four CPUs Intel(R) Core(TM) i5 CPU 760 at 2.80GHz and 7.7 GB memory (RAM).

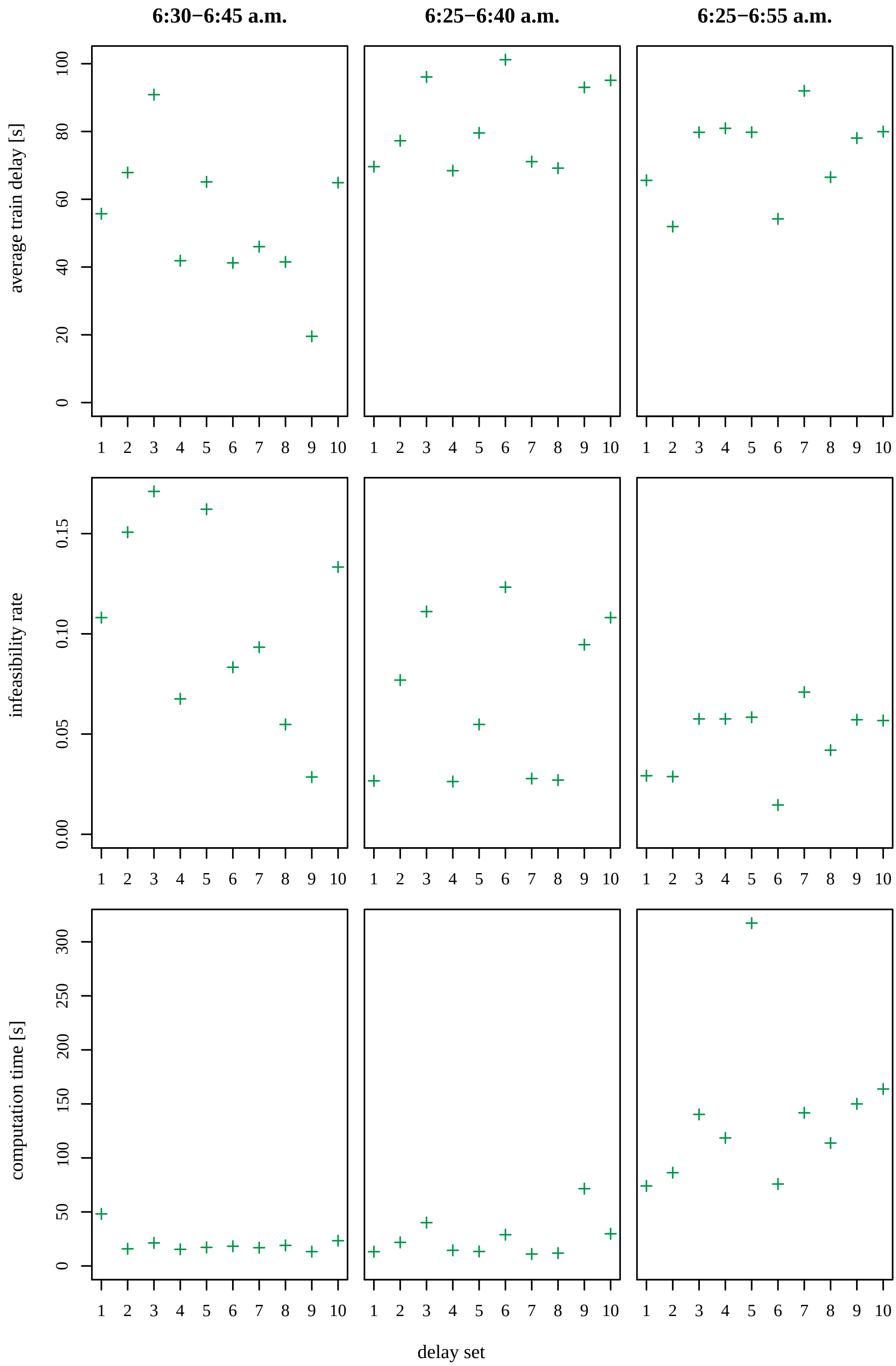
The tests are run on three different rescheduling horizons:

1. 6:30-6:45 a.m., same as for parameter testing of previous section;
2. 6:25-6:40 a.m., to fully include the movements connected with the ”6:30-node” of RBS fully integrated periodic timetable;
3. 6:25-6:55 a.m., two timetable periods long rescheduling horizon.

Each experiment includes a ten minutes training phase before the rescheduling horizon. In the training phase, initial delays are added to departure and train passing events according to the Weibull probability distribution for regional passenger trains calibrated on SBB data and described in Section C.2 (parameters: shape = 1.76, scale = 123.01, shift = -73.5). Ten experiments, each with a different delay scenario, are executed for each of the three rescheduling horizons.

First, the solution quality per computation time of the algorithm is analysed in Section 5.6.2.1. Then, the resulting schedules are compared against the standard first come first served policy of the simulator in Section 5.6.2.2.

Figure 5.24: Results of rescheduling several delayed scenarios of RBS test case with cgApP algorithm with LP relaxation.



5.6.2.1 Performance of cgApP

For each delay set and rescheduling horizon, cgApP produced a new schedule. Figure 5.24 shows the statistics of the 30 experiments. The average train delay differs considerably among the delay sets. The number of train movements that cannot be scheduled within the rescheduling horizon (infeasibility rate) is quite large for the shorter horizons (about 10 %), and it is smaller for the longer horizon (about 5 %).

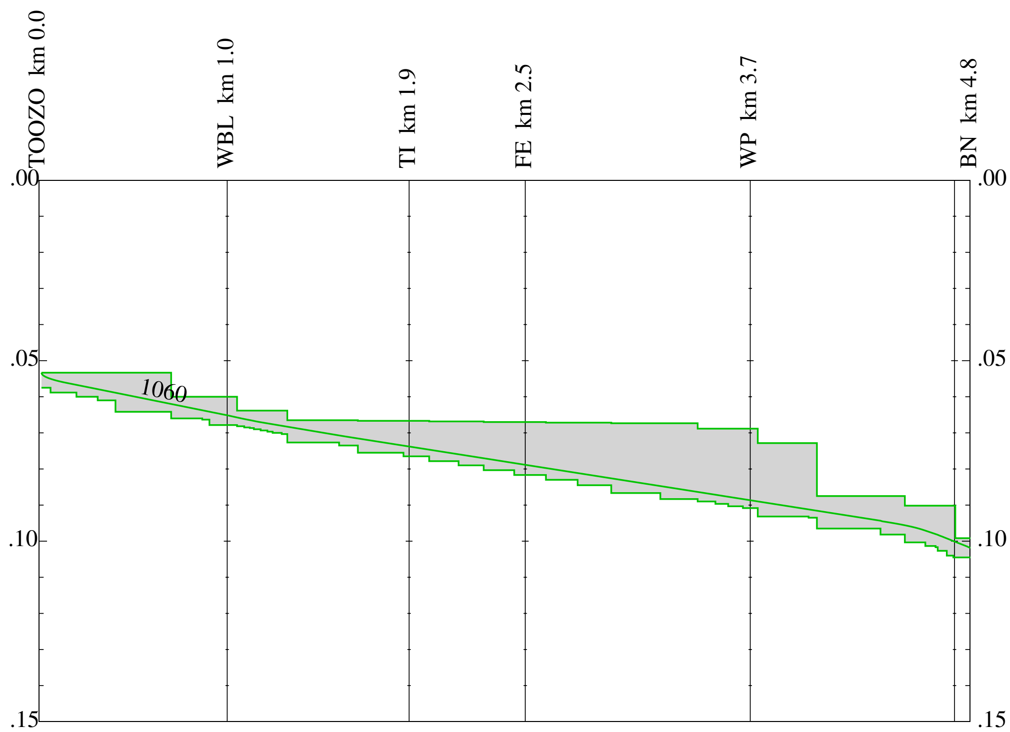
The relatively high number of train movements hat cannot be scheduled within the corresponding horizons can be explained with the scarce residual capacity of the considered section that usually requires trains to travel quite close to the preceding ones. This kind of operation is not supported by the patterns in the database of blocking time stairways used to initialise the restricted master RCG problem and define the pricing problems. The patterns have been generated using the simulation-based approach from Section 4.2.3, which requires each train to be simulated separately to conform to the green wave principle. It follows that the train routes tend to be reserved long time in advance, given that the infrastructure resources are not needed by any other train. In fact, the analysis of these patterns reveals that some include extremely long route reservation times (see, e.g., Figure 5.25).

Most of the computation times for the first horizon are around 20 seconds but the first experiment which needs almost 50 seconds to complete. For the 6:25-6:40 a.m. horizon, the computation times are comparable to the ones of the 6:30-6:45 a.m. horizon and range from 13.2 to 71.53 seconds. Longer computation times are needed for the 6:25-6:55 a.m. horizon, where times between 74.07 and 317.32 seconds are observed. Most computation times are below the aforementioned 180 seconds threshold (Rodriguez, 2007).

5.6.2.2 Comparison with first come first served

Each experiment is executed twice. The first time, the simulator is allowed to change the train orders according to a first come first served procedure (FCFS) that can be influenced by the train priorities and schedules (please refer Section 7.3 in OpenTrack Railway Technology Ltd., 2015, for further details). The second time, train orders and routes are set according to the new schedule computed by cgApP. Table 5.5 reports the overall train delays, the number of unplanned stops and braking actions as well as the number of rerouting and retiming actions. The last column indicates the computation time needed by cgApP for rescheduling each scenario.

Figure 5.25: Blocking time stairway pattern with long route reservation times.



Delays are computed according to OpenTrack output data with a tolerance of 60 seconds.

In seven of the ten experiments on horizon 6:30-6:45 a.m., rescheduling reduces the overall train delay up to the end of the considered horizon. In one case, the statistics of cgApP and FCFS are almost identical. In the remaining two, cgApP reduces either the number of unscheduled stops or braking actions but at the cost of increasing the other and the overall delay with respect to FCFS.

In six of the ten experiments on horizon 6:25-6:40 a.m., cgApP and FCFS schedules produced almost identical statistics. There are significantly less rerouting actions than the previous case.

In only one of the ten experiments on horizon 6:25-6:55 a.m., cgApP reduces the overall delay with respect to FCFS. Nevertheless, in seven experiments, cgApP improves traffic flows by significantly decreasing the number of unplanned stops and braking actions.

In general, the improvements of the traffic flows at expense of additional train delays are attributable to the used database of blocking time stairway patterns. In fact, FCFS allows trains to travel quite close to each other and encounter a number of yellow and red signals, while the database contains only conflict-free trajectories compatible with the green

Table 5.5: Comparison of rescheduling several delayed train schedules with cgApP with LP relaxation against FCFS.

Horizon	delay set	Train delay [s]		#stops at signals		#braking for route		energy [MJ]		#retimings	#reroutings	computation
		FCFS	cgApP	FCFS	cgApP	FCFS	cgApP	FCFS	cgApP	cgApP	cgApP	time [s]
6:30-6:45 a.m.	1	172	325	1	2	5	7	1987	1934	22	7	48.19
	2	133	226	4	5	7	6	2052	1918	20	9	15.86
	3	2082	1431	14	10	27	15	2318	2059	23	9	21.31
	4	270	199	5	6	11	9	2100	2112	16	7	15.4
	5	144	73	3	4	5	2	1984	1806	17	7	17.18
	6	135	64	3	4	5	4	2002	1954	17	5	18.21
	7	2090	1805	10	7	26	16	2355	2007	23	9	16.92
	8	244	173	1	2	6	8	2018	2086	15	5	19.05
	9	71	70	3	4	7	5	2048	2024	19	7	13.29
	10	964	343	5	4	5	2	2012	1857	20	9	23.37
6:25-6:40 a.m.	1	322	322	5	6	7	6	1925	1831	14	0	13.2
	2	712	235	8	9	11	4	2001	1260	13	0	21.84
	3	324	92	5	2	14	6	2024	1711	13	2	40.12
	4	219	219	4	6	15	14	1994	1855	15	0	14.46
	5	75	75	4	5	6	4	1907	1837	12	0	13.39
	6	618	1048	5	8	13	9	2025	1706	16	0	28.93
	7	130	130	3	5	6	5	1908	1769	10	0	11.08
	8	189	189	5	7	15	14	1993	1920	11	0	11.87
	9	126	155	3	3	13	10	1949	1912	12	0	71.53
	10	379	379	4	4	9	3	1943	1636	14	2	29.79

Table 5.5: (continued) Comparison of rescheduling several delayed train schedules with cgApP with LP relaxation against FCFS.

Horizon	delay set	Train delay [s]		#stops at signals		#braking for route		energy [MJ]		#retimings	#reroutings	computation
		FCFS	cgApP	FCFS	cgApP	FCFS	cgApP	FCFS	cgApP	cgApP	cgApP	time [s]
6:25-6:55 a.m.	1	201	351	8	8	12	11	3441	3418	15	2	74.07
	2	260	410	10	11	21	20	3526	3404	19	2	86.34
	3	197	313	8	7	19	16	3481	3366	19	0	140.29
	4	450	600	9	8	15	9	3476	3308	24	2	118.51
	5	295	445	8	7	12	7	3439	3269	19	9	317.32
	6	201	213	8	8	12	9	3436	3299	17	5	75.84
	7	491	576	8	4	12	9	3439	3334	24	0	141.70
	8	431	290	11	16	19	11	3571	2119	23	5	113.77
	9	308	458	8	7	12	6	3436	3176	22	7	150.00
	10	531	756	8	7	12	8	3439	3322	22	7	163.78

wave principle that require larger headways between the trains. In such densely operated sections as the Berne-Worblaufen test network, these long route reservation times (Figure 5.25) contribute to eating up the scarce capacity of the considered network area.

Although energy consumption was not included into the objective function of the RCG model solved by cgApP, Table 5.5 shows that rescheduling reduces the energy consumption in the most part of the experiments. This is probably linked with the lower number of stops and unnecessary braking actions than FCFS.

5.6.2.3 No shunting

Given the lack of residual track capacity at station Berne, there is no shunting possibility there, and each train departs from the arrival track of the service previously utilising the rolling stock. The turnaround is done at the station platform during the time passengers alight and board. It follows, that the two services are linked and a minimum turnaround time must be guaranteed. To model this interdependence, the following additional rolling stock circulation constraints are added to model (4.1-4.17)

$$r_{t^{next},0} \leq r_{t,n_t} \qquad \forall t, t^{next} \in \mathcal{T} | t \rightarrow_{RS} t^{next} \tag{5.1}$$

$$\sum_{b \in A_{t,p}} x_b - \sum_{b \in D_{t^{next},p}} x_b \leq 0 \qquad \forall p \in P_{s_{n_t},t}, \forall t, t^{next} \in \mathcal{T} | t \rightarrow_{RS} t^{next} \tag{5.2}$$

$$\sum_{b \in U} x_b + \sum_{b \in V} x_b \leq 1 \qquad \begin{array}{l} \forall (U,V) \in \Omega^{t,t'}_{p,p}, \forall p \in P_{s_{n_t},t}, \\ \forall t, t^{next} \in \mathcal{T} | t \rightarrow_{RS} t^{next} \end{array} \tag{5.3}$$

where $t \rightarrow_{RS} t^{next}$ indicates that at the last station of train t the rolling stock changes to train t^{next}; (5.1) indicates that a train cannot depart within the current horizon if the previous one does not have a path to the common station; (5.2) imposes that the arrival track of the previous train is used for departure; and (5.3) ensures that a train departs only after the one using the same rolling horizon has arrived and the turnaround time has elapsed.

Ten experiments are executed on RBS infrastructure and the timetable in Figure 5.12 with a rescheduling horizon of ten minutes (6:30-6:40 a.m.) and a warm-up phase of ten minutes (6:20-6:30 a.m). The following rolling stock circulations are considered:

- 1056 ↦ 1065
- 2060 ↦ 2065
- 3062 ↦ 3065

Table 5.6: RBS rescheduling with and without Rolling stock circulation constraints.

	without circulation constraints			with circulation constrains	
Scenario	objective value	comp. time [s]	remarks	objective value	comp. time [s]
no delay	-597861	3.991	4602 arrives at platform 3, 4605 departs from platform 1 before 1065	-577984	4.311
1	-647944	5.415	no track inconsistency	-647900	4.9
2	-658436	4.299	4602 arrives at platform 3, 4605 departs from platform 1	-577955	4.226
3	-658313	4.745	4602 arrives at platform 3, 4605 departs from platform 1	-577861	4.602
4	-597861	3.931	4602 arrives at platform 3, 4605 departs from platform 1 before 1065	-627930	4.133

- 4602 $\mapsto$ 4605

In addition, train 4065 is already at station platform at the beginning of the rescheduling horizon. The non-delayed timetable and four delay scenarios generated by OpenTrack software with a mean event delay of 40 seconds (cfr. average delay of regional passenger trains in Section C.2) are rescheduled twice: once with constraints (5.1-5.3) and once without. The following parameters setting are used by the rescheduling algorithm for each experiment:

- the initial restricted master is built with ten time-shifted copies of the planned trajectories with a time-shift of ten seconds.
- the algorithm can execute up to ten master-pricing iterations.
- at each iteration one column per train-section is generated using arc pricing with precomputed blocking time stairways and LP relaxation and added to the master.

Table 5.6 summarises the results. In most cases the lack of rolling stock circulation constraints results in inconsistent platform usage at Berne station. The algorithm tends to move departures to the right platforms to prevent crossings when leaving the station. Adding these constrains does not appear to have a significant effect on the computation time nor on the objective value.

Lessons learned:

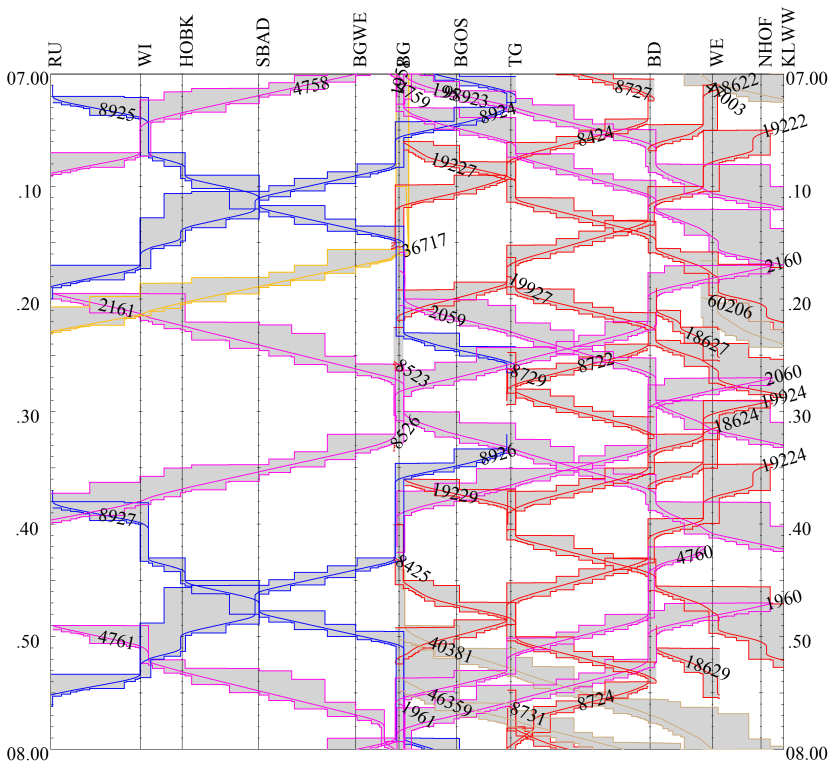
1. *The column generation frameworks with arc pricing problems outperform the monolith for rescheduling with the RCG model in terms of solution quality per computation time;*
2. *cgApP with LP relaxation of the restricted master RCG model has the best performance in terms of overall train delays per computation time;*
3. *Using few time-shifts to build the initial restricted master RCG model decreases the computation time of the column generation algorithms, while small time shifts in the pricing problems using precomputed blocking time stairway patterns improve precision without significantly affecting the computation time;*
4. *Tuning the parameters of cgApP with LP relaxation appropriately, railway traffic rescheduling in a real-world condensation zone with simple track topology and extremely high traffic density takes about 30 seconds, which is compatible with real-time railway operations requirements;*
5. *In many cases, cgApP improves the traffic flows with respect to FCFS by reducing the number of unnecessary stops and braking actions, but this at the price of larger train delays, which suggests the need for modifying the blocking time stairway generator to produce patterns with shorter route reservation times.*

5.7 Column generation approaches for a large real-world network partition around a condensation zone with complex topology

In this section, the different approaches to solve the RCG formulation are tested on a large real-world instance based on the SBB network around station Brugg and the official timetable 2017 (see Figures 5.26-5.26). The aim is to check whether these approaches scale up and, thus, they are suitable for large rescheduling problems. The structure of the experiments is similar to the one presented in the previous section. First, the different algorithms are tested with different parameters to identify the most appropriate one (Section 5.7.1). Then, the best performing algorithm and parameters setting is used on different delay scenarios (Section 5.7.2).

5.7.1 Algorithms and parameters testing

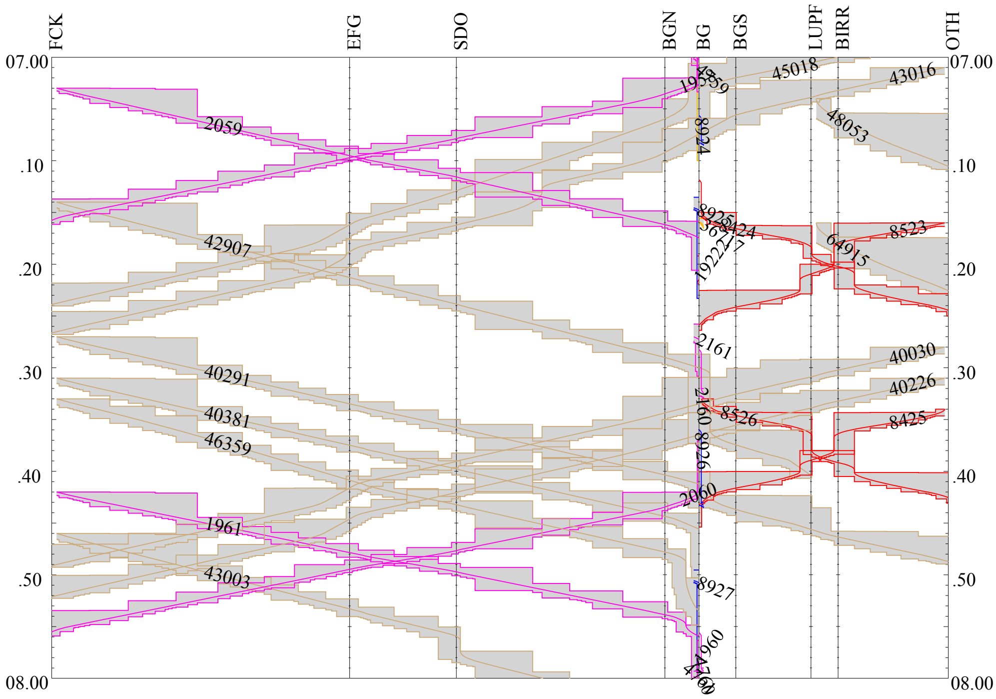
To test the time versus solution quality performance of the monolith and column generation algorithms to rescheduling in a large real-world



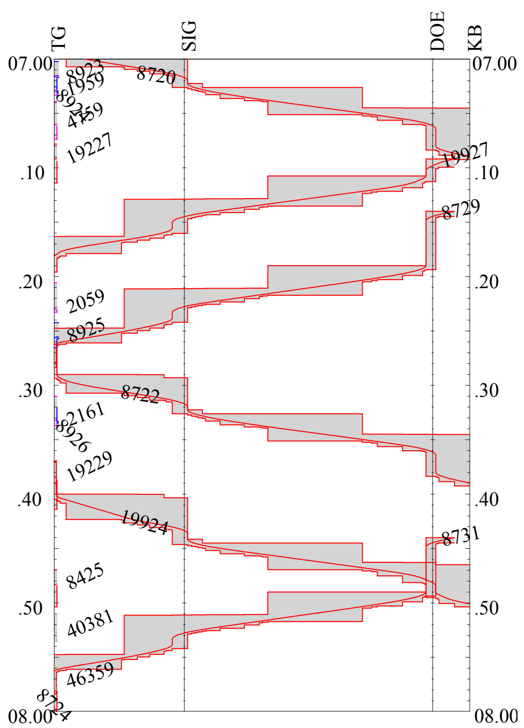
(a) East-West main corridor

Figure 5.26: Official timetable (year 2017) around Brugg (BG)

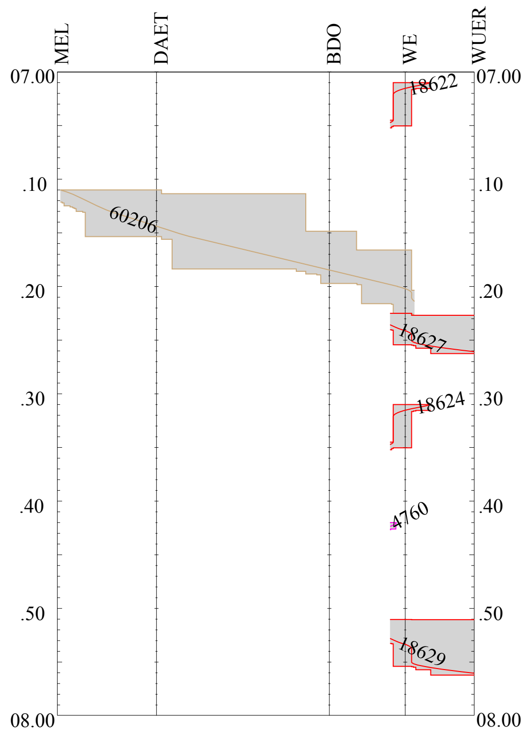
.



(b) North-South main corridor



(c) Turgi-Koblenz side corridor



(d) Branches to Mellingen and Wuerenlos

Figure 5.26: (continued) official timetable (year 2017) around Brugg (BG)

network partition around a condensation zone with complex topology, several simulations are performed on a common delay scenario generated by railway simulation software OpenTrack. All simulations start at time 6:45 a.m. and end at 8:00 a.m., and the rescheduling horizon is fixed from 7:30 a.m. to 8:00 a.m. The following parameters settings are applied to gain an overview of the algorithms' performance in terms of computation time and objective value:

- Blocking time stairway patterns: A database of 7'187 blocking time stairway patterns for all train movements is available. This is generated a-priori using the C++ implementation of the previous experiments of the simulation-based patterns generator described in Section 4.2.3. Three different speed commands can be passed at each signal: maximum track speed, 50 km/h, 80 km/h.

- Monolith: the monolith is tested using six different time shift sizes for the blocking time stairways copies ($3s$, $6s$, $10s$, $15s$, $30s$, $60s$) and two sets of blocking time stairway patterns (rerouting + speed advisory, rerouting only). In addition, a computation time limitation of 10'000 seconds is set.

- Initial restricted master RCG model of column generation: the initial restricted RCG model is built using different numbers of time-shifted copies (10, 20, 30) and six different time shift sizes ($3s$, $6s$, $10s$, $15s$, $30s$, $60s$).

- Pricing problems: both arc and path pricing problems are tested both using precomputed blocking time stairways and constructing new ones, i.e. cgApP, cgApC, cgPpP, and cgPpC. Six different time shift sizes are tested ($3s$, $6s$, $10s$, $15s$, $30s$, $60s$).

- Master-pricing iterations: at most ten master-pricing iterations are allowed for each experiment. Fewer iterations are executed if either no new column can be generated or both the delay of each event is smaller than 60 seconds and all runs have an allocated blocking time stairway. Both LP relaxation settings are tested: solving the LP relaxed restricted master RCG model and passing the dual prices to the pricing problems; and solving the restricted master RCG model as a MILP and passing the slack variables to the pricing problems.

- Terminal costs: 30 minutes, i.e., equals the length of the rescheduling horizon and the timetable period (see discussion in Section 5.6).

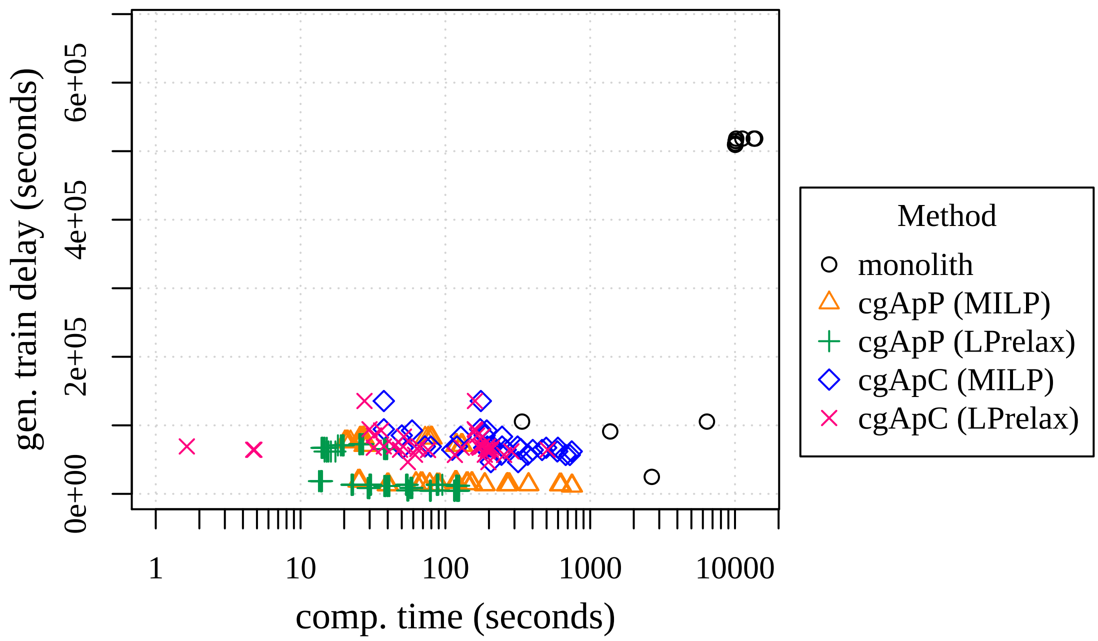


Figure 5.27: Objective value vs. computation time of the monolith and different column generation approaches with arc pricing problems. Please note the logarithmic scala applied to the time (x-axis).

For these tests the same rescheduling program implementation as the previous experiments is used, which runs on the same 64-bit linux server with eighty CPUs Intel(R) Xeon(R) CPU E7-4870 at 2.40GHz and 528 GB memory (RAM).

5.7.1.1 Tuning the parameters of column generation with arc pricing problems

Figure 5.27 shows the performance (i.e., objective value per computation time) of the monolith and column generation approaches with the different arc pricing problems and LP relaxation settings. In many cases, the monolith returns suboptimal objective values due to reaching the maximum computation time allowed (10'000 seconds). As concluded for the previous experiments, smaller time grids result in longer computation times. Again, cgApP with LP relaxed restricted master problem shows the best objective value per computation time. In this case, cgApC cannot achieve equally low values of the objective function as cgApP. This is because only the pricing graphs for cgApC with larger time shifts can be successfully built, while the others fail due to extremely high RAM requirements. Given that infrastructure resources near to stations are very short, the new blocking time stairways constructed using large time shifts

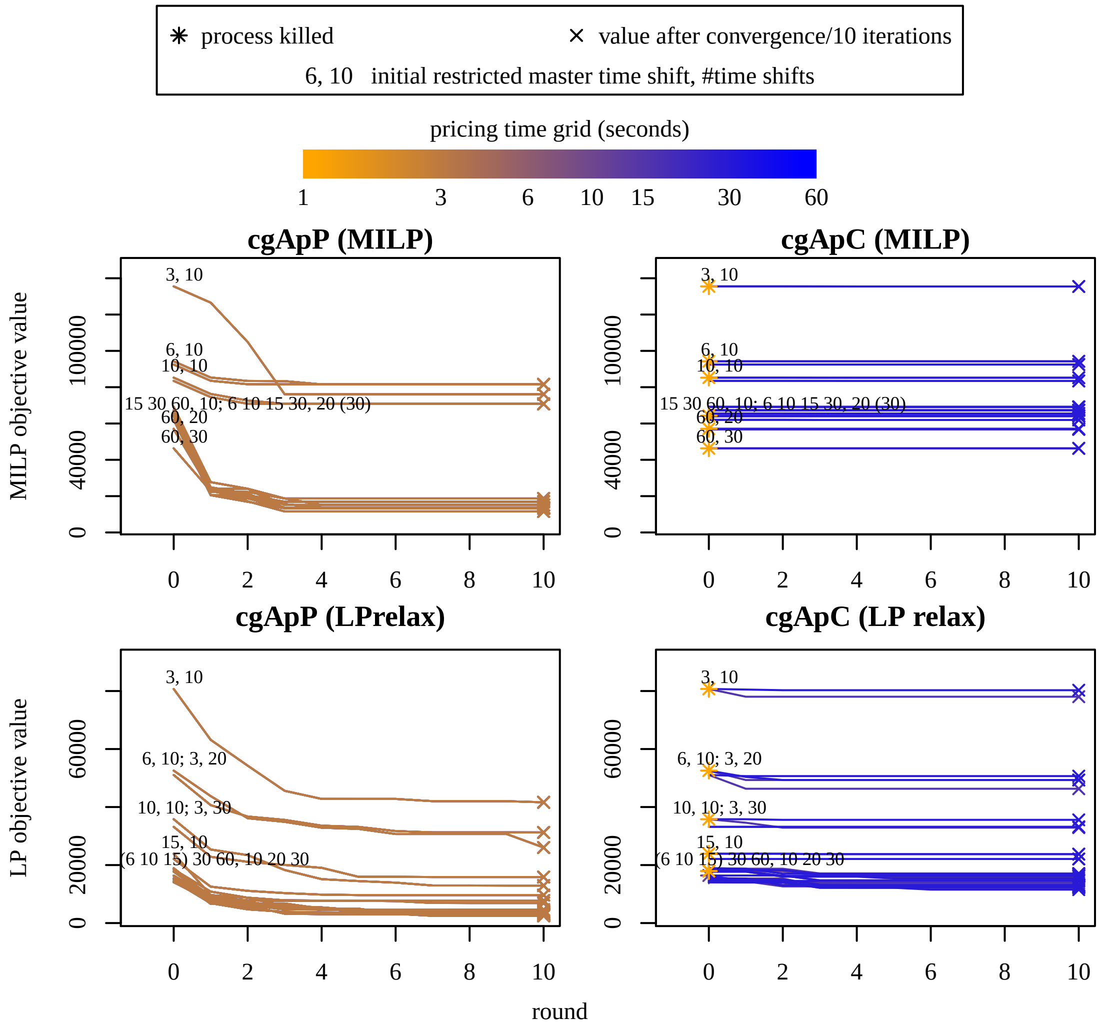


Figure 5.28: MILP and LP convergence of arc pricing approaches.

in the pricing graph have very long travel times, and consequently they cannot reduce the generalised train delay of RCG.

Figure 5.28 shows the convergence of the four column generation frameworks with arc pricing problems (with precomputed blocking time stairways/constructive, MILP/LP relaxation). Again, the LP relaxation of the restricted master RCG model shows better convergence than the original MILP formulation. In addition, in most cases, very little improvement is achieved at later master-pricing iterations.

Figure 5.29 shows the time statistics of the pricing problems. Differently than the previous case study, here the highest average computation time is not imputable to the purely "station sections" (e.g., BG-BGN, BG-BGS) but to the line sections with the most signals (i.e., EFG-FCK, FCK-EFG, BD-TG. TG-BD, with 7-8 routes each), and thus with the highest number of speed profiles combinations. In all cases, the pricing problems with precomputed blocking time stairways require considerably

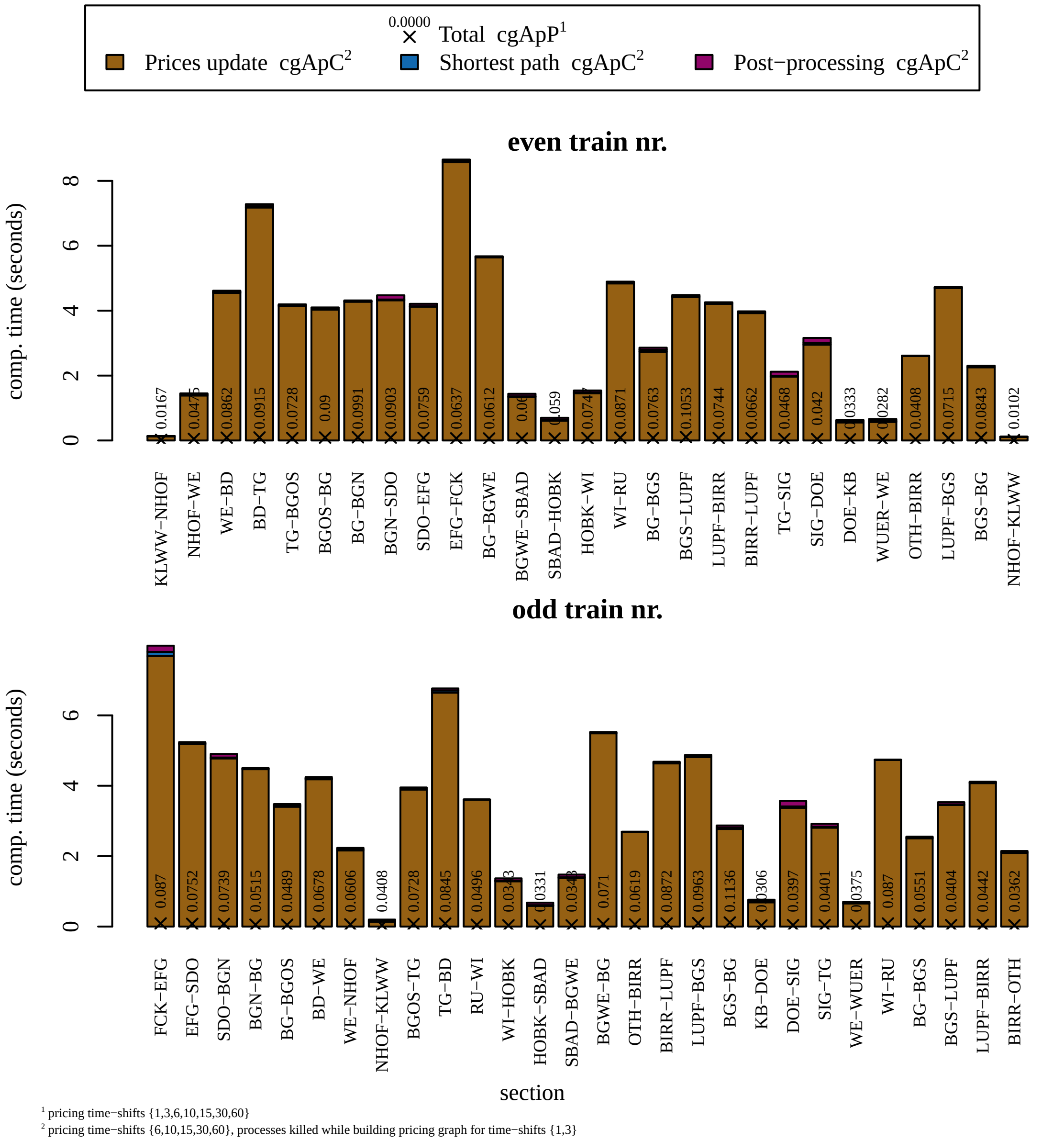


Figure 5.29: Average computation time for generating one new column to be added to the master.

less time than the constructive approach. As in the previous experiments, the price update routine takes the largest share of the overall computation time of constructive pricing problems.

5.7.1.2 K.o. for column generation with path pricing problems

Figure 5.30 shows the computation time versus the objective value of the monolith and cgPpP. Even if the monolith returns suboptimal objective values due to reaching the maximum computation time allowed (10’000 seconds), cgPpP performs even worse. In fact, it needs extremely long times to give solutions that are no better than the monolith’s. Thus, no

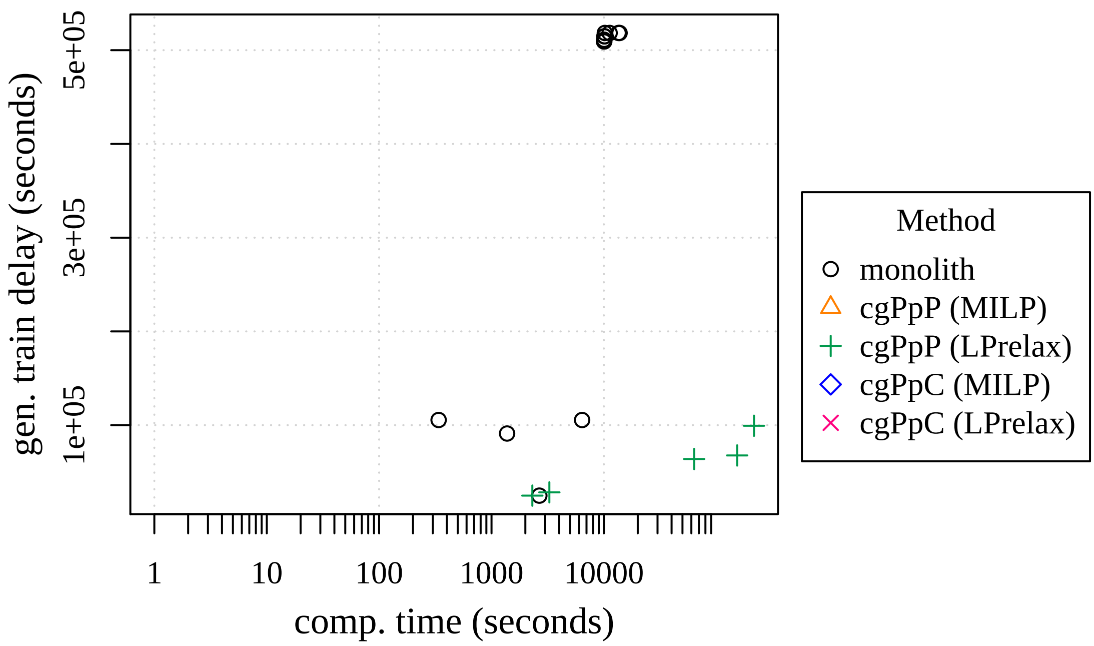


Figure 5.30: Objective value vs. computation time of the monolith and cgPpP. Please note the logarithmic scala applied to the time (x-axis).

further investigation on column generation approaches with path pricing problems is performed on this scenario.

5.7.1.3 Arc vs. path pricing problems and parameters' comparison

Given the unsatisfactory performance of cgPpP, column generation frameworks using arc pricing problems must be preferred. Figure 5.31 shows the correlations of the different algorithm's parameters, RCG model sizes, objectives, and computation times. Again, cgApP has lower computation times and better objective values than cgApC. In addition, the positive correlation of constructive pricing and the size of the time-shift of the pricing problem remind, once again, that the constructive pricing graphs with fine time grids could not be initialised due to excessive RAM requirements. In this case, large time shifts in the initial restricted master should be preferred and the number of such time shifts has a negative correlation with the objective (i.e., the larger the number the smaller the objective value) and a positive correlation with the computation time (i.e., the larger the number the larger the computation time).

Figure 5.32 shows the computation time of the pricing problem by cgApP versus the number of available blocking time stairway candidates. Again, the latter does not have a significant effect on the computation time.

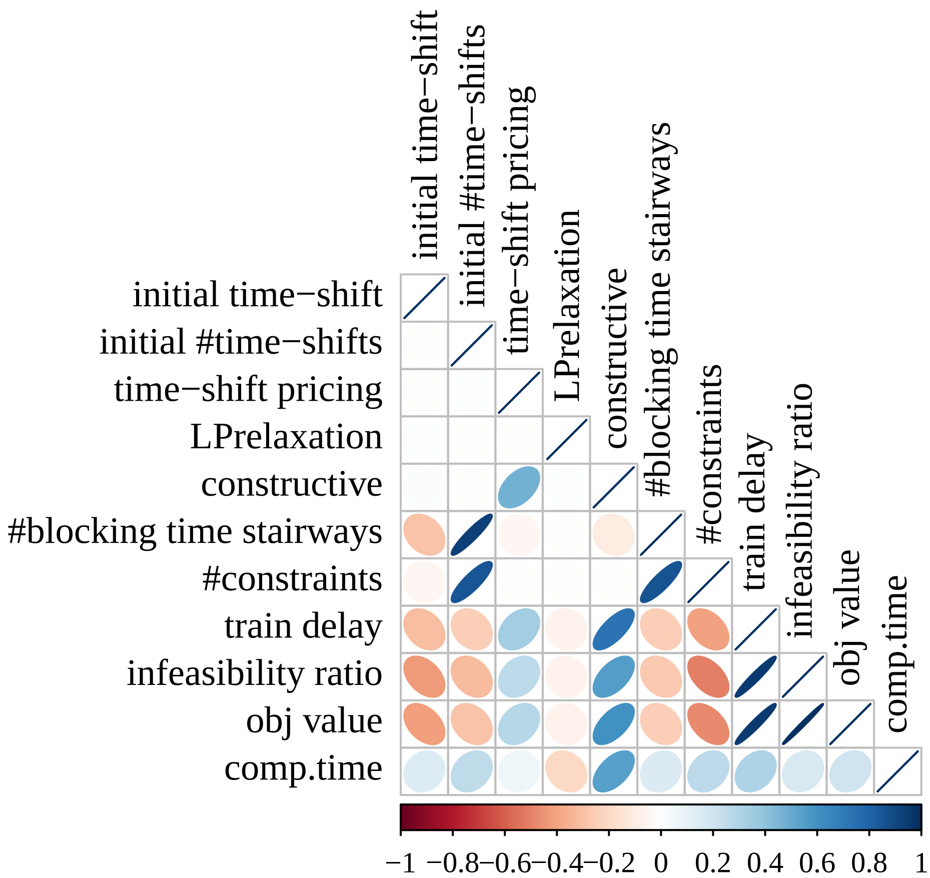


Figure 5.31: Correlations ("arc pricing")

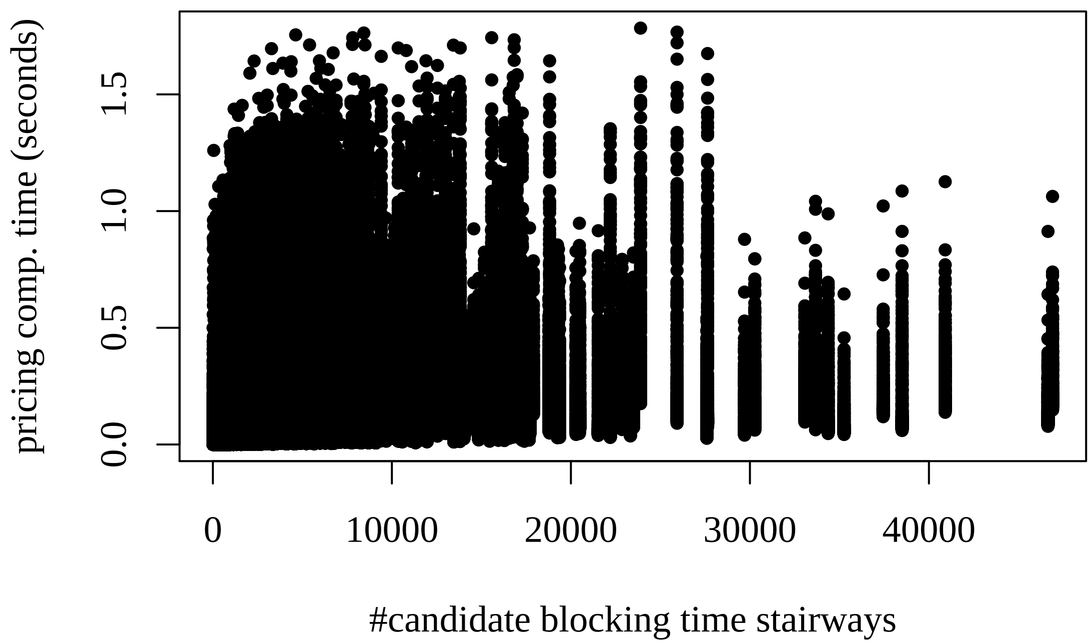


Figure 5.32: Number of candidates vs computation time for arc pricing with precomputed blocking time stairways.

From this analysis, it is possible to conclude that cgApP with LP relaxation of the restricted master problem is the most promising algorithm for rescheduling in the SBB test network. Thus, it is applied to further experiments.

5.7.2 Rescheduling different delay scenarios

The most promising algorithm from the previous analysis is tested on different delay scenarios in this section:

- algorithm = cgApP with LP relaxation (best solution quality per computation time performance, Figure 5.27, and convergence behaviour, Figure 5.28);
- time-shift in initial restricted master = 60 s (negative correlation with objective values, Figure 5.31);
- number of time-shifts in initial restricted master = 10 (shorter computation times, Figure 5.31);
- time-shift in pricing problems = 3 s (higher precision without significant loss of computation time, Figure 5.32);

For these tests the rescheduling program implemented in C++ in Section 5.6.2 is used, and it runs on the same 64-bit linux machine with four CPUs Intel(R) Core(TM) i5 CPU 760 at 2.80GHz and 7.7 GB memory (RAM).

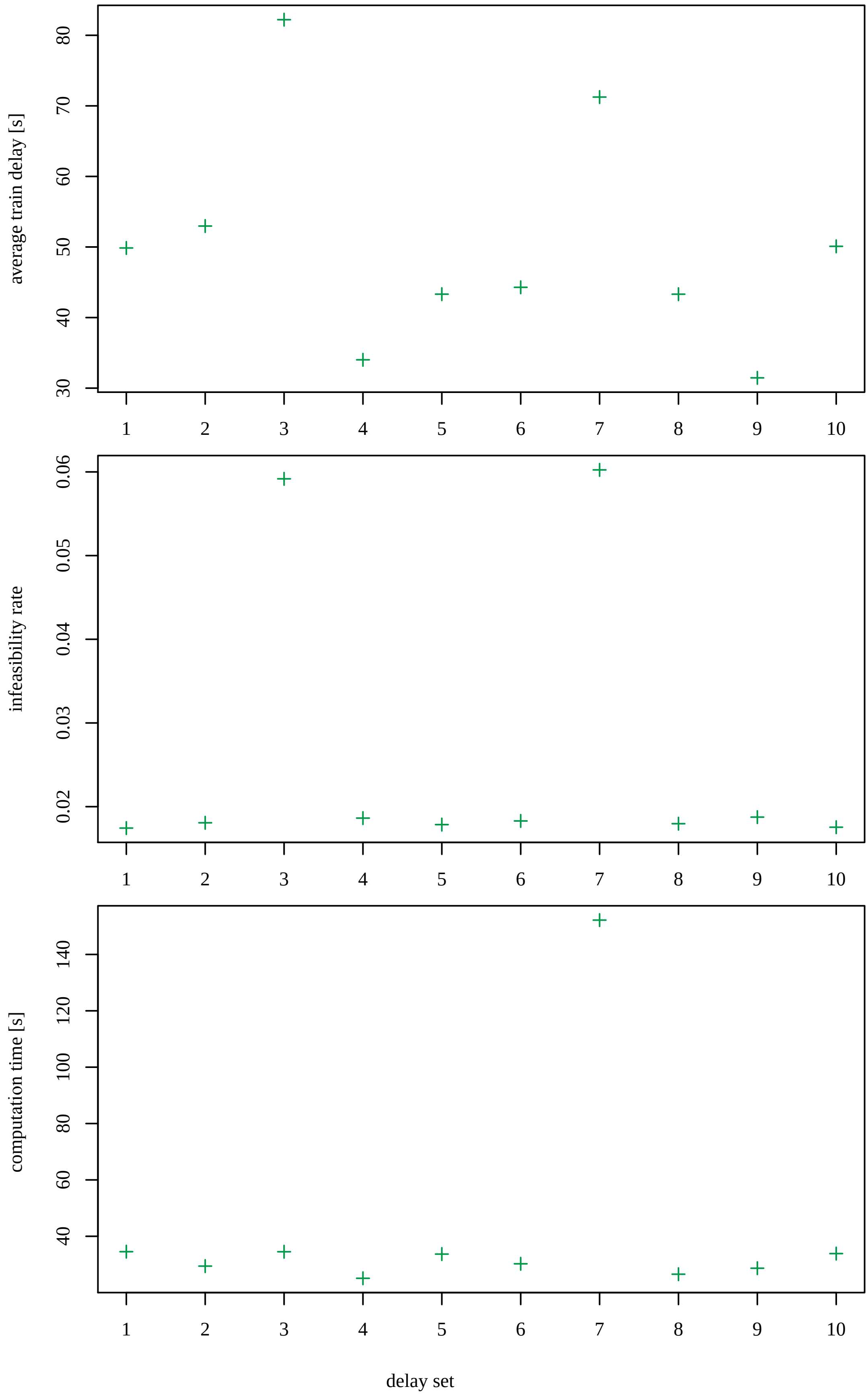
The tests are run with a rescheduling horizon of 30 minutes from 7:20 a.m. to 7:50 a.m. Before the beginning of the rescheduling horizon, each experiment includes a 35-minutes-training phase in which initial delays are added to departure and train passing events according to the Weibull probability distributions for the different train classes calibrated on SBB data and described in Section C.2. Ten experiments, each with a different delay scenario, are executed.

First, the results and computation times of the algorithm are tested in Section 5.7.2.1. Then, the resulting schedules are compared against the standard first come first served policy of the simulator in Section 5.7.2.2.

5.7.2.1 Performance of cgApP

Figure 5.33 shows the statistics of the ten experiments. The average train delay differs considerably among the delay sets. The number of train movements that cannot be scheduled within the rescheduling horizon is

Figure 5.33: Results of rescheduling several delayed scenarios of SBB test case with cgApP algorithm with LP relaxation.



very low (about 1 %) for most experiments, while it is slightly larger (about 6 %) for the two sets with the highest average delays. All computation times are below the aforementioned 180 seconds threshold (Rodriguez, 2007). While in nine of ten experiments cgApP needs about 40 seconds to find a new schedule, in experiment 7 it needs about two and half minutes. Note that experiment 7 has has an high average train delay compared to most other experiments, which suggests the existence of some limit for the size of the disturbances that can be handled by cgApP.

5.7.2.2 Comparison with first come first served

For each of the ten experiments, the new schedule computed by cgApP is passed to the simulator and the resulting delays are computed according to OpenTrack simulation output with a tolerance of 60 seconds. The results are compared with the in-built FCFS policy of the simulator (please refer Section 7.3 in OpenTrack Railway Technology Ltd., 2015, for further details). Table 5.7 summarises the results.

Similar to the RBS test case (Section 5.6.2), for most of the delay sets, the rescheduling algorithm reduces the number of stops and the number of braking actions between operation points. However, these improvements of the traffic flows are associated with some increases of the overall train delays. As before, this phenomenon is due partly to the set of precomputed blocking time stairway patterns used to initialise the restricted master RCG problem and define the pricing problems for cgApP. In fact, some of these pattern have extremely long route reservation times (cfr. Figure 5.25). Nevertheless, the low infeasibility rates shown in Figure 5.33 suggest that this issue is much less critical than in the RBS test case. Differently from the previous case, cgApP increases the energy consumption with respect to cgApP. A closer look at the results reveals that cgApP schedules more stops at operation points (OPs) than the original timetable (see, e.g., Figure 5.34).

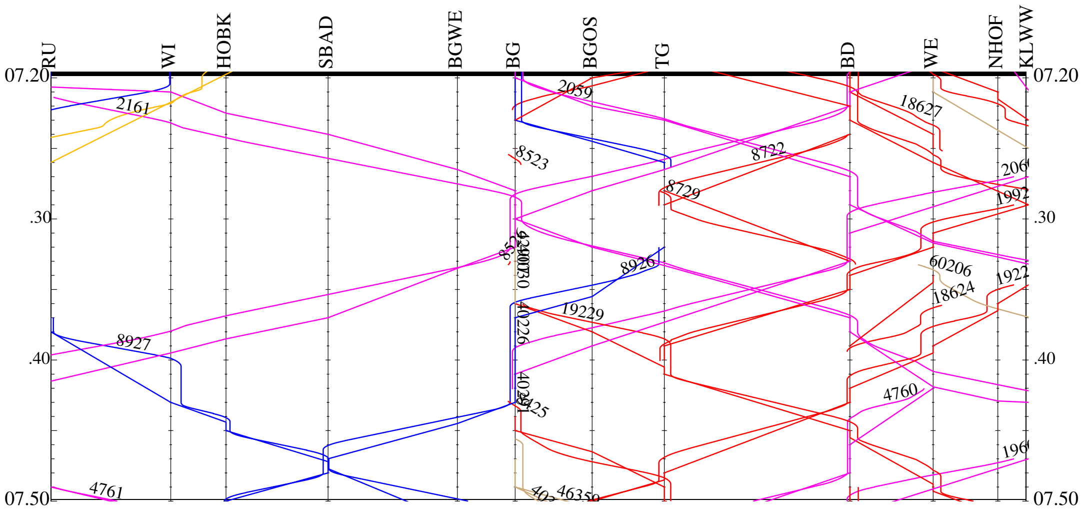
Thus, it should be concluded that FCFS results in lower train delays and stops at operation points than cgApP, but cgApP results in fewer unscheduled stops and braking actions between the operation points. This is due to two factors. First, all patterns in the used database are generated from green wave trajectories that require trains to be separated by quite large headways. Given that FCFS does not have to satisfy this requirement, it allows trains to travel closer to each other and encounter a number of yellow and red signals between consecutive operation points, which can reduce the overall train delay. Second, the previous section has shown that the convergence of cgApP becomes quite slow after some iterations.

Table 5.7: Comparison of rescheduling several delayed scenarios of SBB test case with cgApP algorithm with LP relaxation against FCFS.

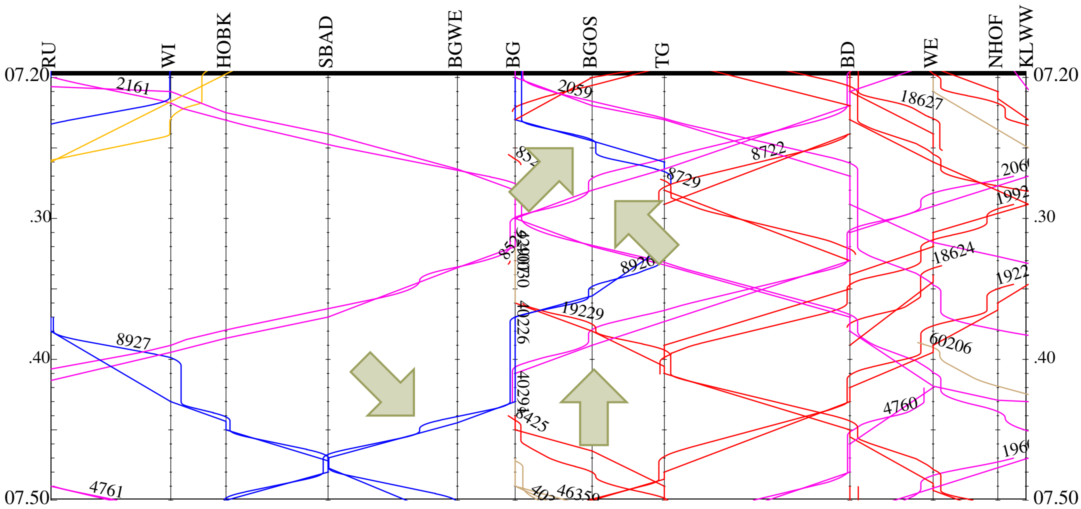
Horizon	delay set	Train delay $[s]$		#stops at signals		#braking for route		Energy $[MJ]$		#stops at OPs		#reroutings	#retimings	computation
		FCFS	cgApP	FCFS	cgApP	FCFS	cgApP	FCFS	cgApP	FCFS	cgApP	cgApP	cgApP	time $[s]$
7:20-7:50 a.m.	1	3315	4104	10	12	20	17	58309	62184	92	124	109	40	34.54
	2	3201	3791	9	4	24	12	59142	62439	92	129	99	9	29.42
	3	3395	4100	8	2	24	12	58771	61750	92	128	99	24	34.52
	4	1608	2265	9	8	20	17	58200	60268	92	117	99	13	25.10
	5	1999	2949	10	9	22	15	59045	62246	92	125	106	17	33.69
	6	2720	3484	10	8	18	16	58382	61585	92	119	97	14	30.24
	7	3374	4283	9	7	16	13	57653	57317	92	122	102	13	152.18
	8	3411	4183	6	8	11	7	58160	60337	92	118	101	13	26.55
	9	1014	1627	9	10	22	18	58519	60615	92	116	97	14	28.67
	10	3469	4056	17	13	38	27	60914	61975	92	118	102	4	33.86

As a consequence, given the huge number of decision alternatives and constraints of this large test case, cgApP might not change a consistent schedule at an operation point for many iterations, even if it causes an additional train stop.

Figure 5.34: cgApP schedules additional stops at operation points. Example from delay scenario 1.



(a) FCFS



(b) cgApP

Lessons learned:

1. *The column generation frameworks with arc pricing problems result lower computation times per solution quality than the monolith in a large test network around a condensation zone with complex topology;*

2. *cgApP with LP relaxation of the restricted master RCG model has the best performance in terms of overall train delays versus computation time, while column generation frameworks with path pricing problems perform very poorly;*

3. *Using few time-shifts to build the initial restricted master RCG model decreases the computation time of the algorithms, while using large time-shifts in the initial restricted master and small time-shifts in the pricing problems improves precision without significantly affecting the computation time;*

4. *After tuning the parameters of cgApP with LP relaxation appropriately, railway traffic rescheduling in a large test network around a condensation zone with extremely heterogeneous traffic takes about 40 seconds, which is compatible with real-time railway traffic management requirements;*

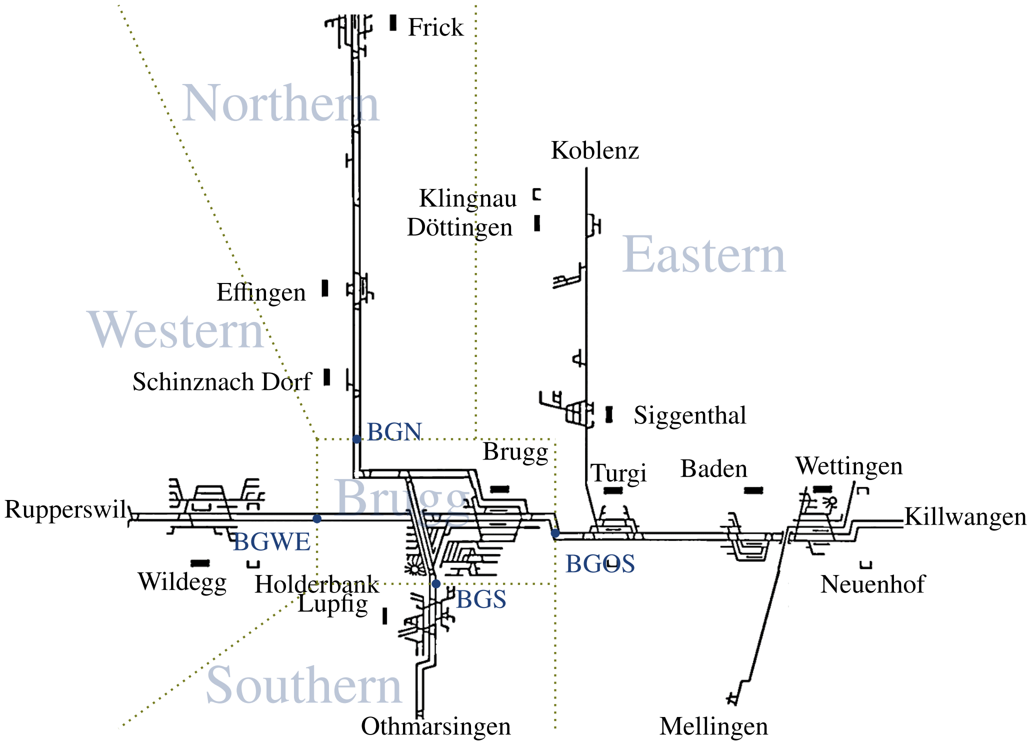
5. *The cgApP algorithm with LP relaxation can improve the traffic flows by reducing the number of unnecessary stops and braking actions between operation points with respect to FCFS in a large network partition around a condensation zone with complex topology, but this at the price of larger train delays and possibly additional stops at operation points.*

5.8 Coordinated rescheduling over multiple areas

In this section, the partitioning and coordination approaches described in Sections 4.1 and 4.5 are validated using the SBB test network (see Sections 5.2.3 and 5.7). The infrastructure is partitioned into five zones (Figure 5.35):

- Brugg: the condensation zone with complex topology centred on station Brugg (BG) and spanning to the operation points Brugg West

Figure 5.35: SBB test network partitioned



(BGWE), Brugg North (BGN), Brugg East (BGOS), and Brugg South (BGS).

- Eastern: the condensation zone with simple topology including stations Turgi (TG), Baden (BD) and Wettingen (WE) as well as stops Neuenhof (NHOF), Siggenthal (SIG), Doetting (DOE), Baden Dorf (BDO) and Daetwil (DAE). The operation point Brugg East (BGOS) serves as portal to zone Brugg.

- Western: the compensation zone with simple topology including station Wildegg (WI) and stops Holderbank (HOBK) and Schinznach Bad (SBAD). The operation point Brugg West (BGWE) serves as portal to zone Brugg.

- Northern: the compensation zone with simple topology including station Effingen (EFG) and stop Schinznach Dorf (SDO). The operation point Brugg North (BGN) serves as portal to zone Brugg.

- Southern: the compensation zone with simple topology including stops Lupfig (LUPF) and Birr (BIRR). The operation point Brugg South (BGS) serves as portal to zone Brugg.

The operation points BGOS, BGS, BGWE, and BGN are no-wait portals (Definition 4.5). The decomposition is a no-wait network partitioning according to Definition 4.6.

The non-hierarchical coordination approaches from Sections 4.5.2 and 4.5.3 are tested on the first five delay scenarios generated in Section 5.7.2 using the Weibull distributions calibrated in Appendix C.2. Each scenario is solved several times. The first time, all zones are rescheduled independently with no coordination. The other times, rescheduling of adjacent zones is coordinated using the non-hierarchical approach from Section 4.5 with the following settings:

- subgradient: the subradient method described in Section 4.5.2 is used to coordinate decisions at portals;
- hierarchic: the negotiation-based approach described in Section 4.5.3 is used, where the decisions taken by condensation zones have weights equals to one and the ones taken by compensation zones equals to zero;
- uniform: the negotiation-based approach described in Section 4.5.3 is used, where all decisions in both zones have weights equals 0.5;
- direction: the negotiation-based approach described in Section 4.5.3 is used, where the decisions of each zones concerning outbound movements have weights equals to one and inbound movements equals to zero.

The cgApP framework for RCG model with the parameters setting from the previous experiments is used for local rescheduling in each area:

- algorithm = cgApP with LP relaxation;
- time-shift in initial restricted master = 60 s;
- number of time-shifts in initial restricted master = 10;
- time-shift in pricing problems = 3 s;
- number of master-pricing iterations = 10;

In each experiment, all local rescheduling problems are solved once without any coordination objective. Then, each zone informs its neighbours about the values it has obtained at the common portal and adapts its own objective function according to the values received from the neighbours. The updated RCG formulations are then used as initial restricted master problems for the next execution of cgApP. These steps are repeated nine times or until consistent times and speeds are assigned at all portals. For this experiment, passing times at portals of adjacent rescheduling areas are consistent if their difference is less than 90 seconds[5].

For these tests the rescheduling program implemented in C++ in Section 5.6.2 is adapted to include the coordination loop and objectives. The executable runs on the same 64-bit linux machine with four CPUs Intel(R) Core(TM) i5 CPU 760 at 2.80GHz and 7.7 GB memory (RAM).

Table 5.8 shows the statistics of the new schedules, while Figure 5.36 gives an insight into the consistency at portals for one of the delay scenarios. All results of the consistency at the portals for the five delay scenarios are shown in Appendix D.

Given that all local rescheduling algorithms aim at minimising the deviation from the published timetable, one might expect that coordination is naturally achieved implicitly. Nevertheless, Figure 5.36 shows that this assumption is not always justified: some grey segments are quite long, which indicates that the two adjacent rescheduling algorithms have assigned very different passing times at the common portal. Moreover, the vertical overlapping of the grey segments linked to trains travelling in the same direction (i.e., even/odd train numbers) indicates that the algorithms have assigned different orders to the trains at the portal as well. Thus, in general, a coordination strategy is needed.

Coordinated rescheduling using the subgradient method does not generally improve the consistency at the portals, and in many cases the solution degenerates considerably during the nine iterations. In particular, some train runs are deferred after the rescheduling horizon, probably, due to the lack of a perfect match at the portals. Negotiation-based coordination algorithms with direction-based weights return very similar results as the non-coordinated approach in all experiments. The coordinated rescheduling algorithms using uniform or hierarchic weights reduce the overall train delay and the inconsistencies at the portals. In the first three scenarios, hierarchic weights perform best in terms of consistency of decisions at portals, while uniform ones give better results in the remaining two. Figure 5.36 shows how the negotiation-based coordination strategy with hierarchic weights solves the most part of the inconsistencies at the portals for the first delay scenario.

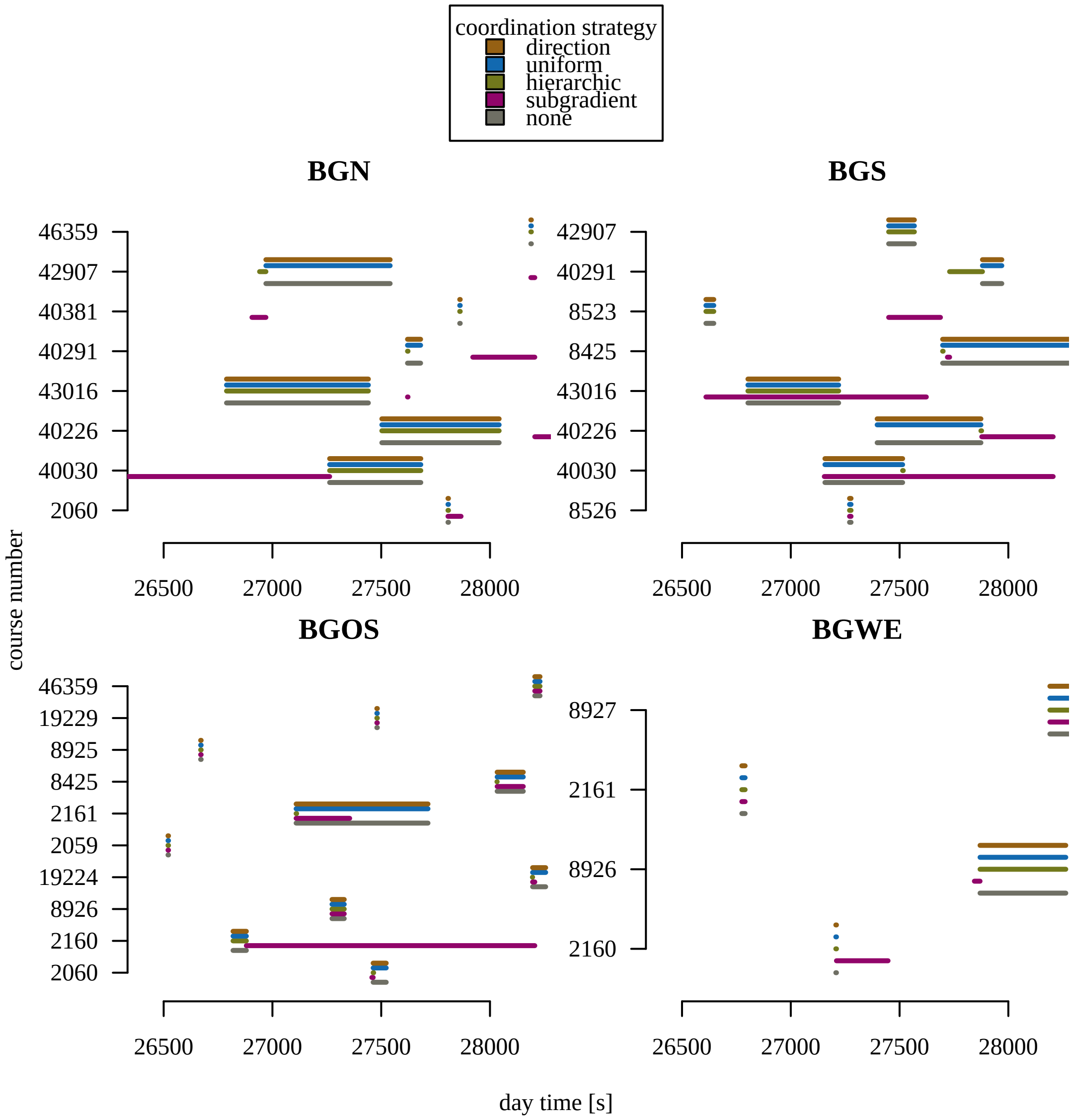
The subgradient method needs extremely long computation times compared to the negotiation-based methods. In addition, the timetable statistics are not comparable to the ones of the other methods due to a

[5]In this case, train orders will be consistent too, because this is the minimum technical headway between consecutive runs.

Table 5.8: Results of coordinated rescheduling of several delayed scenarios of SBB test case.

delay set	strategy	train delay [s]	#stops at signals	#braking for route	#stops at OPs	#retimings	#reroutings	computation time [s]
1	FCFS	3315	10	20	92	0	0	0
	cgApP centr.	4104	12	17	124	109	40	34.54
	cgApP distr.							
	-none	5097	12	24	107	94	17	15.46
	-subgradient	3987	14	20	90	93	14	21957.52
	-hierarchic	3936	11	19	103	93	15	91.63
	-uniform	5128	12	26	104	95	12	90.51
	-direction	5128	12	26	107	95	17	90.79
2	FCFS	3201	9	24	92	0	0	0
	cgApP centr.	3791	4	12	129	99	9	29.42
	cgApP distr.							
	-none	3592	10	22	101	93	21	12.36
	-subgradient	1375	9	16	94	86	31	24819.4
	-hierarchic	3006	12	27	92	87	22	97.92
	-uniform	3726	11	21	98	92	12	94.55
	-direction	3592	11	22	100	93	15	94.94
3	FCFS	3395	8	24	92	0	0	0
	cgApP centr.	4100	2	12	128	99	24	34.52
	cgApP distr.							
	-none	4915	5	18	104	92	9	12.83
	-subgradient	1846	16	22	83	88	14	19194.46
	-hierarchic	2115	14	20	86	85	15	86.65
	-uniform	5107	6	17	104	93	16	90.55
	-direction	4915	5	18	104	92	9	93.74
4	FCFS	1608	9	20	92	0	0	0
	cgApP centr.	2265	8	17	117	99	13	25.10
	cgApP distr.							
	-none	2544	11	21	95	83	8	9.61
	-subgradient	1975	14	24	91	80	17	940.10
	-hierarchic	2062	12	24	99	80	15	100.34
	-uniform	2109	14	24	89	73	25	89.34
	-direction	2544	11	21	95	83	8	98.55
5	FCFS	1999	10	22	92	0	0	0
	cgApP centr.	2949	9	15	125	106	17	33.69
	cgApP distr.							
	-none	3481	12	24	105	92	9	12.49
	-subgradient	2348	11	16	90	81	12	23504.07
	-hierarchic	2277	12	18	102	88	16	138.52
	-uniform	3204	23	29	88	80	20	135.86
	-direction	3481	12	24	105	92	9	140.17

Figure 5.36: Intervals between arrival at and departure from portals. Different colors correspond to different coordination strategies. Delay set 1.



large number of deferred runs. Note that the coordinated rescheduling algorithm with hierarchic weights results in lower overall delays than the ”centralised” resolution of the same scenario with cgApP (cfr. Table 5.7). This can be explained in two different ways. First, cgApP can better explore the solution space of smaller problems. This means that it can converge to a better objective value in the ten master-pricing iterations allowed. Second, loosening the consistency at operation points BGN, BGS, BGWE, BGOS and rewarding perfect matches of arrival and departure times partially solves the problem of the additional stops at operation points highlighted in the previous section.

Table 5.9: Computation times of local rescheduling algorithms (in seconds).

Weights	delay set	Zone				
		Southern	Northern	Eastern	Western	Brugg
none	1	7.80	10.75	15.46	0.04	11.63
	2	9.19	9.76	12.36	0.04	10.12
	3	8.02	10.03	12.83	0.04	9.72
	4	4	9.61	9.18	0.04	6.72
	5	7.5	11.75	12.49	0.03	9.85
subgradient	1	327.74	2786.10	455.37	0.14	21957.52
	2	542.65	47.33	120.42	0.09	24819.40
	3	551.7	68.01	183.66	0.12	19194.46
	4	940.1	0.23	100.13	0.15	545.91
	5	137.58	124.28	102.54	0.14	23504.07
hierarchic	1	20.16	58.51	91.63	0.03	34.85
	2	31.27	14.29	97.92	0.03	30.37
	3	30.25	12.88	86.65	0.03	31.17
	4	30.08	12.37	100.34	0.04	31.02
	5	31.11	60.1	138.52	0.04	37.34
uniform	1	36.27	59.11	90.51	0.04	35.73
	2	32.62	12.70	94.55	0.04	31.48
	3	35.73	12.97	90.55	0.04	33.59
	4	25.6	12.02	89.34	0.05	26.15
	5	36.06	59.94	135.86	0.04	38.24
direction	1	37.58	59.03	90.80	0.05	36.21
	2	31.81	12.13	94.94	0.036	31.00
	3	35.59	12.49	93.74	0.038	33.66
	4	29.91	10.46	98.55	0.04	30.78
	5	34.32	58.11	140.17	0.03	35.47

Table 5.9 reports the overall computation times of each local rescheduling algorithm. Note that the experiments with the coordination loop run cgApP ten times in each zone while the non coordinated ones only once. As mentioned in the previous paragraph, the subgradient method results in extremely long computation times. Using the subgradient method, the local rescheduling algorithm in Brugg, which is the zone with most portals to other zones, takes the most time. In negotiation-based approaches, local rescheduling in the largest condensation zone (Eastern) requires much longer times than in the other zones. This suggests that reducing the

complexity of the largest local rescheduling problem should contribute to reduction of the overall computation times.

Lessons learned:

1. *Using the published timetable as common reference does not suffice to ensure consistent rescheduling decisions at portals between adjacent zones;*

2. *The negotiation-based approach with uniform and hierarchic weights described in Section 4.5 improves consistency at portals with respect to distributed rescheduling and also produces smaller train delays than the "centralised" application of cgApP;*

3. *The overall computation time of the negotiation-based distributed rescheduling algorithm is given by the complexity of the largest local rescheduling problem, which should be small to ensure short computation times.*

Chapter 6

Discussion

This chapter brings together the different pieces of the thesis to answer the research hypotheses and to support the synthesis that will follow in the final chapter. The aim of the present thesis is to investigate how algorithmic methods can support real-time traffic management when small disturbances occur. In Chapter 4, several methods (models and solution algorithms) have been developed, and in Chapter 5, these methods have been experimentally tested using simulations. The next section summarises the methods, while the experimental results are discussed in Section 6.2. Section 6.3 presents an assessment of the methods in light of the experimental results. In Section 6.4, it is described how these methods can be integrated into the current control loops of railway operations. Finally, the research hypotheses are answered in Section 6.5.

6.1 Summary of methods

Four methodological enhancements of the railway traffic rescheduling approach by Fuchsberger (2012) have been proposed in Chapter 4:

1. The original RCG model had been developed for complex central station areas with high traffic density and many routing options. That model focused on a single operation point (a large station) and modelled the other operation points (usually stops and junctions) in the control area implicitly as part of the blocking time stairways linked to the decision variables. In condensation zones with simpler topologies and compensation zones, the definition of that "focal operation point" is not straightforward and several operation points have to be modelled explicitly to enable the necessary degrees of freedom for conflict resolution. Thus, an **RCG formulation extension for rescheduling over multiple operation points** has been

developed in Section 4.2.1. The main novelty is the application of the concepts for ”compatibility at the focal operation point” of the original formulation to all operation points and their extension to account for speed differences for non-stopping trains[1]. The new formulation enables the blocking time stairways to be cut at the operation points and to be recombined to enlarge the set of available rescheduling decision alternatives. In addition, given that railway traffic rescheduling objectives can differ considerably among different types of zones, several objective functions considering operators’ and customers’ preferences have been proposed in Section 4.2.2 and integrated into the model.

2. In the complex and busy central station areas that were the focus of the original RCG formulation, the high number of switches opens many routing possibilities but limits the train speeds. This considerably reduces the number and the heterogeneity of speed profiles applicable by the trains. In addition, given that train movements are highly interdependent in these areas, it has been often assumed that each train should travel as fast as possible to minimise the interferences with potentially many other trains (see Section 4.1.2 for further details). Consequently, the original RCG approach focused on selecting the most appropriate rerouting and retiming decisions without considering speed advisory. Simpler topologies are characterised by limited routing alternatives but usually higher speed limits. Thus, the number of speed profiles applicable by trains is much larger than in complex topologies. There, conflicts often occur when a fast train catches up with a slower one that precedes it. It follows that speed advisory must be considered for train rescheduling in simpler topologies. Practically, this means that the RCG model must be built using the blocking time stairways corresponding to different speed profiles. The accurate computation of such blocking time stairways must consider both train dynamics and the interlocking system. To support this data preparation phase a **simulation-based blocking time stairways generator** has been proposed in Section 4.2.3. The main idea is to use a railway simulation software application to reproduce each train movement with different routing and speed parameters with the necessary precision.

3. The inclusion of multiple operation points and speed advisory decisions into the RCG model increases the problem complexity, which

[1]The original formulation implicitly assumed that all trains stopped at the large central station in its focus.

results in high computation times (see Section 4.3). To reduce the computation times such that the algorithms are compatible with real-world applications, **a column generation framework with different pricing problems** has been introduced in Section 4.4. The main idea is to build a restricted RCG model with a small number of decision alternatives and iteratively add further alternatives. This reduces the size of the problem and thus enables shorter computation times. New alternatives to be added to RCG are selected by pricing problems. Two different pricing strategies have been proposed: one using a database of blocking time stairway patterns generated with the simulation based approach of Section 4.2.3 (cgApP, Section 4.4.4.1); and one constructing new blocking time stairways from a shortest path computation on a time-expanded infrastructure graph (cgApC, Section 4.4.4.3). In addition, so-called ”path versions” of the RCG model that delegate consistency at operation points to the pricing problems have been proposed (cgPpP, Section 4.4.4.2, and cgPpC, Section 4.4.4.4).

4. The original RCG approach focused on a single (complex condensation) zone and assumed that the zone was surrounded by areas where enough time reserves are available to recover from delays (cfr. the network decomposition by Laube et al., 2007; Caimi, 2009). Thus, no coordination was needed for rescheduling in adjacent zones. However, to match the problem size with the performance of the available compute server, it might be necessary to partition a railway traffic rescheduling problem into several local rescheduling problems, which necessarily have to find consistent solutions at the common boundaries. To coordinate rescheduling in adjacent zones a **non-hierarchical coordination approach** has been proposed in Section 4.5. The main idea is to add some terms enhancing coordination into the objective function of each local rescheduling problem and iteratively correct them with pieces of information exchanged with the neighbours until rescheduling decisions at the common boundaries are compatible. These terms are approximations of the Lagrangian duals of the constraints of the RCG model linked to the railway operations at portals.

6.2 Discussion of the experimental results

The four enhancements summarised in the previous section have been empirically tested using simulations in Chapter 5. This section discusses the experimental results and highlights the strengths and the weaknesses of the proposed models and algorithms.

Validity of the RCG formulation extension for rescheduling over multiple operation points. The tests summarised in Sections 5.3-5.4 have demonstrated the validity of the RCG formulation extension for rescheduling over several operation points that has been introduced in Section 4.2.1. The tests in Section 5.3 have been performed in a compensation zone built on a small toy network and focused on the operator's perspective (i.e., on the minimisation of train delays and energy consumption), while the tests in Section 5.4 have been performed in a condensation zone built on the same network and focused on the customer's perspective, (i.e., on the minimisation of passengers' delays and inconvenience). Feasible new schedules adapted to the given traffic situation have been found in all experiments. The lower objective values with respect to the non-optimised delayed schedule demonstrated that the RCG formulation extension proposed in Section 4.2.1 is valid for application in both compensation and condensation zones, and it is flexible to support different rescheduling objectives.

Need for terminal costs and fast algorithms to solve the RCG formulation. Section 5.5 presented the first experiments performed on a partition of a real-world railway network using the official timetable. A multi-objective approach concurrently solved several copies of the RCG model with a commercial MIP-solver. Each copy was augmented by some ϵ-constraints to differentiate the solutions according to the multi-criteria approach presented in Section 4.2.2. The framework appeared to be successful in creating a large variety of solutions. Unfortunately, much of the variety was obtained by deferring trains after the considered rescheduling horizon, which highlighted the need for terminal costs to prevent it. In addition, the computation times of these experiments were around six minutes and, consequently, not satisfactory. Thus, the experiments have been repeated with terminal costs and a shorter rescheduling horizon. Although the new computation times were compatible with real-time applications (i.e., under the 180 seconds limit usually applied by researchers in this field and initially proposed by Rodriguez, 2007), the new results were much less diversified. In particular, the delay statistics of the new schedules were very similar to each other. This was due to the lack of degrees of freedom caused by the coarse time grid used for generating the time-shifted copies of the blocking time stairway patterns linked to the decision variables of the RCG model with respect to the short rescheduling horizon. Nevertheless, the energy consumption statistics showed that the multi-criteria approach is able to create a set of different solutions, if enough diversity is provided by the blocking time stairways linked with the decision variables of the RCG model. It followed that a finer time-shift was needed. However, this was not viable without an improvement of the solution algorithm because of the high computation times required by the MIP-solver. These results highlight the need for terminal costs to prevent that trains fall out of the rescheduling horizon and for faster algorithms to solve the RCG formulation within the short time available during real-time traffic management.

The column generation framework with arc pricing problems using precomputed blocking time stairway patterns and with LP relaxation of the restricted master problem has computation times compatible with real-time applications. The numerical experiments in Sections 5.6-5.7 have shown that the proposed column generation framework with arc pricing problems gives better objective values per computation time than the straightforward application of a commercial solver to the full RCG formulations (Figures 5.13 and 5.27). Using pricing problems based on precomputed blocking time stairway patterns and the LP relaxation of the restricted master RCG problem, so-called cgApP algorithm, gave the best results and the computation times were usually well below the mentioned 180 seconds limit. The good performance of cgApP with respect to the other algorithms could be linked to two reasons. First, building the space-time routing graph for the constructive pricing problems with fine time grids often exceeded the available amount of RAM of the compute server, thus requiring the corresponding processes to be killed without returning any solution. The consequence was that new blocking time stairways had to be built on large time grids that exceeded the necessary travel time of many infrastructure resources. Thus, the new blocking time stairways had excessively long travel times that made them unappealing for the next iterations of the restricted master problem resolution. Second, after solving the LP relaxation of the restricted master RCG problem the resulting dual prices could be used by the pricing problems for selecting new columns. This improved the convergence with respect to the slack variables of the MILP formulation (see Figures 5.14 and 5.28). Thus, cgApP with LP relaxation of the restricted master problem is the most promising of the algorithms investigated by the current thesis, as it enables the computation times of a new schedule to be compatible with real-time applications even for those problems that could not be solved within more than two hours by the straightforward application of a commercial MIP-solver to the full RCG formulation.

The column generation framework with arc pricing problems using precomputed blocking time stairway patterns and with LP relaxation of the restricted master problem can improve traffic flows in condensation zones with simple track topology and highly dense and homogeneous traffic with respect to a first come first served policy. However, the new schedule might conflict with the objective of dynamic capacity improvement, if the used database of blocking time stairway patterns is generated according to the green wave principle. The experiments in Sections 5.6.2 showed that cgApP with LP relaxation can reduce the number of unplanned stops and braking actions on different delay scenarios and rescheduling horizons with respect to a first come first served policy (FCFS). Fewer unplanned stops and braking actions were also reflected into lower energy consumptions. Unfortunately, the improvements in the traffic flows were usually obtained through increased headways between the trains and resulted into larger overall delays than FCFS. Larger headways are not necessarily bad as they contribute, in most cases, to reduce unplanned stops and braking actions. In addition, given that red and yellow signals are usually displayed when a train travels relatively close to the preceding one, some railways are focusing on reducing the number of red and yellow signals encountered by trains in order to achieve higher safety in case of drivers' errors (see, e.g., the report by van den Top, 2010). However, in condensation zones with extremely dense schedules, such as the RBS test case, where trains travel relatively slowly (max. 80 km/h), large headways can oversaturate the network and result in large delays. The experiments highlighted that this problem is accentuated in shorter rescheduling horizons and by the long route reservation times of the blocking time stairway patterns used to build the initial restricted master RCG model and the pricing problems. The improved traffic flows with respect to FCFS suggest that the approach is valid for rescheduling in condensation zones with highly dense and homogeneous traffic, in the sense that it prevents a number of conflicts which would result in red or yellow signals. Nevertheless, these results indicate that the classical green wave principle, i.e., each train encounters only green signals on its path, is in contradiction with the objective of dynamic capacity improvement explained in Section 1.1. In fact, the long route reservation times required by the green wave principle result in large headways that absorb a significant part of the track capacity. As a consequence, train runs acquire additional delays to keep the required headways from the preceding trains, and some runs are deferred due to the lack of capacity in the current rescheduling horizon. Thus, future research should address the problem of how to generate blocking time stairway patterns for the RCG model with reduced route reservation times that meet both the safety requirements and the capacity consumption limits of a timetable, i.e., a kind of *yellow wave* principle.

The column generation framework with arc pricing problems using precomputed blocking time stairway patterns and with LP relaxation of the restricted master problem can improve the traffic flow with respect to a first come first served policy in a large network partition around a complex condensation zone. However, due to excessively strict modelling assumptions, an increased number of stops at operation points increases both the overall train delay and energy consumption. The experiments in Section 5.7.2 showed empirically that cgApP with LP relaxation of the restricted master problem can improve the traffic flow with respect to FCFS by preventing a number of stops and braking actions between the operation points. Similarly to the previous case, these traffic flow improvements were achieved at the costs of higher overall train delays than FCFS. This was due to the fact that all patterns in the used database were generated according to the green wave principle, which required trains to be separated by large headways. Given that FCFS did not have to satisfy this requirement, it allowed trains to travel closer to each other and encounter a number of yellow and red signals. Nevertheless, this was much less critical than in the previous experiments and only very few runs had to be deferred to the next horizon. The improved traffic flows with respect to FCFS suggest that the approach is valid for rescheduling in a large network partitions around a complex condensation zone, in the sense that it prevents resource allocation conflicts. However, in this case, no energy savings were observed due to cgApP inserting many additional stops at operation points. These additional stops result from the modelling assumptions. Stops at operation points enable the already mentioned long route reservation times required by the green wave principle to be limited. In fact, if a train stops at an operation point, it can reserve the route for the continuation short before departing, while a train travelling through the operation point without stopping must reserve it long in advance. In addition, given that a stopping train needs some time to accelerate, it might use blocking time stairways with shorter reservation times until it reaches higher travel speeds. Moreover, at operation points, consistency constraints are easily satisfied by trajectories including a stop, because they have more flexibility with respect to arrival and departure times. As a consequence, given the large number of possible combinations of decisions for this scenario, the algorithm might take many iterations before finding a schedule with fewer stops at operation points. The results obtained with the coordination framework summarised in the next paragraph show that selecting those operation points where the timetable does not plan any train to stop as portals for a network partitioning partially overcomes the problem.

The non-hierarchical coordination framework with uniform or hierarchical negotiation weights improves consistency at common portals of adjacent zones. The overall computation time is determined by the most complex local rescheduling problem. Decomposing a large rescheduling problem and coordinating the solution of the partitions with a non-hierarchical approach can even produce smaller overall delays than the centralised resolution through cgApP. The experiments in Section 5.8 showed empirically that the network partitioning and negotiation-based coordination approach with uniform and hierarchic weights presented in Section 4.5 was successful in reducing the inconsistency of passing times at portals in the schedules produced by adjacent local rescheduling algorithms with respect to independent resolution of the local problems. Given the reduced sizes of the local rescheduling problems, they could be solved several times to improve consistency without exceeding the 180 seconds threshold. Surprisingly, the solution obtained by combining the local schedules resulted in lower overall train delays than the one of the centralised problem solved with cgApP algorithm in the previous experiments. The reason for this phenomenon is twofold. First, the column generation frameworks can most probably investigate the solution space of the smaller local rescheduling problems sufficiently well within the number of master-pricing iterations allowed at each round. Second, loosening the consistency at the operation points used as portals and rewarding exact matches of departure and arrival times reduced the problem of the additional stops at operation points highlighted in the previous paragraphs. The experiments highlighted that the overall computation time is strictly linked to the size of the most complex local rescheduling problem. It follows that the negotiation approach with hierarchic weights is valid for coordinating local rescheduling algorithms in adjacent zones. Nevertheless, further research is required for the definition of a network partitioning strategy that considers the complexity of the infrastructure, the traffic density and the performance of the available compute server and for a formal definition of the negotiation weights.

6.3 Assessment of methods

As a summary of the previous discussion, this section highlights the strengths and weaknesses of the proposed methods in light of the experimental results. The RCG model extension for several operation points finds feasible conflict-free schedules. On the one hand, simulations on a toy network with low to medium traffic density have shown that the proposed approach can considerably reduce the overall train delay as well as improve energy efficiency and reduce customer delays. On the other hand, simulations on real-world networks with medium to high traffic density have shown that the algorithm significantly reduces the number of stops and braking actions at signals on lines between operation points with respect to a first come first served policy. Nevertheless, this usually resulted in larger overall train delays due to some additional stops scheduled at operation points and to the large headways required by the blocking time stairways linked to the decision variables. Thus, it can be concluded that the green wave principle used to generate these blocking time stairways conflicts with the objective of dynamic capacity improvement. Indeed, the long route reservation times required such that trains encounter only green signals eat away significant amounts of capacity.

Tests on different railway networks and traffic densities have shown empirically that the proposed column generation framework with arc pricing problems and LP relaxation of the restricted master, so-called cgApP, enables a considerable reduction of computation time without significant loss of solution quality with respect to the straightforward application of a commercial MIP-solver to the full RCG formulation. The proposed algorithm could return the new adapted schedule almost always much before the 180 seconds time limit prescribed by Rodriguez (2007) for real-time traffic management applications. In particular, it was able to solve large real-world traffic rescheduling problems that the straightforward application of a commercial software could not even solve in more than two hours. Thus, it can be concluded that the cgApP is suitable for real-world real-time railway traffic rescheduling in both zones with simple track topology and large network partitions around complex condensation zones.

Tests on a large subset of the SBB network have shown empirically that partitioning the problem basing on the track topology and traffic and coordinating the local rescheduling problems with the negotiation-based approach provides comparable schedules with the centralised resolution of a unique huge problem using the column generation framework mentioned above. Negotiation weights that consider the traffic density in each partition and uniform ones perform much better than direction-based ones in improving the consistency of decisions at portals. This non-hierarchical coordination approach considerably improved the consistency of decisions at common portals with respect to the independent resolution of the local rescheduling problems. In addition, it enabled the coordination of two adjacent condensation areas. It follows that the requirement that condensation areas are separated by compensation areas (Laube et al., 2007) can be dropped. This enables large railway traffic rescheduling problems to be decomposed according to the performance of the available compute server such that the overall computation time of a new schedule meets real-time railway operations requirements. Thus, it can be concluded that the negotiation-based approach with uniform and hierarchic weights is valid to coordinate rescheduling in adjacent zones.

The numerical experiments have highlighted that the proposed simulation-based approach to generate blocking time stairway patterns should be further developed before application to large networks and highly dense traffic because it generates excessively long route reservation times, which translate into large headways between trains and, possibly, into larger delays than a first come first serve policy at capacity critical points. Therefore, improved methods to generate these databases are needed, in order to enable reduced headways and, as a consequence, increase the capacity available to schedule more train runs.

6.4 Integration into the railway traffic control loop and links to the customer information system

Figure 6.1 shows the integration of current research's results into railway traffic management. The algorithms of this thesis support conflict detection and resolution in the traffic control loop (green rectangle in the figure). Their initialization lies on data recorded by the supervision system and suitably elaborated (left part of the figure). The return values concerning train routing, ordering and timing decisions at stations, junctions, signals and route reservation points are passed either to a dispatcher for confirmation or directly to the CTC system (upper row right to green rectangle). The return values concerning train speeds and passing times at intermediate points are sent to driver advisory systems (DAS) for implementation within the train operation control loop (lower row).

The data from the supervision system consist of train positions and infrastructure states. Train positioning depends on train-to-infrastructure

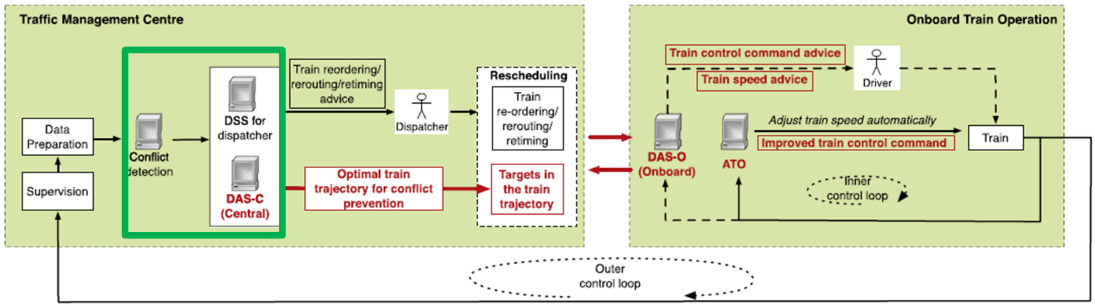


Figure 6.1: Integration of current research (green rectangle) in railway control (adapted from Rao et al., 2016).

communication. With the current signalling system, train positions are detected at track circuits and no information about train speed and acceleration is available. With ETCS Level 2 and newer systems trains can continuously report their position, speed and acceleration to the traffic control centres. The algorithms proposed in this thesis can work with both degrees of precision, as they assume the availability of reliable predictions of the train movements happening in the considered time horizons. These predictions are considered during the construction of the RCG model for selecting only blocking time stairways that are actually applicable to the current traffic situation. The state of the infrastructure resources is given by CTC and infrastructure management systems. The interlocking system records route locks and releases as well as trains occupying tracks. For safety reasons, it is usually not allowed to change a route that has already been reserved for a train run. This is taken into account when constructing the RCG model by subselecting only the blocking time stairways including the routes already set in the corresponding section, if any. The same concept applies to arc and path pricing problems with pre-computed blocking time stairways, while in constructive arc and path pricing problems the source nodes of the time-expanded infrastructure graph are linked to the predicted exit point and time of the already reserved route. To account for trains occupying an operation point's track at the beginning of the rescheduling horizon, a dummy variable with fixed value one is added to constraints (4.10) and (4.12) of the corresponding track and train. Infrastructure managing systems record out-of-order resources. Such cases are modelled in RCG by changing the right-hand side of the corresponding constraint (4.9) or (4.10) from one to zero.

Given the data recorded by the supervision system, the main task of the data preparation phase is the prediction of train movements that will happen in the rescheduling horizon. Prediction is needed to account for the evolution of the system during the elaboration of the new schedule and to compensate for possible time lags from data recording, missing data, and large gaps due to discontinuous communication between infrastructure and train (see, e.g., Weidmann and Sinner, 2016, for detailed analysis of data flows at SBB). Note that prediction precision decreases over long horizons (Kecman et al., 2015). Thus, the size of rescheduling horizon should be adapted to prediction possibilities. For the case studies in Chapter 5, a primitive prediction system based on the scheduled trajectory and the last information of train position was implemented. However, this can be replaced by any more advanced prediction system.

Once the RCG model is solved, several pieces of information must be transmitted to the interlocking plants and to trains. The routes of the chosen blocking time stairways for all trains have to be allowed by the interlocking system. This implies that routes have to be stored when generating blocking time stairways either with simulation based approaches or with a constructive pricing approach. In case of a platform change of a passenger train, the customer information system must be notified either directly by the rescheduling algorithm or indirectly (e.g., through the interlocking system as in case of CUS, see Section 2.2.2). In addition, interlocking plants need information about the ordering on shared infrastructure, which can be extrapolated by the resulting allocation schemas of all resources managed by a single plant. It is enough to read each schema starting from the earliest time and fix the relative ordering of the trains according to the order of the corresponding allocations. New departure (and arrival) times at operation points are sent to trains and to the customer information system. These correspond to the arrival and departure time of the chosen blocking time stairways between consecutive operation points. Decisions about connections must be sent to customer information systems. Speed advices are communicated using either signals or onboard equipment. Speed targets have to be generated together with the blocking time stairways either using the simulation-based approach from Section 4.2.3 or when assessing the feasibility in the constructive pricing approach from Sections 4.4.4.3-4.4.4.4 (the v_i variables).

6.5 Verification of the research hypotheses

Before the next chapter summarises the main findings and answers the underlying research question of the present thesis, this section recalls the hypotheses stated in Section 1.2 and briefly explains how they have been verified.

H1 The relevant functional requirements of traffic management can be described in a way, which makes uniquely determinable whether a rescheduling method satisfies them or not. The analysis of the general features of the railway system and the review of the most relevant literature on railway traffic management systems has determined the functional requirements for railway operations. The list of requirements has been validated thanks to discussions with practitioners from SBB and is summarised in Table 2.1.

H2 One or more mathematical models can represent the functional requirements from H1. An extensive literature review enabled the identification of a number of mathematical models of railway operations (Section 3.1). The evaluation with respect to the functional requirements (H1) has returned two models that satisfy all of them: the alternative graph and the resource conflict graph formulations. The results of the evaluation are summarised in Table 3.1.

H3 The parameters for these models can be obtained with either the current safety system based on lineside signalling and tracking or the European Train Control System (ETCS) Level 2. To be applied for real time rescheduling both models mentioned above have to be initialised with the current train positions and availability of infrastructure resources. In both cases, the model can be initialised with both discrete and continuous time-position information. Thus, the models are compatible with both systems.

H4 Traffic management consists of subtasks, whose boundary conditions and goals can be written using the models of H2. Then, the subtasks of traffic management are optimisation problems. From a literature review and discussions with practitioners it has been possible to identify the subtasks of traffic management (part of H1). In general, two main tasks of traffic management could be identified: the first is to find the best local schedule adapted to the current traffic situation for each zone of the railway network, the second is to coordinate rescheduling of adjacent zones. The evaluation of H2 highlighted that the RCG model can represent the different subtasks of local rescheduling (rerouting, retiming, reordering, speed advisory and wait-depart passenger connections) as well as the boundary conditions of railway operations. In addition, during this study, it was empirically proven to be very flexible with respect to the objective. A literature review revealed that the task of coordinating rescheduling of adjacent zones can be performed by either non-hierarchical or multi-level approaches. The current thesis proposed an iterative framework for non-hierarchical coordination, where additional terms enhancing convergence towards commonly agreed decisions at portals are included into the objective functions of the local RCG models (Figure 4.13). This way, coordination with neighbours is an objective of each local rescheduling problem.

H5 For each optimisation problem of H4, there is an algorithmic method that solves it within the short time available during real-time operations without the solution quality failing under a minimal definable threshold. The numerical experiments in Sections 5.6-5.7 have shown that the column generation framework with arc pricing problems using precomputed blocking time stairways developed in Section 4.4 is able to produce good quality solutions in short computation times when applied to real-world scenarios. With an appropriate pricing algorithm and suitable tuned parameters, most of the experiments performed on the RBS test network needed about 25 seconds to be solved and most of the ones performed on the SBB test network needed about 40 seconds, which is much less than the conventional 180 seconds limit usually applied in real-time railway traffic management evaluations[2]. Using the statistics resulting from a first come first served policy as minimum threshold, these experiments have highlighted that the mentioned column generation framework improves the traffic flows in different types of zones with respect to a first come first served policy, as it reduces the number of unplanned stops and braking actions on tracks between operation points. Nevertheless, besides the positive effects on traffic flows, some of the experiments exhibited larger overall train delays, which could be explained by the long headways between trains required by the green wave principle that has been used to generate the alternative trajectories. The experiments in Section 5.8 have shown that the non-hierarchical approach to coordinate rescheduling in adjacent zones improves the consistency of the schedules at the common portals and that the global schedules are comparable with the ones generated by the centralised rescheduling problem. These experiments also highlighted that the computation times to define the new global schedule were below the aforementioned 180 seconds but depend on the size of the largest local rescheduling problem considered.

[2]This value has been originally introduced by Rodriguez (2007) referring to a joint project with the French infrastructure manager SNCF, and is currently applied by many researchers all around the world.

H6 There is a combination of the algorithms of H5 that returns either a feasible new schedule adapted to the current traffic situation or the proof that no such schedule exists within the modelling assumptions. Any algorithm based on the RCG model can be used for the local rescheduling problems within the non-hierarchical coordination approach proposed in Section 4.5. In the experiments in Section 5.8 the column generation approach with arc pricing using precomputed blocking time stairways and the LP relaxation of the restricted master RCG model was used. These experiments showed empirically that the insertion of the proposed local rescheduling algorithm into the negotiation-based coordination approach with uniform and hierarchical weights could improve the traffic flow and the consistency at the portals between zones in short computation times. Unfortunately, not all inconsistent decisions at portals could be corrected within the number of iterations executed. In addition, given that the algorithms are based on heuristics, it is not possible to get a proof that no better solution exists.

H7 The most common rescheduling measures (e.g., retiming departures, changing the train orders, giving up passenger connections, etc.) can be handled within the algorithms of H6. The decision variables of the RCG model selected in H4 represent exactly these rescheduling measures. Thus, by solving the RCG model, the algorithms of H6 actually choose the best rescheduling measures to apply to the given traffic situation.

H8 The application of the optimal rescheduling measure can be automated. As explained in Section 6.4 the control variables of the models of H4, and thus the return values of the algorithm of H6, are a combination of routing and timetable decisions that can be passed to the interlocking system and speed advisory information that can be transmitted to trains.

H9 Consistency and timeliness of passenger information can be improved thanks to the algorithms of H6. Given that customer information in Switzerland is a combination of timetable information and real-time feeds from the interlocking system (see Section 2.2.2), the algorithm of H6 is expected to improve consistency and timeliness of passenger information because it provides a new schedule adapted to the current traffic situation that can be followed more precisely than the original one computed months in advance. This reduces the differences between the two sources of information used by the customer information system, thus reducing the cases of inconsistency. Moreover, the new schedule provides up-to-date information before the interlocking system sets the required routes and tracks the actual train movements, thus improving timeliness.

H10 It is possible to define a category of ”small disturbances” and the proposed approach provides feasible schedules for every scenario rising from this category. Small disturbances were defined by Cacchiani et al. (2014) as those perturbations of railway traffic that can be resolved by modifying the timetable and not the crew and vehicle schedules. This category includes train delays and failures of single infrastructure elements. The proposed algorithms have been tested in Chapter 5 with different delay scenarios generated with parameters that were estimated using SBB data of actual train runs. In all experiments, the proposed algorithms returned new feasible schedules that often outperformed a first come first served policy in terms of unplanned stops and unnecessary braking actions. The algorithms have not been tested in case of infrastructural failures, which should be done in future research.

Chapter 7

Synthesis

7.1 Summary of findings

The aim of this thesis is the identification of models and algorithms to support railway traffic rescheduling and, as a consequence, to improve customer information in case of small perturbations of railway operations. More precisely, it focuses on the research question:

> How can algorithmic real-time rescheduling procedures support the resolution of small disturbances in railway operations in condensation zones and inbound lines, in order to make traffic management automatable and, as a consequence, improve consistency and timeliness of passenger information?

Two tasks of traffic management that needed algorithmic support have been identified and addressed:

- local railway traffic rescheduling in different types of zones (in terms of track topology complexity and traffic density);
- coordination of railway traffic rescheduling in adjacent zones.

Methods based on Operations Research have been proposed for the algorithmic support of each task, and they have been validated by simulations using real-world railway traffic data. In addition, a possible integration of these methods into the current information systems of SBB has been suggested.

The method to support local railway traffic rescheduling in different types of zones is based on the mathematical representation of the problem as a resource conflict graph model, as proposed by Caimi (2009) and Fuchsberger (2012). Caimi (2009) proposed a multilevel timetabling approach, where the microscopic timetables of different areas were computed separately. In condensation areas, many routing options and a unique speed profile per route and train type were considered. In compensation areas, different speed profiles were considered. Fuchsberger (2012) adapted the microscopic timetabling approach for condensation zones to real-time rescheduling in central station areas with highly complex track topology. Considered by his approach were routing options, alternative times for departures at the tracks of the station and at the boundaries of the zone, and wait-depart decisions for connections. The current thesis enhanced this approach to make it more suitable for rescheduling in different types of zones. A model augmentation extended the departure time and wait-depart decisions to several operation points in a single zone; while a simulation-based approach enabled the integration of speed profile alternatives with very realistic train dynamics and ensured that the new adapted schedules are conform to the green wave principle. Thanks to a column generation framework, the enhanced mathematical representation could be solved in short times in a number of numerical experiments based on real-world railway data. Thus, the proposed method for local railway traffic rescheduling is suitable for application to all types of track topologies and timetables.

This approach differentiates itself from other local rescheduling approached in several ways. First, the method proposed in this thesis relies on an extremely detailed representation of infrastructure and train dynamics. Very few approaches to local rescheduling model railway infrastructure at track-circuit level (one example is, e.g., Pellegrini et al., 2014) or consider acceleration and braking capabilities of rolling stock in their estimations of train dynamics (one example is, e.g., D'Ariano and Albrecht, 2006). Thanks to the problem representation as a resource conflict graph model and the simulation-based framework for speed profile alternatives, the proposed approach has both these features, which enables very precise conflict detection and compatibility with current interlocking systems. Second, many researchers based their approaches on continuous-time formulations of the problem (e.g., D'Ariano et al., 2007b, 2008, 2014; Corman et al., 2012a; Lu et al., 2004; Pellegrini et al., 2014; Törnquist and Persson, 2007); while the resource conflict graph model is a time-indexed formulation. Both types of formulations have known weaknesses (Lamorgese et al., 2016). Continuous-time formulations require Big-M constraints for modelling resource allocation conflicts, which cause slow convergence of solution methods. Time-indexed formulations require a huge number of variables (and possibly of constraints) to model all timing, routing, and speed options, which involve an even larger number of possible combinations that have to be investigated to find a solution. During the past decades, ad-hoc solution methods have been developed for continuous-time formulation to obviate the slow convergence caused by Big-M constraints (see, e.g., Corman et al., 2010b). This thesis proposes an ad-hoc solution method for the time-indexed formulation based on the RCG to obviate the model size issue. The column generation framework reduces the number of decision variables in the resource conflict graph formulation, which improves its solution time and its competitiveness.

The method to support the coordination of railway traffic rescheduling in adjacent zones is based on the Lagrangian relaxation of the consistency constraints at the portals between the zones. Differently from approaches featuring microscopic rescheduling only in busy station areas (e.g., Dollevoet et al., 2014), this non-hierarchical method enables the trajectories of trains to be adjusted before they reach a condensation area. Thus, it enlarges the decision space to resolve conflicts to adjacent zones that might have capacity reserves. The proposed coordination framework can be applied to almost any network partitioning, which invalidates the requirement that condensation zones are separated by compensation zones (Laube et al., 2007). As a consequence, it supports the definition of zones where local rescheduling is manageable by the available compute server. That is, the proposed non-hierarchical coordination method improves the scalability with respecto to the multi-level timetabling approach by Caimi (2009), which makes it eligible to support real-time railway traffic rescheduling. Still, the global problem characteristics are retained by the distributed approach, which enables the optimisation of objectives such as train delays, passenger inconvenience and energy efficiency.

All pieces of information required to initialise these methods are currently available in state-of-the-art traffic management systems. Similarly, the results can be passed to the interlocking, signalling, driver advisory and customer information systems for automated implementation or after validation by a dispatcher. As a consequence, consistency and timeliness of customer information can be improved. In fact, given that the new schedules are adapted to the current traffic situation, there will be fewer inconsistencies among the different sources that feed the customer information system.

7.2 Future research needs

The current thesis leaves a number of open questions for further research. First of all, given that the size of a local rescheduling problem has a large influence on the computation time of the algorithms, it is necessary to define a process for network partitioning into zones that are tractable by the local rescheduling algorithms in the short time available during real-time railway operations. The partitioning should consider the complexity of the track topology, the traffic density, and the performance of the available compute server.

Second, it is important to know whether the proposed coordination approach is applicable to larger networks with many condensation and compensation zones. If not, the limits of the current approach should be specified, and a global coordination strategy should be developed. In addition, given the good results on the consistency of decisions at portals of both uniform and hierarchic coordination weights observed in some experiments, further investigations on how to determine the negotiation weights of adjacent rescheduling zones should be made.

Third, future research should address the problem of how to generate blocking time stairway patterns for the RCG model with reduced route reservation times. In fact, the computational experiments have highlighted that the green wave principle conflicts with the objective of dynamic capacity improvement as it requires large headways between trains, which eat away significant amount of track capacity and cause additional delays at critical points. The generation should meet both the safety requirements and the capacity consumption limits of a timetable. Nevertheless, capacity losses should be minimised.

Last, but not least, new ideas for improving the constructive pricing problems should be tested. The approach proposed in this thesis often exceeded the available RAM of the compute server. However, it should not be necessary to build the entire pricing graph for each train run to find good new columns and a smart algorithm could extremely improve this approach.

Abbreviations

ACP Arc Configuration Problem

AG Alternative Graph

API Application Programming Interface

APP Arc Packing Problem

ATO Automatic Train Operation

BB Branch and Bound

cgApC column generation with Arc pricing Constructing new blocking time stairways

cgApP column generation with Arc pricing using Precomputed blocking time stairways

cgPpC column generation with Path pricing Constructing new blocking time stairways

cgPpP column generation with Path pricing using Precomputed blocking time stairways

CTC centralised train control, system to manage railway interlockings

DAS Driver Advisory System

DSS Decision Support System

EBL Eisenbahnbetriebslabor, Railway Operations Laboratory

ERTMS European Rail Traffic Management System

ESP Event Scheduling Problem

ETCS European Train Control System

FCFS First Come First Served

FP Flexible Path

MILP Mixed-Integer Linear Programming

PCP Path Configuration Problem

PESP Periodic Event Scheduling Problem

PPP Path Packing Problem

RBS Regionalverkehr Bern-Solothurn, commuter network

RCG Resource Conflict Graph

REF-SRR REFormulated Simultaneous train Rerouting and Rescheduling

RR Route-lock-Route-release formulation

RTCG Resource Tree Conflict Graph

SAT Boolean satisfiability problems

SBB Schweizerische Bundesbahnen, Swiss Federal Railways

TCG Tree Conflict Graph

ZLD Zuglenkdaten, train routing data

Bibliography

Albrecht, A., P. Howlett, P. Pudney, X. Vu and P. Zhou (2015) Energy-efficient train control: the two-train separation problem on level track, paper presented at *6th International Conference on Railway Operations Modelling and Analysis (RailTokyo2015), Narashino, Japan, March 23-26, 2015*.

Almodóvar, M. and R. García-Ródenas (2013) On-line reschedule optimization for passenger railways in case of emergencies, *Computers & Operations Research*, **40** (3): 725–736, ISSN 0305-0548. doi: 10.1016/j.cor.2011.01.013.

Aradi, S., T. Becsi and P. Gaspar (2013) *A predictive optimization method for energy-optimal speed profile generation for trains*. In *CINTI 2013 - 14th IEEE International Symposium on Computational Intelligence and Informatics, Budapest, Hungary, November 19-21, 2013*, 135–139. doi: 10.1109/CINTI.2013.6705179.

Baccelli, F. L., G. Cohen, G. J. Olsder and J.-P. Quadrat (1992) *Synchronization and Linearity: An Algebra for Discrete Event Systems*, Wiley series in probability and mathematical statistics. Chichester: John Wiley and Sons.

Bai, Y., B. Mao, F. Zhou, Y. Ding and C. Dong (2009) Energy efficient driving strategy for freight trains based on power consumption analysis, *Journal of Transportation system engineering and information technology*, **9**: 43–50. doi: 10.1016/S1570-6672(08)60062-8.

Balmelli, G. (2010) Efficient generation of train speed profiles, Master thesis, ETH Zurich.

Bärmann, A. (2016) *Case Study for the German Railway Network*. In *Solving Network Design Problems via Decomposition, Aggregation and Approximation*, 85–102, Wiesbaden: Springer Fachmedien Wiesbaden, ISBN 978-3-658-13913-1. doi: 10.1007/978-3-658-13913-1_6.

Barnhart, C., E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh and P. H. Vance (1998) Branch-and-price: Column generation for solving huge integer programs, *Operations Research*, **46** (3): 316–329. doi: 10.1287/opre.46.3.316.

Bélanger, N., G. Desaulniers, F. Soumis and J. Desrosiers (2006) Periodic airline fleet assignment with time windows, spacing constraints, and time dependent revenues, *European Journal of Operational Research*, **175** (3): 1754–1766, ISSN 0377-2217. doi: 10.1016/j.ejor.2004.04.051.

Binder, S., Y. Maknoon and M. Bierlaire (2015) Passenger-oriented railway disposition timetables in case of severe disruptions, paper presented at *15th Swiss Transport Research Conference (STRC), Ascona, Switzerland, April 2015*.

Borndörfer, R. and T. Schlechte (2007) *Models for Railway Track Allocation*. In C. Liebchen, R. K. Ahuja and J. A. Mesa (eds.) *7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'07)*, vol. 7 of *OpenAccess Series in Informatics (OASIcs)*, 62–78, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, ISBN 978-3-939897-04-0. doi: 10.4230/OASIcs.ATMOS.2007.1170.

Boyd, S., L. Xiao and A. Mutapcic (2003) Subgradient methods, lecture notes, EE392o, Autumn Quarter 2004–2005, Stanford University.

Brännlund, U., P. O. Lindberg, A. Nõu and J. E. Nilsson (1998) Railway timetabling using Lagrangian Relaxation, *Transportation Science*, **32** (4): 358, ISSN 00411655. doi: 10.1287/trsc.32.4.358.

Brünger, O. and E. Dahlhaus (2014) *Running Time Estimation*. In I. A. Hansen and J. Pachl (eds.) *Railway Timetabling and Operations*, 65–89, Hamburg, Germany: Eurail press, 2 edn.

Brüngger, J., P. Grossenbacher, P. Zwahlen, K. Gärtner, T. Wäfler, J. Zimmermann and K. Fischer (2014) Human-machine interaction in the swiss federal railways: anticipating consequences of automation on signalling and train dispatching, paper presented at *First German Workshop on Rail Human Factors, Braunschweig, Germany, March 20-21, 2014*, 1.

Bundesversammlung der Schweizerischen Eidgenossenschaft (2013) Bundesgesetz über die Finanzierung und den Ausbau der Eisenbahninfrastruktur, 21. Juni 2013. Swiss law. `http://www.admin.ch/opc/de/official-compilation/2015/651.pdf`.

Cacchiani, V., A. Caprara and P. Toth (2008) A column generation approach to train timetabling on a corridor, *4OR*, **6** (2): 125–142, ISSN 1619-4500. doi: 10.1007/s10288-007-0037-5.

Cacchiani, V., A. Caprara and P. Toth (2010) Scheduling extra freight trains on railway networks, *Transportation Research Part B: Methodological*, **44** (2): 215–231, ISSN 0191-2615. doi: http://dx.doi.org/10.1016/j.trb.2009.07.007.

Cacchiani, V., D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf and J. Wagenaar (2014) An overview of recovery models and algorithms for real-time railway rescheduling, *Transportation Research Part B: Methodological*, **63** (0): 15–37, ISSN 0191-2615. doi: 10.1016/j.trb.2014.01.009.

Caimi, G., F. Chudak, M. Fuchsberger, M. Laumanns and R. Zenklusen (2011) A new resource-constrained multicommodity flow model for conflict-free train routing and scheduling, *Transportation Science*, **45** (2): 212–227. doi: 10.1287/trsc.1100.0349.

Caimi, G., M. Fuchsberger, M. Laumanns and M. Lüthi (2012) A model predictive control approach for discrete-time rescheduling in complex central railway station areas, *Computers and Operations Research*, **39** (11): 2578–2593, ISSN 0305-0548. doi: 10.1016/j.cor.2012.01.003.

Caimi, G. C. (2009) Algorithmic decision support for train scheduling in a large and highly utilised railway network, Ph.D. thesis, Dissertation No. 18581, ETH Zurich.

Caprara, A., M. Fischetti and P. Toth (2002) Modeling and solving the train timetabling problem, *Operations Research*, **50** (5): 851–861. doi: 10.1287/opre.50.5.851.362.

Chou, Y., P. Weston and C. Roberts (2009) Collaborative rescheduling in a distributed railway control system, paper presented at *3rd international seminar on railway operations modelling and analysis (RailZurich2009), Zurich, Switzerland, February 11-13, 2009*.

Cordeau, J.-F., P. Toth and D. Vigo (1998) A survey of optimization models for train routing and scheduling, *Transportation Science*, **32** (4): 380–404. doi: 10.1287/trsc.32.4.380.

Corman, F., A. D'Ariano, D. Pacciarelli and M. Pranzo (2009) Evaluation of green wave policy in real-time railway traffic management, *Transportation Research Part C: Emerging Technologies*, **17** (6): 607–616, ISSN 0968-090X. doi: 10.1016/j.trc.2009.04.001.

Corman, F., A. D'Ariano, D. Pacciarelli and M. Pranzo (2010a) Centralized versus distributed systems to reschedule trains in two dispatching areas, *Public Transport*, **2**: 219–247, ISSN 1866-749X. doi: 10.1007/s12469-010-0032-7.

Corman, F., A. D'Ariano, D. Pacciarelli and M. Pranzo (2010b) A tabu search algorithm for rerouting trains during rail operations, *Transportation Research Part B: Methodological*, **44** (1): 175–192, ISSN 0191-2615. doi: 10.1016/j.trb.2009.05.004.

Corman, F., A. D'Ariano, D. Pacciarelli and M. Pranzo (2012a) Bi-objective conflict detection and resolution in railway traffic management, *Transportation Research Part C: Emerging Technologies*, **20** (1): 79–94, ISSN 0968-090X. doi: 10.1016/j.trc.2010.09.009.

Corman, F., A. D'Ariano, D. Pacciarelli and M. Pranzo (2012b) Optimal inter-area coordination of train rescheduling decisions, *Transportation Research Part E: Logistics and Transportation Review*, **48** (1): 71–88, ISSN 1366-5545. doi: 10.1016/j.tre.2011.05.002.

Corman, F., A. D'Ariano, D. Pacciarelli and M. Pranzo (2014) Dispatching and coordination in multi-area railway traffic management, *Computers and Operations Research*, **44**: 146–160. doi: 10.1016/j.cor.2013.11.011.

Corman, F., A. D'Ariano, M. Pranzo and I. Hansen (2011) Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area, *Transportation Planning and Technology*, **34** (4): 341–362. doi: 10.1080/03081060.2011.577152.

Corman, F. and L. Meng (2013) *A review of online dynamic models and algorithms for railway traffic control*. In *ICIRT 2013 - IEEE International Conference on Intelligent Rail Transportation*, 128–133. doi: 10.1109/ICIRT.2013.6696281.

D'Ariano, A. and T. Albrecht (2006) *Running time re-optimization during real-time timetable perturbations*. In J. Allan, C. Brebbia, A. Rumsey, G. Sciutto, S. Sone and C. Goodman (eds.) *Computers in Railways X*, vol. 88 of *WIT Transactions on the Built Environment*, 531–540, WIT Press. doi: 10.2495/CR060531.

D'Ariano, A., F. Corman, D. Pacciarelli and M. Pranzo (2008) Reordering and local rerouting strategies to manage train traffic in real time, *Transportation Science*, **42** (4): 405–419. doi: 10.1287/trsc.1080.0247.

D'Ariano, A., D. Pacciarelli and M. Pranzo (2007a) A branch and bound algorithm for scheduling trains in a railway network, *European Journal of Operational Research*, **183** (2): 643–657, ISSN 0377-2217. doi: 10.1016/j.ejor.2006.10.034.

D'Ariano, A., M. Pranzo and I. Hansen (2007b) Conflict resolution and train speed coordination for solving real-time timetable perturbations, *Intelligent Transportation Systems, IEEE Transactions on*, **8** (2): 208–222, June 2007, ISSN 1524-9050. doi: 10.1109/TITS.2006.888605.

D'Ariano, A., M. Samà, P. D'Ariano and D. Pacciarelli (2014) Evaluating the applicability of advanced techniques for practical real-time train scheduling, *Transportation Research Procedia*, **3** (0): 279–288, ISSN 2352-1465. doi: 10.1016/j.trpro.2014.10.007.

Davydov, B., B. Dynkin and V. Chebotarev (2014) Optimal rescheduling for the mixed passenger and freight line, *WIT Transactions on the Built Environment*, **135**: 649–661. doi: 10.2495/CR140541.

De Martinis, V., U. Weidmann and M. Gallo (2014) *Towards a simulation-based framework for evaluating energy-efficient solutions in train operation*. In *COMPRAIL 2014 -14th International Conference on Railway Engineering Design and Optimization, Rome, Italy, June 24-26, 2014*, vol. 135 of *WIT Transactions on the Built Environment*, 721–732.

Degraeve, Z. and L. Schrage (1997) A tire production scheduling system for bridgestone/firestone off-the-road, *Operations Research*, **45** (6): 789–796, ISSN 0030364X. doi: 10.1287/opre.45.6.789.

Desaulniers, G., J. Desrosiers and M. M. Solomon (eds.) (2005) *Column Generation*. Berlin: Springer-Verlag. doi: 10.1007/b135457.

Dezsõ, B., A. Jüttner and P. Kovács (2011) LEMON – an open source C++ graph template library, *Electronic Notes in Theoretical Computer Science*, **264** (5): 23–45, ISSN 1571-0661. doi: 10.1016/j.entcs.2011.06.003.

Dolder, U., M. Krista and M. Voelcker (2009) RCS - rail control system - realtime train run simulation and conflict detection on a net wide scale based on updated train positions, paper presented at *3rd International Seminar on Railway Operations Modelling and Analysis*.

Dollevoet, T., F. Corman, A. D'Ariano and D. Huisman (2014) An iterative optimization framework for delay management and train scheduling, *Flexible Services and Manufacturing Journal*, **26** (4): 490–515, ISSN 1936-6590. doi: 10.1007/s10696-013-9187-2.

Dollevoet, T., D. Huisman, L. Kroon, M. Schmidt and A. Schöbel (2015) Delay management including capacities of stations, *Transportation Science*, **49** (2): 185–203. doi: 10.1287/trsc.2013.0506.

Dollevoet, T., D. Huisman, M. Schmidt and A. Schöbel (2012) Delay management with rerouting of passengers, *Transportation Science*, **46** (1): 74–89. doi: 10.1287/trsc.1110.0375.

Dündar, S. and T. Şahin (2013) Train re-scheduling with genetic algorithms and artificial neural networks for single-track railways, *Transportation Research Part C: Emerging Technologies*, **27**: 1–15. doi: 10.1016/j.trc.2012.11.001.

Espinosa-Aranda, J. L. and R. García-Ródenas (2013) A demand-based weighted train delay approach for rescheduling railway networks in real time, *Journal of Rail Transport Planning & Management*, **3** (1-2): 1–13, ISSN 2210-9706. doi: 10.1016/j.jrtpm.2013.10.001.

Federal Office for Spacial Development (2016) *Perspektiven des Schweizerischen Personen- und Güterverkehrs bis 2040-Synthesebericht*. Bern: Bundesamt für Raumentwicklung, www.are.admin.ch.

Fischer, F. (2013) Dynamic graph generation and an asynchronous parallel bundle method motivated by train timetabling, Ph.D. thesis, Chemnitz University of Technology, http://nbn-resolving.de/urn:nbn:de:bsz:ch1-qucosa-118358.

Fischer, F. and C. Helmberg (2010) *Dynamic Graph Generation and Dynamic Rolling Horizon Techniques in Large Scale Train Timetabling*. In T. Erlebach and M. Lübbecke (eds.) *10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS'10)*, vol. 14 of *OpenAccess Series in Informatics*

(OASIcs), 45–60, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, ISBN 978-3-939897-20-0. doi: 10.4230/OASIcs.ATMOS.2010.45.

Fischer, F., C. Helmberg, J. Janßen and B. Krostitz (2008) *Towards Solving Very Large Scale Train Timetabling Problems by Lagrangian Relaxation*. In M. Fischetti and P. Widmayer (eds.) *8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'08)*, vol. 9 of *OpenAccess Series in Informatics (OASIcs)*, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, ISBN 978-3-939897-07-1. doi: 10.4230/OASIcs.ATMOS.2008.1585.

Foglietta, S., G. Leo, C. Mannino, P. Perticaroli and M. Piacentini (2014) *An optimized, automatic TMS in operations in Roma Tiburtina and Monfalcone stations*. In *COMPRAIL 2014 -14th International Conference on Railway Engineering Design and Optimization, Rome, Italy, June 24-26, 2014*, vol. 135 of *WIT Transactions on the Built Environment*, 635–647. doi: 10.2495/CR140531.

Frank, P. (2013) Methodik zur Effizienzbeurteilung der Kapazitätsnutzung und -entlicklung von Bahnnetzen, Ph.D. thesis, Dissertation No. 21386, ETH Zürich.

Franke, R., P. Terwiesch and M. Meyer (2000) *An algorithm for the optimal control of the driving of trains*. In *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 3, 2123–2128. doi: 10.1109/CDC.2000.914108.

Fuchsberger, M. (2012) Algorithms for railway traffic management in complex central station areas, Ph.D. thesis, Dissertation No. 20398, ETH Zurich.

Galli, L. (2017) Algorithms for integer programming, lecture notes, Ricerca operativa II, University of Pisa, `http://www.di.unipi.it/optimize/Courses/RO2IG/aa1718/IP_algorithms.pdf`.

Gholami, O. and Y. N. Sotskov (2012) Train routing and timetabling via a genetic algorithm, *IFAC Proceedings Volumes*, **45** (6): 158–163, ISSN 1474-6670. doi: 10.3182/20120523-3-RO-2023.00294.

Großmann, P., S. Hölldobler, N. Manthey, K. Nachtigall, J. Opitz and P. Steinke (2012) Solving periodic event scheduling problems with sat, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **7345 LNAI**: 166–175. doi: 10.1007/978-3-642-31087-4_18.

Gurobi Optimization, I. (2016) Gurobi optimizer reference manual, `http://www.gurobi.com`.

Haimes, Y., L. Lasdon and D. Wismer (1971) On a bicriterion formation of the problems of integrated system identification and system optimization, *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-1** (3): 296–297.

Hansen, I. A. and J. Pachl (eds.) (2014) *Railway Timetabling and Operations*, 2 edn. Hamburg, Germany: DVV Media Group GmbH, Eurailpress.

Helmberg, C. (2005) The ConicBundle library for convex optimization, `https://www-user.tu-chemnitz.de/~helmberg/ConicBundle/`.

Herrigel, S., M. Laumanns, J. Szabo and U. Weidmann (2013a) Periodic railway timetabling with hierarchical decomposition in the PESP model, paper presented at *5th International Seminar on Railway Operations Modelling (RailCompenhagen2013)*.

Herrigel, S., M. Laumanns, U. A. Weidmann and A. Nash (2013b) Hierarchical decomposition methods for periodic railway timetabling problems, *Transportation Research Record*, **2374**: 73–82. doi: 10.3141/2374-09.

Herrmann, T. M. (2006) Stability of timetables and train routings through station regions, Ph.D. thesis, Dissertation No. 16398, ETH Zürich.

Hooghiemstra, J., L. Kroon, M. Odijk, M. Salomon and P. Zwaneveld (1999) Decision support systems support the search for win-win solutions in railway network design, *Interfaces*, **29** (2): 15–32. doi: 10.1287/inte.29.2.15.

Howlett, P. (2000) The optimal control of a train, *Annals of Operations Research*, **98**: 65–87. doi: 10.1023/A:1019235819716.

Huisman, D., R. Freling and A. Wagelmans (2005) Multiple-depot integrated vehicle and crew scheduling, *Transportation Science*, **39** (4): 491–502. doi: 10.1287/trsc.1040.0104.

Hürlimann, D. (2002) Objektorientierte Modellierung von Infrastrukturelementen und Betriebsvorgängen im Eisenbahnwesen, Ph.D. thesis, ETH Zurich.

IBM (2015) IBM ILOG CPLEX optimization studio user's manual, `http://www.ibm.com`.

Iyer, R. and S. Ghosh (1995) DARYN - a distributed decision-making algorithm for railway networks: modeling and simulation, *IEEE Transactions on Vehicular Technology*, **44** (1): 180–191, ISSN 0018-9545. doi: 10.1109/25.350284.

Jia, L.-M. and X.-D. Zhang (1994) Distributed intelligent railway traffic control: A fuzzy-decisionmaking-based approach, *Engineering Applications of Artificial Intelligence*, **7** (3): 311–319, ISSN 0952-1976. doi: 10.1016/0952-1976(94)90058-2.

Kanai, S., K. Shiina, S. Harada and N. Tomii (2011) An optimal delay management algorithm from passengers' viewpoints considering the whole railway network, *Journal of Rail Transport Planning & Management*, **1** (1): 25–37, ISSN 2210-9706. doi: 10.1016/j.jrtpm.2011.09.003.

Kecman, P., F. Corman, A. D'Ariano and R. Goverde (2013) Rescheduling models for railway traffic management in large-scale networks, *Public Transport*, **5** (1-2): 95–123. doi: 10.1007/s12469-013-0063-y.

Kecman, P., F. Corman, A. Peterson and M. Joborn (2015) Stochastic prediction of train delays in real-time using Bayesian networks, paper presented at *Conference on Advanced Systems in Public Transport (CASPT2015)*.

Kersbergen, B., T. Van Den Boom and B. De Schutter (2014) *Distributed model predictive control for rescheduling of railway traffic*. In *17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, 2732–2737. doi: 10.1109/ITSC.2014.6958127.

Kersbergen, B., T. van den Boom and B. D. Schutter (2016) Distributed model predictive control for railway traffic management, *Transportation Research Part C: Emerging Technologies*, **68**: 462–489, ISSN 0968-090X. doi: 10.1016/j.trc.2016.05.006.

Khmelnitsky, E. (2000) On an optimal control problem of train operation, *IEEE Transactions on Automatic Control*, **45**: 1257–1266. doi: 10.1109/9.867018.

Kraay, D. and P. Harker (1995) Real-time scheduling of freight railroads, *Transportation Research Part B*, **29** (3): 213–229. doi: 10.1016/0191-2615(94)00034-W.

Kroon, L., D. Huisman, E. Abbink, P.-J. Fioole, M. Fischetti, G. Maròti, A. Schrijver, A. Steenbeek and R. Ybema (2009) The new dutch timetable: The OR revolution, *Interfaces*, **39** (1): 6–17. doi: 10.1287/inte.1080.0409.

Kümmling, M., P. Großmann, K. Nachtigall, J. Opitz and R. Weiß (2015) A state-of-the-art realization of cyclic railway timetable computation, *Public Transport*, **7** (3): 281–293. doi: 10.1007/s12469-015-0108-5.

Lamma, E., P. Mello and M. Milano (1997) A distributed constraint-based scheduler, *Artificial Intelligence in Engineering*, **11** (2): 91–105, ISSN 0954-1810. doi: 10.1016/S0954-1810(96)00002-7.

Lamorgese, L., C. Mannino and M. Piacentini (2016) Optimal train dispatching by Benders'-like reformulation, *Transportation Science*, **50** (3): 910–925. doi: 10.1287/trsc.2015.0605.

Landex, A. (2008) Methods to estimate railway capacity and passenger delays, Ph.D. thesis, Technical University of Denmark.

Laube, F., S. Roos, R. Wüst, M. Lüthi and U. Weidmann (2007) Puls 90 - ein systemumfassender Ansatz zur Leistungssteigerung von Eisenbahnnetzen, *ETR - Eisenbahntechnische Rundschau*, **03**: 104–107.

Laumanns, M., L. Thiele and E. Zitzler (2006) An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method, *European Journal of Operational Research*, **169** (3): 932–942, ISSN 0377-2217. doi: 10.1016/j.ejor.2004.08.029.

Lee, T. and S. Ghosh (1998) RYNSORD: a novel decentralized algorithm for railway networks with "soft reservation", *IEEE Transactions on Vehicular Technology*, **47** (4): 1350–1364. doi: 10.1109/25.728526.

Li, L., D. Wen and D. Yao (2014) A survey of traffic control with vehicular communications, *IEEE Transactions on Intelligent Transportation Systems*, **15** (1): 425–432. doi: 10.1109/TITS.2013.2277737.

Li, X. and H. Lo (2014) An energy-efficient scheduling and speed control approach for metro rail operations, *Transportation Research Part B: Methodological*, **64**: 73–89. doi: 10.1016/j.trb.2014.03.006.

Lin, M.-H., J. G. Carlsson, D. Ge, J. Shi and J.-F. Tsai (2013) A review of piecewise linearization methods, *Mathematical Problems in Engineering*, **2013**. doi: 10.1155/2013/101376.

Lindner, T. (2011) Applicability of the analytical UIC code 406 compression method for evaluating line and station capacity, *Journal of Rail Transport Planning & Management*, **1** (1): 49–57, ISSN 2210-9706. doi: 10.1016/j.jrtpm.2011.09.002.

Liu, R. and I. M. Golovicher (2003) Energy-efficient operation of rail vehicles, *Transportation Research Part A*, **37**: 917–931. doi: 10.1016/j.tra.2003.07.001.

Lu, Q., M. Dessouky and R. Leachman (2004) Modeling train movements through complex rail networks, *ACM Transactions on Modeling and Computer Simulation*, **14** (1): 48–75.

Lübbecke, M. E. and J. Desrosiers (2005) Selected topics in column generation, *Operations Research*, **53** (6): 1007–1023. doi: 10.1287/opre.1050.0234.

Lusby, R. M., J. Larsen, M. Ehrgott and D. Ryan (2011) Railway track allocation: models and methods, *OR Spectrum*, **33** (4): 843–883, ISSN 0171-6468. doi: 10.1007/s00291-009-0189-0.

Lusby, R. M., J. Larsen, M. Ehrgott and D. M. Ryan (2013) A set packing inspired method for real-time junction train routing, *Computers & Operations Research*, **40** (3): 713–724, ISSN 0305-0548. doi: 10.1016/j.cor.2011.12.004.

Lüthi, M. (2009) Improving the efficiency of heavily used railway networks through integrated real-time rescheduling, Ph.D. thesis, Dissertation No. 18615, ETH Zürich.

Mascis, A. and D. Pacciarelli (2002) Job-shop scheduling with blocking and no-wait constraints, *European Journal of Operational Research*, **143** (3): 498–517, ISSN 0377-2217. doi: 10.1016/S0377-2217(01)00338-1.

Mazzarello, M. and E. Ottaviani (2007) A traffic management system for real-time traffic optimisation in railways, *Transportation Research Part B: Methodological*, **41** (2): 246–274, ISSN 0191-2615. doi: 10.1016/j.trb.2006.02.005.

Mehta, F., C. Rößiger and M. Montigel (2010) *Latent energy savings due to the innovative use of advisory speeds to avoid occupation conflicts*. In N. Ning and C. Brebbia (eds.) *Computers in Railways XII*, vol. 114 of *WIT Transactions on The Built Environment*, 99–108, WIT Press. doi: 10.2495/CR100101.

Meng, L. and X. Zhou (2014) Simultaneous train rerouting and rescheduling on an N-track network: A model reformulation with network-based cumulative flow variables, *Transportation Research Part B: Methodological*, **67** (0): 208–234, ISSN 0191-2615. doi: 10.1016/j.trb.2014.05.005.

Meyer, A. (2017) Smart unterwegs in die Zukunft, presented at SmartRail 4.0 Industrietag, Bern, Switzerland, August 16, 2017.

Min, Y.-H., M.-J. Park, S.-P. Hong and S.-H. Hong (2011) An appraisal of a column-generation-based algorithm for centralized train-conflict resolution on a metropolitan railway network, *Transportation Research Part B: Methodological*, **45** (2): 409–429, ISSN 0191-2615. doi: 10.1016/j.trb.2010.08.001.

Montigel, M. (2009) Operations control system in the Lötschberg base tunnel, *Rail Technology Review*, no. 2, 42–44.

Mu, S. and M. M. Dessouky (2014) Control rules for dispatching trains on general networks with multiple train speeds, paper presented at *Conference Proceedings Transport Research Arena 2014, Paris, France*.

Müller, N. (2017) Analyse der Häufigkeit und Art von Konflikten und dispositiven Eingriffen in einer Betriebszentrale, *Semester paper*, ETH Zurich.

Murali, P., F. Ordóñez and M. M. Dessouky (2016) Modeling strategies for effectively routing freight trains through complex networks, *Transportation Research Part C: Emerging Technologies*, **70**: 197–213, ISSN 0968-090X. doi: 10.1016/j.trc.2015.08.022.

Narayanaswami, S. and N. Rangaraj (2015) A MAS architecture for dynamic, realtime rescheduling and learning applied to railway transportation, *Expert Systems with Applications*, **42** (5): 2638–2656, ISSN 0957-4174. doi: 10.1016/j.eswa.2014.11.013.

Nedic, A. and A. Ozdaglar (2009) Distributed subgradient methods for multi-agent optimization, *IEEE Transactions on Automatic Control*, **54** (1): 48–61, ISSN 0018-9286. doi: 10.1109/TAC.2008.2009515.

Oettich, S. and R. Caspar (2015) RCS-HOT-automated optimization of train movement in dense networks, paper presented at *3rd WORKSHOP ICT ON TRAINS, Birmingham, UK, September 8-9, 2015*.

Official Timetable (2017) Graphic timetables, `http://www.fahrplanfelder.ch/en/archives/graphic-timetables.html`.

OpenTrack Railway Technology Ltd. (2015) OPENTRACK simulation of railway networks-user's manual, `http://www.opentrack.ch`.

Pachl, J. (2008) Timetable design principles, In I. A. Hansen and J. Pachl (eds.) *Railway Timetable and Traffic*, 1 edn., 9–42, Hamburg, Germany: Eurail press.

Pachl, J. (2015) *Railway Operation and Control*, 3 edn. Gorham Printing, Rochester WA: VTD Rail Publishing.

Parbo, J., O. A. Nielsen and C. G. Prato (2016) Passenger perspectives in railway timetabling: A literature review, *Transport Reviews*, **36** (4): 500–526. doi: 10.1080/01441647.2015.1113574.

Parkes, D. and L. Ungar (2001) *An auction-based method for decentralized train scheduling*. In *Proceedings of the International Conference on Autonomous Agents*, 43–50. doi: 10.1145/375735.375866.

Parodi, G., G. Vernazza and R. Zunino (1996) Stability and deadlock avoidance in distributed systems for traffic control, *IEEE Transactions on Vehicular Technology*, **45** (4): 732–743. doi: 10.1109/25.543743.

Peeters, L. and L. Kroon (2001) A cycle based optimization model for the cyclic railway timetabling problem, In S. Voss and J. R. Daduna (eds.) *Computer-Aided Scheduling of Public Transport*, vol. 505 of *Lecture Notes in Economics and Mathematical Systems*, 275–296, Berlin Heidelberg: Springer, ISBN 978-3-540-42243-3. doi: 10.1007/978-3-642-56423-9-16.

Pellegrini, P., G. Marlière and J. Rodriguez (2014) Optimal train routing and scheduling for managing traffic perturbations in complex junctions, *Transportation Research Part B: Methodological*, **59**: 58–80, ISSN 0191-2615. doi: 10.1016/j.trb.2013.10.013.

Plutowiki (2017) Schmalspurbahnen der Schweiz, Wikimedia-Bild mit CC-SA-2.0-Lizenz `https://de.wikipedia.org/`

wiki/Datei:Schmalspurnetz_Schweiz.svg. Accessed 25.09.2017.

Proença, H. and E. Oliveira (2005) An adaptive multi-agent system for railway traffic control, *GESTS International Transactions*, **9** (1):.

Quaglietta, E. (2011) A microscopic simulation model for supporting the design of railway systems: Development and applications, Ph.D. thesis, University of Naples Federico II, Naples.

R I-30111 complement (2015) *Bezeichnung der Züge Fahrplan 2016-Désignation des trains Horaire 2016*. Complement to technical guideline SBB, version V9.2 (04.11.2015), SBB AG, BLS AG, SOB AG, available at https://www.sbb.ch/content/sbb/de/desktop/sbb-konzern/sbb-als-geschaeftspartner/zugang-zum-bahn-netz/onestopshop/grundlagen-netzzugang/_jcr_content/contentPar/tabs/tabItems/allgemeine_nutzungsb/tabPar/downloadlist_0/downloadList/bezeichnung_der_z_ge.spooler.download.pdf, accessed 02.02.2016.

R I-30131 (2016) Streckentabellen-tableaux des parcours-tabelle delle tratte, technical guideline SBB, version 249-0 (07.02.2016), SBB AG, BLS AG, SOB AG.

Radtke, A. (2014) Infrastructure modelling, In I. A. Hansen and J. Pachl (eds.) *Railway Timetabling and Operations*, chap. 3, 47–63, Hamburg, Germany: Eurail press.

Rao, X. (2015) Holistic rail network operation by integration of train automation and traffic management, Ph.D. thesis, ETH Zurich.

Rao, X., M. Montigel and U. Weidmann (2016) A new rail optimisation model by integration of traffic management and train automation, *Transportation Research Part C: Emerging Technologies*, **71**: 382–405, ISSN 0968-090X. doi: 10.1016/j.trc.2016.08.011.

Reisezüge.ch (2005-2011) Zugskompositionen der Schweiz online, www.reisezuege.ch. Accessed 10.02.2016.

Rinaldi, M. and C. M. Tampère (2015) An extended coordinate descent method for distributed anticipatory network traffic control, *Transportation Research Part B: Methodological*, **80**: 107–131, ISSN 0191-2615. doi: 10.1016/j.trb.2015.06.017.

rmcon (2015) Railsys, booklet, available on `http://www.rmcon.de/`.

Rodriguez, J. (2007) A constraint programming model for real-time train scheduling at junctions, *Transportation Research Part B: Methodological*, **41** (2): 231–245, ISSN 0191-2615. doi: 10.1016/j.trb.2006.02.006.

Rückert, R., M. Lemnian, C. Blendinger, S. Rechner and M. Müller-Hannemann (2017) PANDA: a software tool for improved train dispatching with focus on passenger flows, *Public Transport*, **9** (1): 307–324, Jul 2017, ISSN 1613-7159. doi: 10.1007/s12469-016-0140-0.

Salido, M., M. Abril, F. Barber, L. Ingolotti, P. Tormos and A. Lova (2007) Domain-dependent distributed models for railway scheduling, *Knowledge-Based Systems*, **20** (2): 186–194, ISSN 0950-7051. doi: 10.1016/j.knosys.2006.11.013.

Samà, M., A. D'Ariano, F. Corman and D. Pacciarelli (2017) A variable neighbourhood search for fast train scheduling and routing during disturbed railway traffic situations, *Computers & Operations Research*, **78**: 480–499, ISSN 0305-0548. doi: 10.1016/j.cor.2016.02.008.

Samà, M., C. Meloni, A. D'Ariano and F. Corman (2015) A multi-criteria decision support methodology for real-time train scheduling, *Journal of Rail Transport Planning & Management*, **5** (3): 146–162, ISSN 2210-9706. doi: 10.1016/j.jrtpm.2015.08.001.

Samà, M., P. Pellegrini, A. D'Ariano, J. Rodriguez and D. Pacciarelli (2016) Ant colony optimization for the real-time train routing selection problem, *Transportation Research Part B: Methodological*, **85**: 89–108, ISSN 0191-2615. doi: 10.1016/j.trb.2016.01.005.

Sato, K., K. Tamura and N. Tomii (2013) A mip-based timetable rescheduling formulation and algorithm minimizing further inconvenience to passengers, *Journal of Rail Transport Planning & Management*, **3** (3): 38–53, ISSN 2210-9706. doi: 10.1016/j.jrtpm.2013.10.007.

SBB AG (2014a) Infrastrukturbetreiber-gestionnaires d'infrastructure-gestori di infrastruttura-infrastructure managers, `http://www.sbb.ch/sbb-konzern/sbb-als-geschaeftspartnerin/angebote-fuer-evus/onestopshop/strecken-bahnanlagen.html`. Accessed 03.02.2016.

SBB AG (2014b) *Reglement über die Betriebsführung*.

SBB AG (2015) Die SBB in Zahlen und Fakten 2015, sbb.ch/zahlen-und-fakten.

Schöbel, A. (2007) *Integer Programming Approaches for Solving the Delay Management Problem.* In *Algorithmic Methods for Railway Optimization. Lecture Notes in Computer Science*, vol. 4359, 145–170, Berlin, Heidelberg: Springer, ISBN 978-3-540-74247-0. doi: 10.1007/978-3-540-74247-0_7.

Schöbel, A. (2009) Capacity constraints in delay management, *Public Transport*, **1** (2): 135–154, ISSN 1866-749X. doi: 10.1007/s12469-009-0010-0.

Schranil, S. and U. Weidmann (2013) Forecasting the duration of rail operation disturbances, In *TRB 92nd Annual Meeting Compendium of Papers*, 500 Fifth Street, NW, Washington, DC 20001 USA: Transportation Research Board.

Serafini, P. and W. Ukovich (1989) A mathematical model for periodic scheduling problems, *SIAM J. Disc. Math.*, **2** (4): 550–581. doi: 10.1137/0402049.

Sethi, S. and G. Sorger (1991) A theory of rolling horizon decision making, *Annals of Operations Research*, **29** (1): 387–415, Dec 1991, ISSN 1572-9338. doi: 10.1007/BF02283607.

Siemens (2018) Iltis-Leittechnik, https://w1.siemens.ch/mobility/ch/de/nahverkehr/schienenverkehr/bahnautomatisierung/betriebsleittechnik/iltis/seiten/iltis.aspx.

Stasko, T., B. Levine and A. Reddy (2016) Time-expanded network model of train-level subway ridership flows using actual train movement data, *Transportation Research Record*, **2540**: 92–101. doi: 10.3141/2540-11.

Strässle, S. and T. Schneeberger (2017) CUS, das zentrale Kundeninformationssystem für den öV Schweiz, *SIGNAL + DRAHT*, no. 3, 40–44.

Strotmann, C. (2007) Railway scheduling problems and their decomposition, Ph.D. thesis, Universität Osnabrück.

Su, S., T. Tang and C. Roberts (2015) A cooperative train control model for energy saving, *IEEE Transactions on Intelligent Transportation Systems*, **16** (2): 622–631, April 2015, ISSN 1524-9050. doi: 10.1109/TITS.2014.2334061.

Tamannaei, M., M. Saffarzadeh, A. Jamili and S. Seyedabrishami (2016) A double-track train rescheduling for incident conditions: Optimisation model and decomposition method, *International Journal of Operational Research*, **26** (1): 62–87. doi: 10.1504/IJOR.2016.075650.

Terelius, H., U. Topcu and R. M. Murray (2011) *Decentralized Multi-Agent Optimization via Dual Decomposition*. vol. 44, 11245–11251. doi: 10.3182/20110828-6-IT-1002.01959.

Toletti, A., V. De Martinis and U. Weidmann (2015a) What about train length and energy efficiency of freight trains in rescheduling models?, *Transportation Research Procedia*, **10**: 584–594, ISSN 2352-1465. doi: 10.1016/j.trpro.2015.09.012.

Toletti, A., M. Laumanns, V. De Martinis and U. Weidmann (2017) Multi-criteria train rescheduling by means of an adaptive epsilon-constraint method, paper presented at *6th International Conference on Railway Operations Modelling and Analysis (RailLille2017), Lille, France, April 4-7, 2017*.

Toletti, A., M. Laumanns, P. Grossenbacher and U. Weidmann (2015b) Meeting functional requirements for real-time railway traffic management with mathematical models, paper presented at *Conference on Advanced Systems in Public Transport (CASPT2015)*.

Toletti, A., V. D. Martinis and U. Weidmann (2016) *Energy savings in mixed rail traffic rescheduling: an RCG approach*. In *ITSC2016 - IEEE 19th International Conference on Intelligent Transportation Systems*, 2430–2435. doi: 10.1109/ITSC.2016.7795947.

Toletti, A. and U. Weidmann (2016) Modelling customer inconvenience in train rescheduling, paper presented at *Swiss Transport Research Conference (STRC2016)*.

Tomii, N., Y. Tashiro, N. Tanabe, C. Hirai and K. Muraki (2005) *Train Rescheduling Algorithm Which Minimizes Passengers' Dissatisfaction*. In M. Ali and F. Esposito (eds.) *Innovations in Applied Artificial Intelligence*, vol. 3533 of *Lecture Notes in Computer Science*, 829–838, Berlin Heidelberg: Springer, ISBN 978-3-540-26551-1. doi: 10.1007/11504894_113.

Törnquist, J. (2006) *Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms*. In L. G. Kroon and R. H. Möhring (eds.) *5th Workshop on Algorithmic Methods* *and Models for Optimization of Railways (ATMOS'05)*, vol. 2 of *OpenAccess Series in Informatics (OASIcs)*, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, ISBN 978-3-939897-00-2. doi: 10.4230/OASIcs.ATMOS.2005.659.

Törnquist, J. and J. A. Persson (2007) N-tracked railway traffic re-scheduling during disturbances, *Transportation Research Part B: Methodological*, **41** (3): 342–362, ISSN 0191-2615. doi: 10.1016/j.trb.2006.06.001.

UIC-code 406 (2004) *Capacity*, Union international des chemins de fer (UIC), 1 edn., june 2004.

Ulmke, C. (2017) Objectives for real-time freight traffic management, *Semester paper*, ETH Zurich.

van den Boom, T., B. Kersbergen and B. De Schutter (2012) *Structured modeling, analysis, and control of complex railway operations*. In *2012 IEEE 51th Annual Conference on Decision and Control*, 7366–7371. doi: 10.1109/CDC.2012.6426646.

van den Top, J. (2010) *Modelling Risk Control Measures in Railways*. Next Generation Infrastructures Foundation, ISBN 978-90-79787-15-9, `www.nginfra.nl`.

Venables, W. N. and B. D. Ripley (2002) *Modern Applied Statistics with S*, 4 edn. New York: Springer, `http://www.stats.ox.ac.uk/pub/MASS4`. ISBN 0-387-95457-0.

Vernazza, G. and R. Zunino (1990) A distributed intelligence methodology for railway traffic control, *IEEE Transactions on Vehicular Technology*, **39** (3): 263–270, Aug 1990, ISSN 0018-9545. doi: 10.1109/25.131008.

Völker, M. (2013) Adaptive Lenkung bei den SBB, paper presented at *IT13.rail*.

Weidmann, U. (2015) Extending safety, stability and capacity of rail networks by full automation, paper presented at *Spring Seminar Series 2015, Risk Center, ETH Zurich, March 2015*.

Weidmann, U., M. Laumanns, M. Montigel and X. Rao (2015) Dynamic capacity optimization by fully automated rail operation, *Railway update*, no. 3-4, 58–63.

Weidmann, U. and M. Sinner (2016) Automatisierte Bahnproduktion, *Technical report*, IVT, ETH Zurich. Project C-16-02.

Weidmann, U., J. Wichser, D. Bruckmann, P. Frank, T. Fumasoli, M. Ruesch, P. Hegi, N. Schick and T. Wieczorek (2014) Anforderungen der Güterlogistik an die Netzinfrastruktur und die langfristige Netzentwicklung in der Schweiz : Forschungspaket UVEK/ASTRA "Strategien zum wesensgerechten Einsatz der Verkehrsmittel im Güterverkehr der Schweiz", Teilprojekt C: Logistics requirements in network infrastructure and long-term network design in Switzerland = Exigences de la logistique à l'infrastructure de réseau et au développement long-terme du réseau suisse, *Technical Report*, Bern.

Wendler, E. (2007) Influence of ETCS on line capacity, Presentation at UIC ERMTS World Conference, Bern, Switzerland, September 11-13, 2007.

Westphal, S. and S. Krumke (2008) Pruning in column generation for service vehicle dispatching, *Annals of Operations Research*, **159** (1): 355–371, ISSN 0254-5330. doi: 10.1007/s10479-007-0275-0.

Xie, X.-F., S. Smith, T.-W. Chen and G. Barlow (2014) *Real-time traffic control for sustainable urban living*. In *17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, 1863–1868. doi: 10.1109/ITSC.2014.6957964.

Yan, C. and L. Yang (2012) Mixed-integer programming based approaches for the movement planner problem: Model, heuristics and decomposition, `http://www.informs.org/Community/RAS/Problem-Solving-Competition/2012-RAS-Problem-Solving-Competition`. 2012 RAS Problem Solving Competition Report.

Z/C 505.1 (2014) *Richtlinien für die Beförderung der Reisenden und die Nachführung von Zügen, EWP und SWP-Kurswagen im schweizerischen- und internationalen Verkehr in Verspätungsfällen-Directives pour le transport des voyageurs, l'acheminement des trains, courses EWP / SWP en trafic interne et international en cas de retards.*, SBB, BLS.

Z/C 505.2 (2014) *Vorbehaltene Entschlüsse-Décisions réservées - Decisioni pianificate*, SBB, BLS.

Zhou, X. and M. Zhong (2007) Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds, *Transportation Research Part B: Methodological*, **41** (3): 320–341. doi: 10.1016/j.trb.2006.05.003.

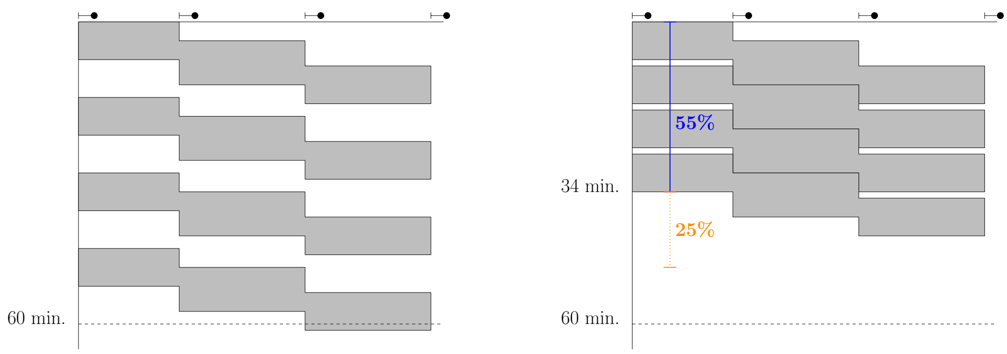
Appendices

Appendix A

Examples of capacity utilisation

The examples below show how different train-mix, timetables, and stability requirements influence line capacity consumption.

Figure A.1(a) shows completely homogeneous traffic over a period of one hour on a line consisting of three blocks. Figure A.1(b) shows the compression, which gives and occupation ratio of the first block equals 56%. With the standard stability requirement of 25% of unplanned capacity, the line is not congested. Obviously, increasing the number of

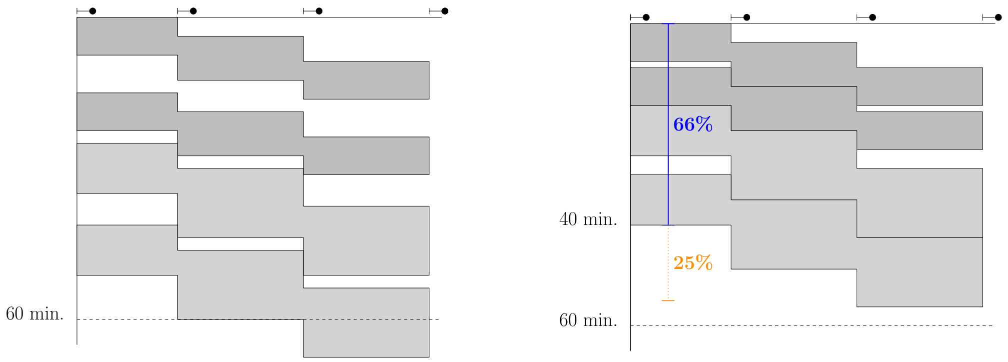


(a) Planned train routes (b) Capacity consumption

Figure A.1: Homogeneous traffic: Line section occupation through compression (blue solid) and stability requirement (orange dotted) do not result in congestion.

trains per time period increases capacity consumption, but even replacing some trains by slower trains increases the network occupation. Figure A.2 shows that replacing the last trains in the previous example by slower ones increases occupation to 66% but does not yield to congestion.

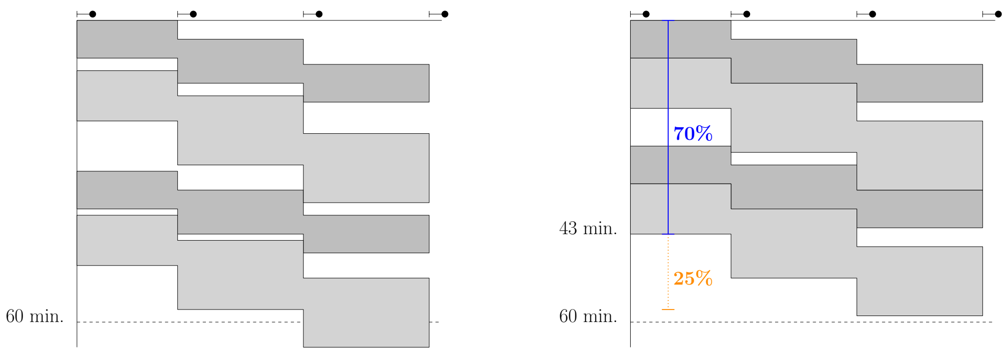
Timetable structure influences capacity consumption too, as shown in Figure A.3. In fact, if the trains of the previous example are not grouped



(a) Planned train routes (b) Capacity consumption

Figure A.2: Heterogeneous traffic: Line section occupation through compression (blue solid) and stability requirement (orange dotted) do not result in congestion.

by their speed the occupation increases due to the time needed between slow trains and fast trains in early tracks and between fast trains and slow trains in late tracks.

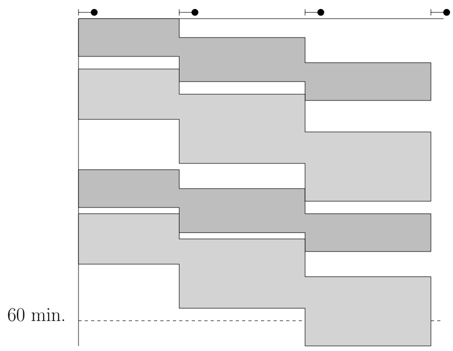


(a) Planned train routes (b) Capacity consumption

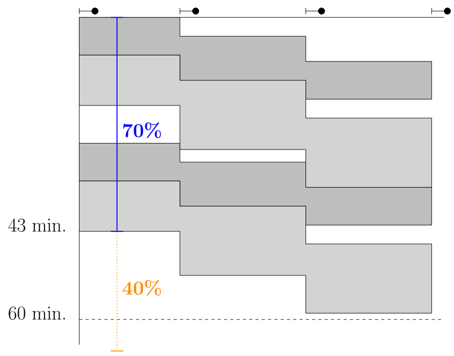
Figure A.3: Timetable not bundling heterogeneous traffic: Line section occupation through compression (blue solid) and stability requirement (orange dotted) do not result in congestion.

Higher stability requirements increase line capacity consumption. Running time supplements increase infrastructure occupation, while buffer times simply add up as shown in previous examples. If more restrictive stability requirements (e.g., unscheduled capacity equals 40%) are included in the previous example, the line section may become congested, with a capacity consumption greater than 100%.

These examples show how traffic heterogeneity, timetable structure, and stability requirements affect capacity consumption.



(a) Planned network occupation in time



(b) Compressed stairways

Figure A.4: Heterogeneous traffic with non-harmonized timetable and higher stability requirements: Line section occupation through compression (blue solid) and stability requirement (orange dotted) result in congestion.

Appendix B

Documents used for SBB test case

B.1 Track topology

SBB Infrastruktur Projekte. *Baden - Übersichtsplan Sicherungsanlagen*, Plan Nr. 38022/100/101/1, version 10.11.16. Zurich.

SBB Infrastruktur Projekte. *Brugg - Übersichtsplan Sicherungsanlagen*, Plan Nr. 2BG-U048-n, version 18.11.16. Luzern.

SBB Infrastruktur Projekte. *Döttingen- Übersichtsplan Sicherungsanlagen*, Plan Nr. 38237-100-101-1, version 31.10.16. Zurich.

SBB Infrastruktur Projekte. *Effingen - Übersichtsplan Sicherungsanlagen*, Plan Nr. 2EFG-U001-c1-n, version 23.05.14. Luzern.

SBB Infrastruktur Projekte. *Frick - Übersichtsplan Sicherungsanlagen*, Plan Nr. 2FCK-U032-d1, version 05.12.16. Luzern.

SBB Infrastruktur Projekte. *Hornussen - Übersichtsplan Sicherungsanlagen*, Plan Nr. 2HNS-U024-e, version 01.07.14. Luzern.

SBB Infrastruktur Projekte. *Killwangen-Spreitenbach+ Hst Neuenhof, SpW Langacher- Übersichtsplan Sicherungsanlagen*, Plan Nr. 38016/100/101/1, version 17.10.16. Zurich.

SBB Infrastruktur Projekte. *Koblenz- Übersichtsplan Sicherungsanlagen*, Plan Nr. 30948/100/101/1, version 01.10.13. Zurich.

SBB Infrastruktur Projekte. *Lupfig - Übersichtsplan Sicherungsanlagen*, Plan Nr. 2LUPF-U001-h1, version 04.03.16. Luzern.

SBB Infrastruktur Projekte. *Mägenwil - Übersichtsplan Sicherungsanlagen*, Plan Nr. 2MAEG-U022-q, version 23.01.17. Luzern.

SBB Infrastruktur Projekte. *Mellingen (Dienststation Gruement) - Übersichtsplan Sicherungsanlagen*, Plan Nr. 2MEL-U001-f, version 28.10.14. Luzern.

SBB Infrastruktur Projekte. *Othmarsingen - Übersichtsplan Sicherungsanlagen*, Plan Nr. 2OTH-U019-j, version 04.10.16. Luzern.

SBB Infrastruktur Projekte. *Rupperswil - Übersichtsplan Sicherungsanlagen*, Plan Nr. 2RU-U035-a1, version 07.04.15. Luzern.

SBB Infrastruktur Projekte. *Schninznach Dorf - Übersichtsplan Sicherungsanlagen*, Plan Nr. 2SDO-U009-p, version 08.05.14. Luzern.

SBB Infrastruktur Projekte. *Siggenthal - Würenlingen - Übersichtsplan Sicherungsanlagen*, Plan Nr. 38231/100/101/1, version 15.05.17. Zurich.

SBB Infrastruktur Projekte. *Turgi - Übersichtsplan Sicherungsanlagen*, Plan Nr. 38027/100/101/1, version 31.10.16. Zurich.

SBB Infrastruktur Projekte. *Villnachern - Übersichtsplan Sicherungsanlagen*, Plan Nr. 2VILL-U001-k-p, version 22.04.14, Luzern.

SBB Infrastruktur Projekte. *Wettingen- Übersichtsplan Sicherungsanlagen*, Plan Nr. 38020/100/101/1, version 11.08.15. Zurich.

SBB Infrastruktur Projekte. *Wildegg + Holderbanck AG + Schninznach Bad - Übersichtsplan Sicherungsanlagen*, Plan Nr. 2WI-U019-b1, version 29.09.14. Luzern.

SBB Infrastruktur Projekte. *Würenlos - Übersichtsplan Sicherungsanlagen*, Plan Nr. 32839/100/101/1, version 02.05.16. Zurich.

B.2 Route setting

SBB. *Handbuch Betrieb Baden*. I-B 87/12, version 17.08.2016.

SBB. *Handbuch Betrieb Brugg AG*. I-B 92018, version 0-0, 23.04.2012.

SBB. *Handbuch Betrieb Döttingen inkl. Haltestelle Klingnau*, I-B 93045, version 3-0, 24.04.2014.

SBB. *BF-Handbuch Effingen*, D BFR Mitte 20/08, version 16.02.2010.

SBB. *Handbuch Betrieb Frick*, I-B 92036, version 2-0, 31.10.2016.

SBB. *Handbuch Betrieb Hornussen*, I-B 92047, version 1-0, 30.06.2014.

SBB. *Handbuch Betrieb Killwangen-Spreitenbach*, I-B 93092, version 1-0, 27.05.2013.

SBB. *Handbuch Betrieb Koblenz + Schwaderloch inkl. Haltestellen Koblenz Dorf und Waldshut*, I-B 93095, version 2-3, 20.10.2015.

SBB. *Handbuch Betrieb Lupfig*, I-B 92065, version 1-0, 01.04.2014.

SBB. *Handbuch- Betrieb Mägenwil*, I-B 92070, version 1-0, 02.12.2013.

SBB. *Handbuch Betrieb Mellingen*, I-B 92073, version 1-0, 02.12.2013.

SBB. *Handbuch- Betrieb Othmarsingen*, I-B 92091, version 3-0, 28.11.2016.

SBB. Handbuch- Betrieb Rupperswil, I-B 92099, version 1-0, 01.07.2016.

SBB. *BF-Handbuch Schninznach Dorf*, I-B 92102, version 1-0, 18.04.2016.

SBB. *Handbuch Betrieb Siggenthal – Würenlingen*, I-B 93170, version 3-0, 21.11.2016. Zurich.

SBB. *Handbuch Betrieb Turgi*, I-B 93193, version 1-0, 27.05.2013. Zurich.

SBB. *Handbuch Betrieb Wettingen*, I-B 93211, version 1-0, 19.06.2013.

SBB. *Handbuch Betrieb Wildegg*, I-B 92128, version 1-0, 02.06.2016.

SBB. *Handbuch Betrieb Würenlos*, I-B 93220, version 18.12.2009. Zurich.

B.3 Timetables

SBB AG. 152 - Olten - Däniken - Aarau - Brugg, in *Official Timetable*, Timetable period 2016 (13.12.2015-10.12.2016), version 27.11.2015, available at http://www.fahrplanfelder.ch/en/archives/graphic-timetables.html, accessed 02.02.2016.

SBB AG. 161 - Brugg AG - Altstetten - Hardbrücke - Zürich HB, in *Official Timetable*, Timetable period 2016 (13.12.2015-10.12.2016), version 23.11.2015, available at http://www.fahrplanfelder.ch/en/archives/graphic-timetables.html, accessed 02.02.2016.

SBB AG. 432 - Zofingen - Suhr - Lenzburg - Mellingen - Wettingen, in *Official Timetable*, Timetable period 2016 (13.12.2015-10.12.2016), version 27.11.2015, available at http://www.fahrplanfelder.ch/en/archives/graphic-timetables.html, accessed 02.02.2016.

SBB AG. 511 - Basel SBB - Stein-S. - Brugg, in *Official Timetable*, Timetable period 2016 (13.12.2015-10.12.2016), version 27.11.2015, available at http://www.fahrplanfelder.ch/en/archives/graphic-timetables.html., accessed 02.02.2016.

SBB AG. 521 - Brugg VL - Wohlen - Muri - Arth-Goldau, in *Official Timetable*, Timetable period 2016 (13.12.2015-10.12.2016), version 27.11.2015, available at http://www.fahrplanfelder.ch/en/archives/graphic-timetables.html,accessed 02.02.2016.

SBB AG. 602 - Baden - Koblenz - Waldshut, in *Official Timetable*, Timetable period 2016 (13.12.2015-10.12.2016), version 27.11.2015, available at http://www.fahrplanfelder.ch/en/archives/graphic-timetables.html, accessed 02.02.2016.

Appendix C

SBB data analysis for numerical experiments

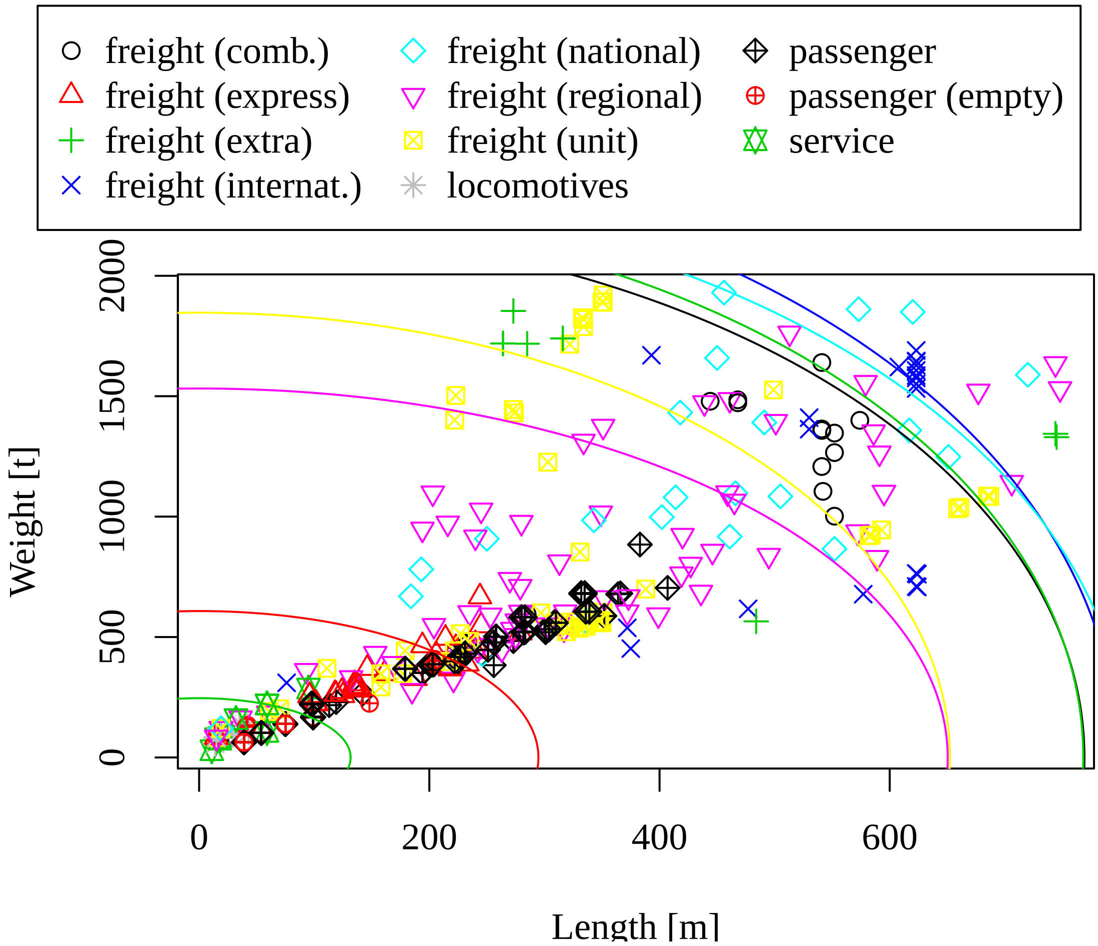
C.1 Rolling stock of freight trains

SBB provided data about 3030 train compositions for one week operations (20-27.03.2015) in the area surrounding Brugg AG. 365 observations concerned freight or service trains. Freight trains were differentiated according to R I-30111 complement (2015) in: combined transport, international, national, regional, express, extra and unit trains.

In order to simplify the implementation of the simulation environment, all trains in a class are supposed to use the same rolling stock. The rolling stock was selected to be the one corresponding to the closest to the 3rd quartiles of length and weight (lines in figure C.1).

Note that the selected rolling stock for extra freight trains (on-demand) is very similar to the one used for combined freight traffic (table C.1 and figure C.1). For this reason, extra freight trains will be represented by the rolling stock of the combined traffic. Express freight trains (e.g., mail trains) are comparable to passenger trains for which concerns weight and length. The only significant difference between the selected national train and the international one is the double traction. However, closer data analysis shows that also most national trains have only one locomotive. Thus, it was decided to utilise the chosen rolling material for international freight trains also for the national ones. Although regional freight trains and unit trains show similar lengths and weights, their breaking characteristics force to represent them with different rolling stock. As the characteristics of service trains do not differ much from simple locomotives and their operation is also very similar, these trains will be represented by locomotives.

Figure C.1: Choice of rolling stock for simulations



C.2 Delays

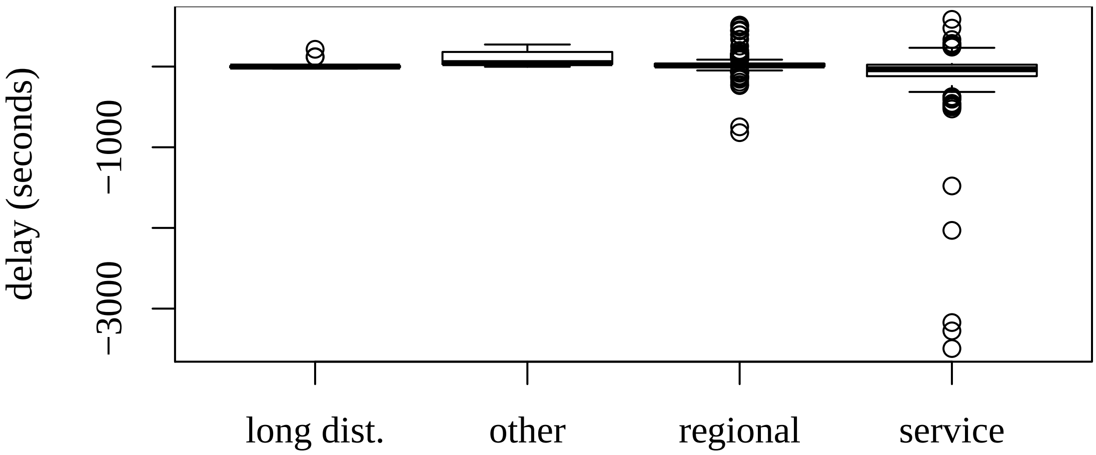
12’325 registered times at Brugg AG have been analysed to estimate the delay distributions to be used for the numerical experiments of chapter 5. Figure C.2 shows an overview. Note that departure and arrival delays at Brugg are only available for trains making commercial stops there. For the numerical experiments in chapter 5 it is assumed that initial delays are generated when trains enter the control area and when they depart from stations within the control area where they have stopped. This implicitly accounts for dwelling time variations and is fully compatible with the chosen simulation software. The probability densities of these delays are approximated from the recorded departure and passing delays at signals (all signals not only entrance) using function *fitdistr* from R software MASS package (Venables and Ripley, 2002). A Weibull distribution is assumed and the parameters found by Corman et al. (2011) are used as

Table C.1: Characteristics of freight trains. Bold ones are used as representatives for simulations.

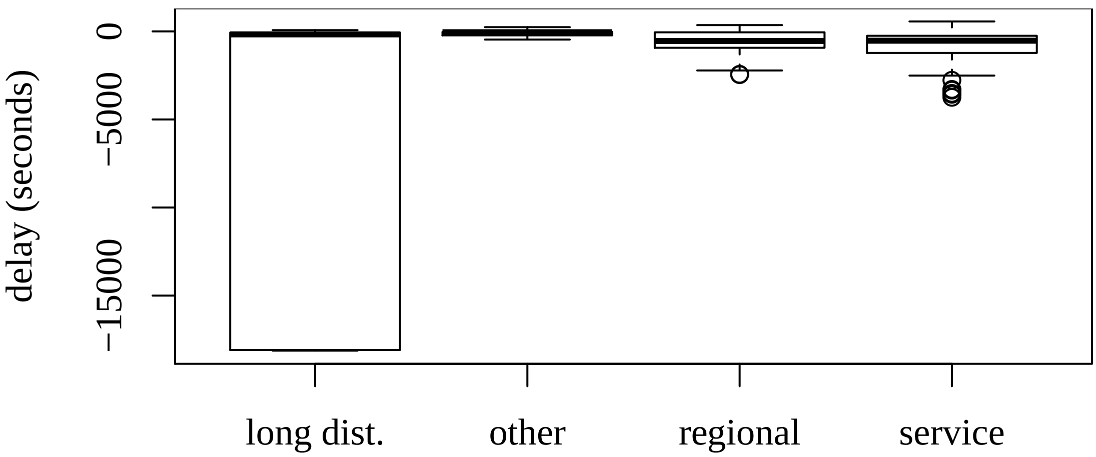
	reference	circulation $[DD.MM.YYYY]$	weight $[t]$	length $[m]$	breaking class	maximal speed $[km/h]$	cars	engines	engine axles
combined	42027	20.03.2015	1508	579	D 80	100	25	2	10
extra	54006	24.03.2015	1505	582	A 80	100	24	2	10
express	50293	24.03.2015	464	194	A 95	120	11	1	4
international	49064	21.03.2015	1551	623	A 70	120	39	1	4
national	60475	24.03.2015	1441	633	D 75	100	35	2	8
regional	62673	25.03.2015	901	526	D 85	100	31	2	8
unit	66422	24.03.2015	921	582	A 105	100	36	1	4
service	99726	22.03.2015	220	59	unknown	100	2	1	4

starting values of the approximation process. Figure C.3 shows the results.

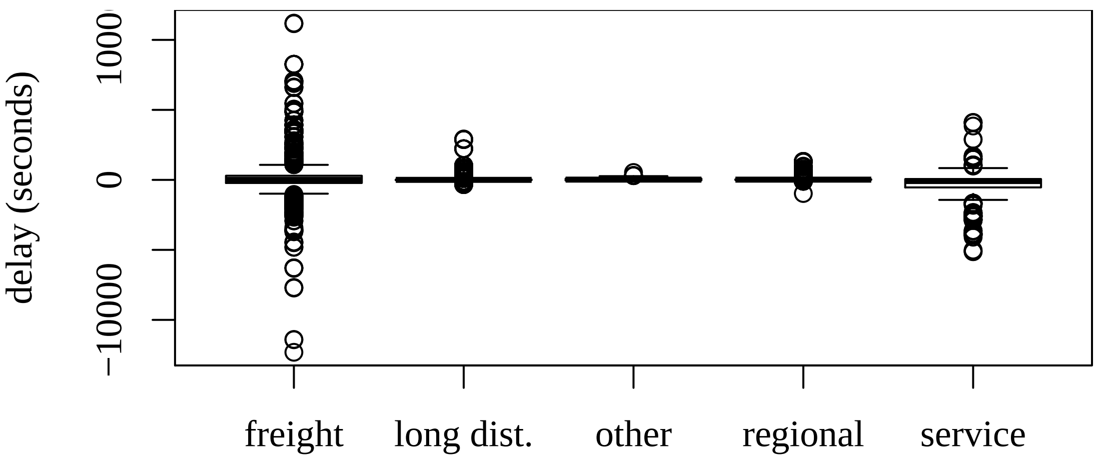
Figure C.2: Delays at Brugg (SBB)



(a) Departure delays

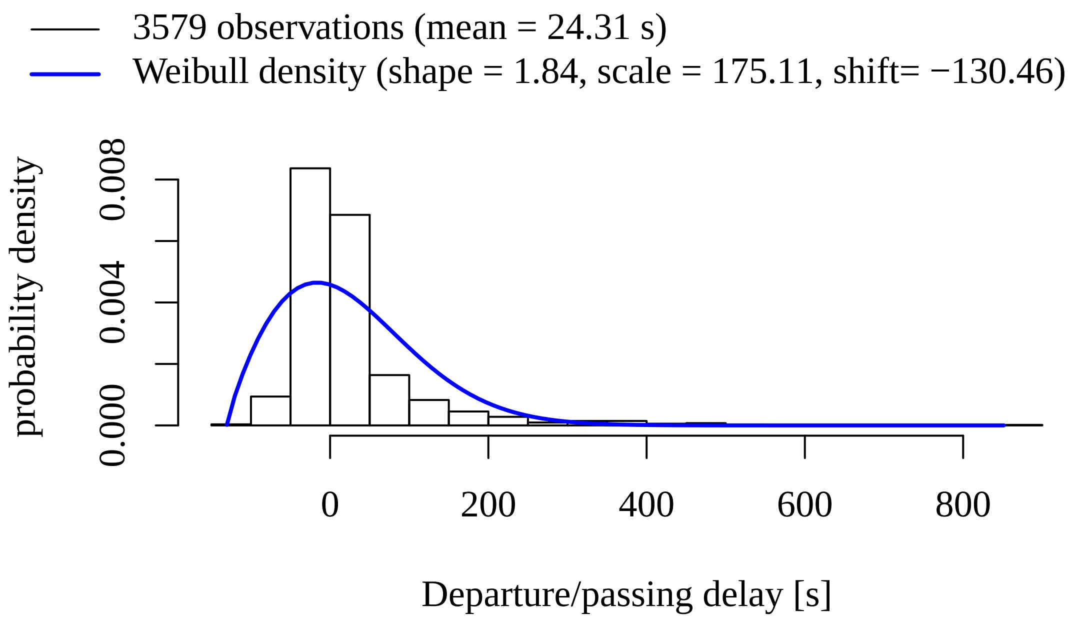


(b) Arrival delays

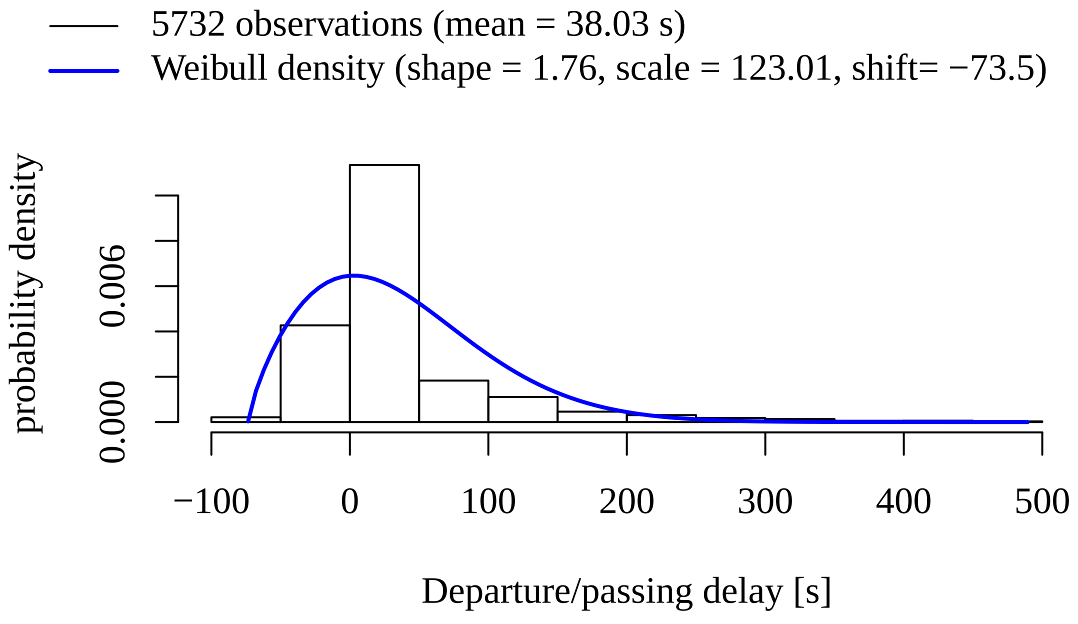


(c) Passing delays

Figure C.3: Initial delays at Brugg (SBB)

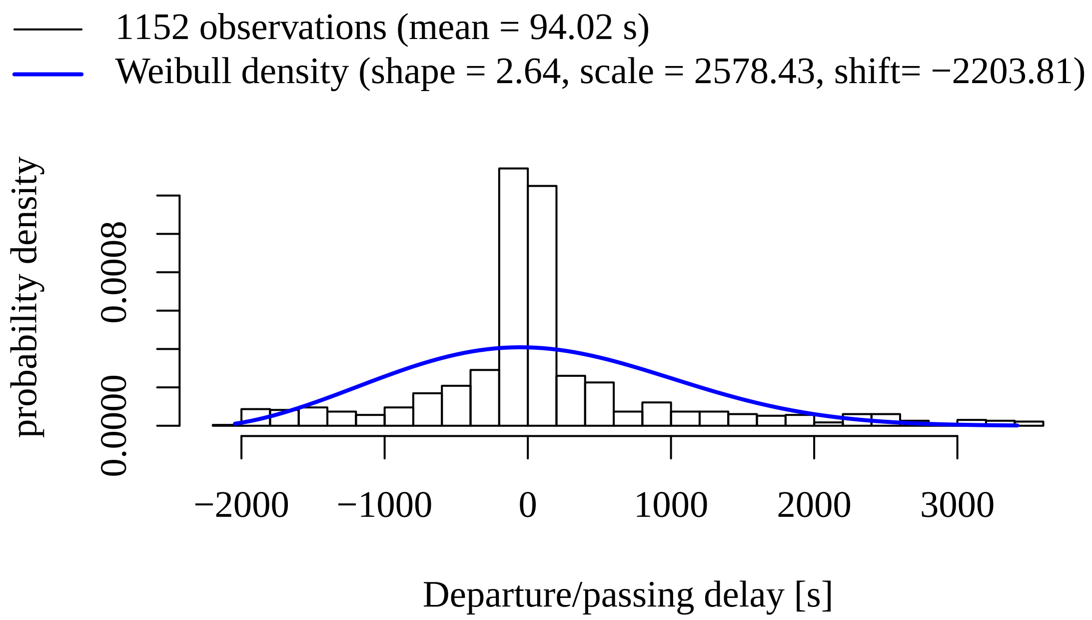


(a) long distance passenger trains

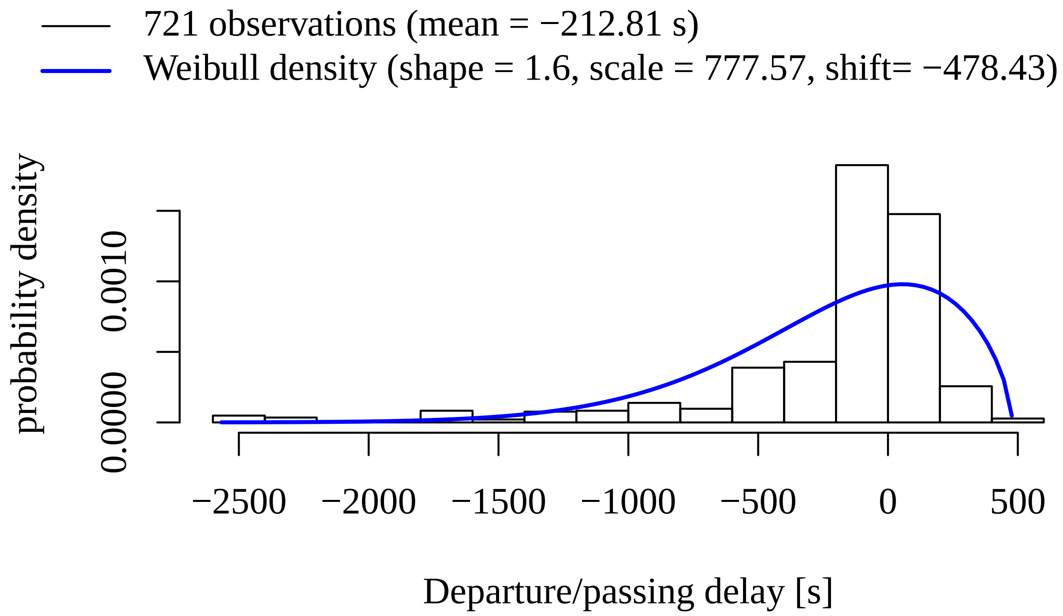


(b) regional passenger trains

Figure C.3: (continued) Initial delays at Brugg (SBB)



(c) freight trains



(d) service trains

Appendix D

Extended results of numerical experiments for coordinated rescheduling

Figure D.1: Intervals between arrival at and departure from portals. Different colors correspond to different coordination strategies. Delay set 1.

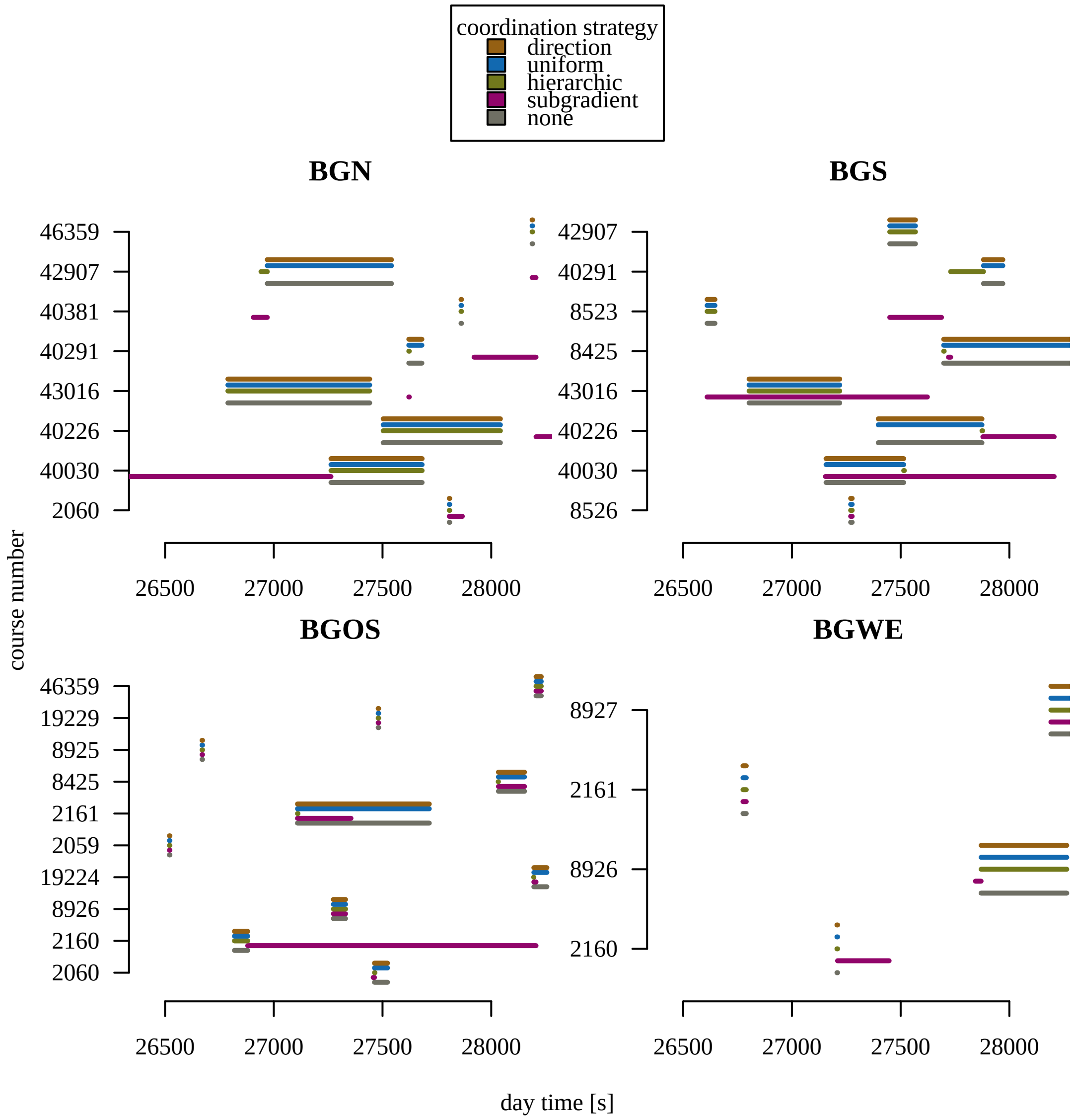


Figure D.2: Intervals between arrival at and departure from portals. Different colors correspond to different coordination strategies. Delay set 2.

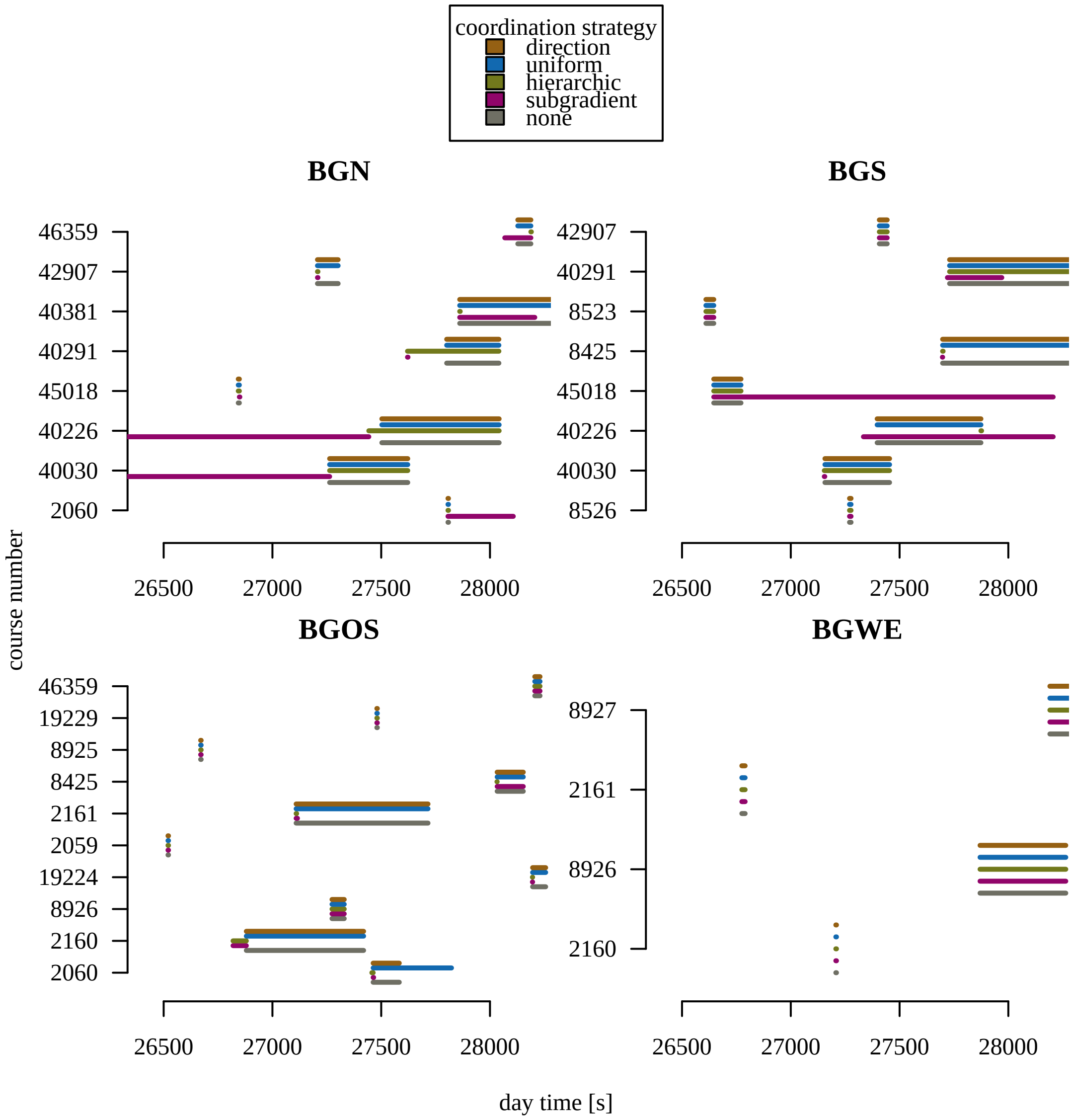


Figure D.3: Intervals between arrival at and departure from portals. Different colors correspond to different coordination strategies. Delay set 3.

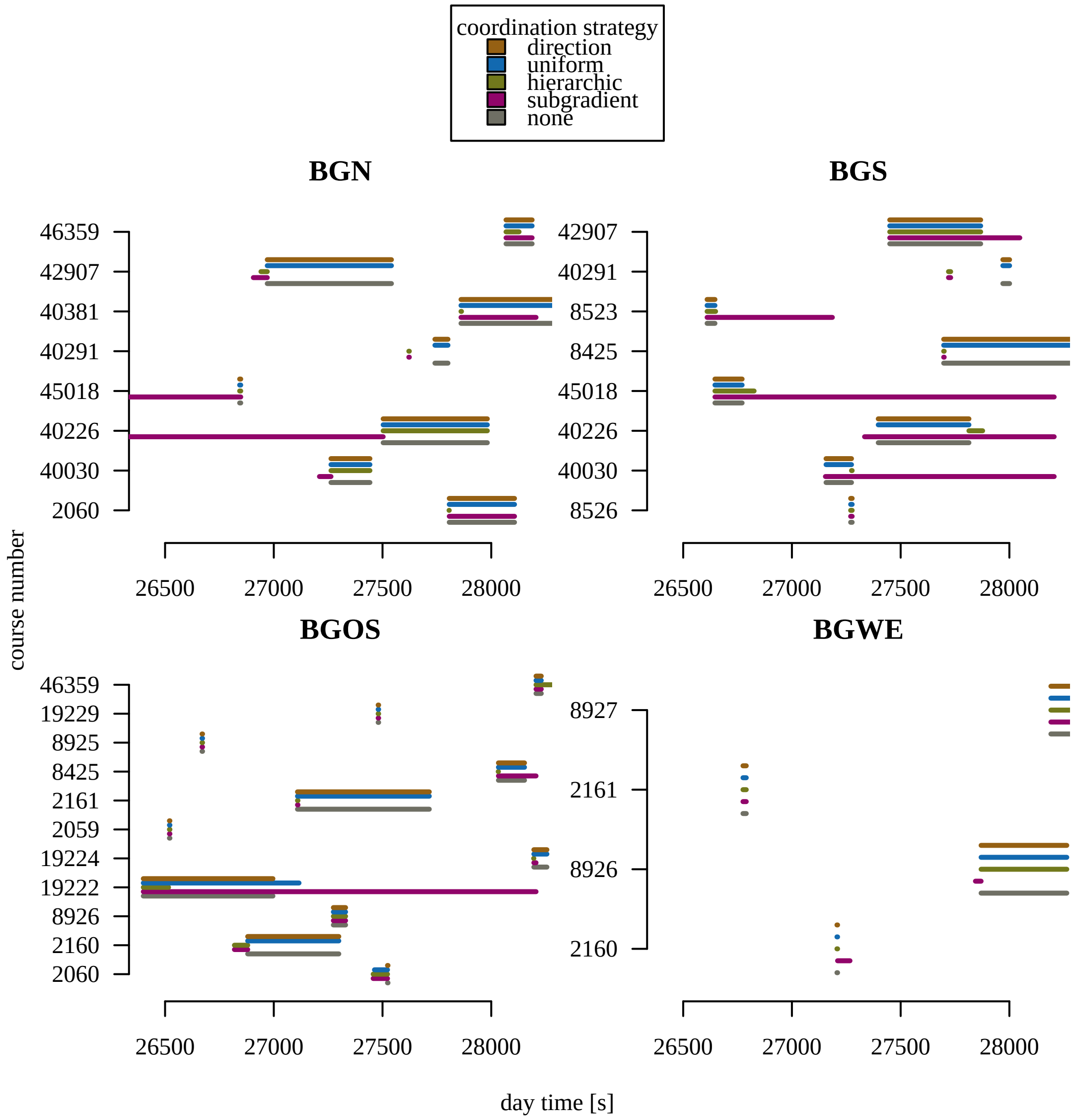


Figure D.4: Intervals between arrival at and departure from portals. Different colors correspond to different coordination strategies. Delay set 4.

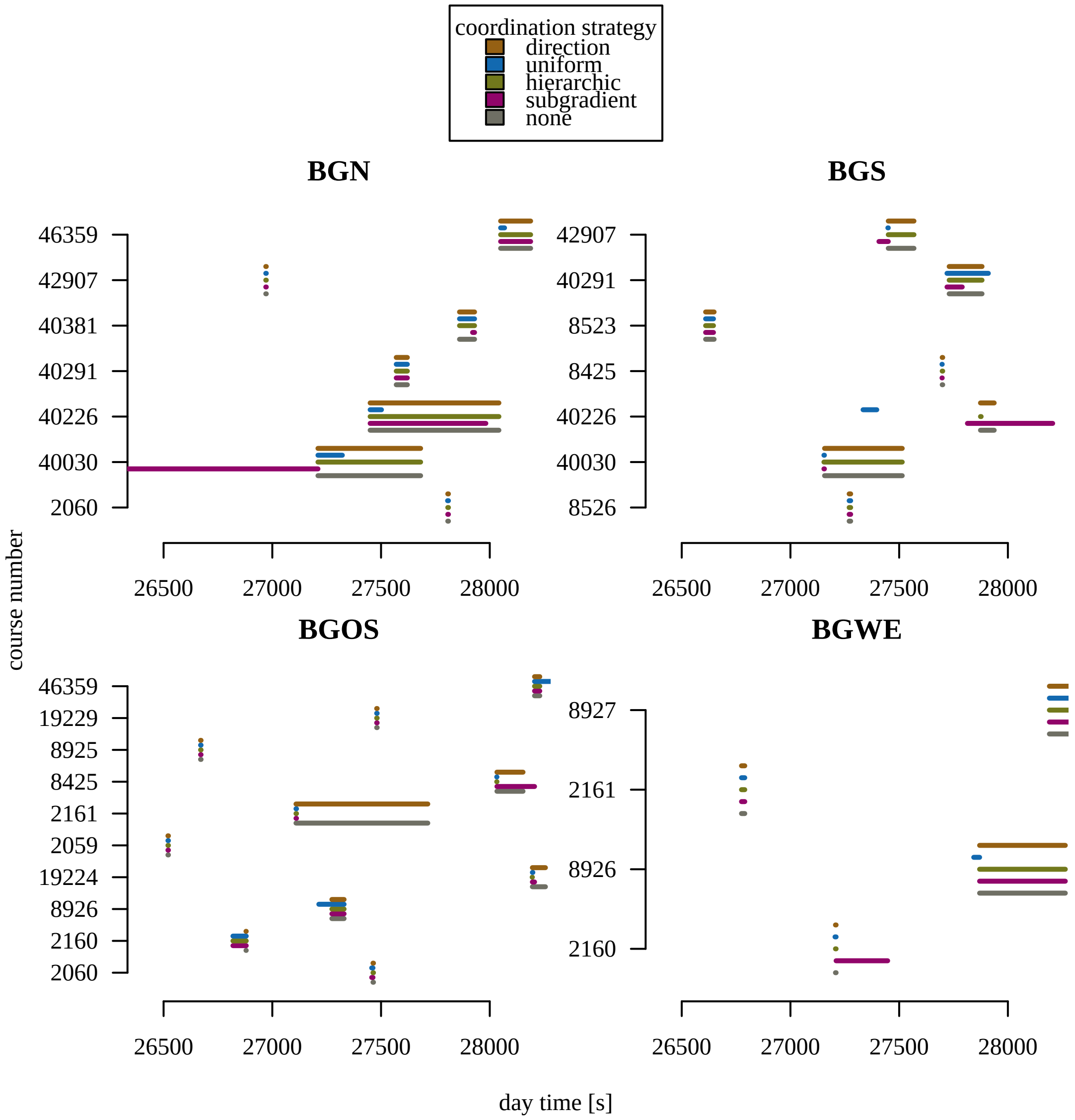
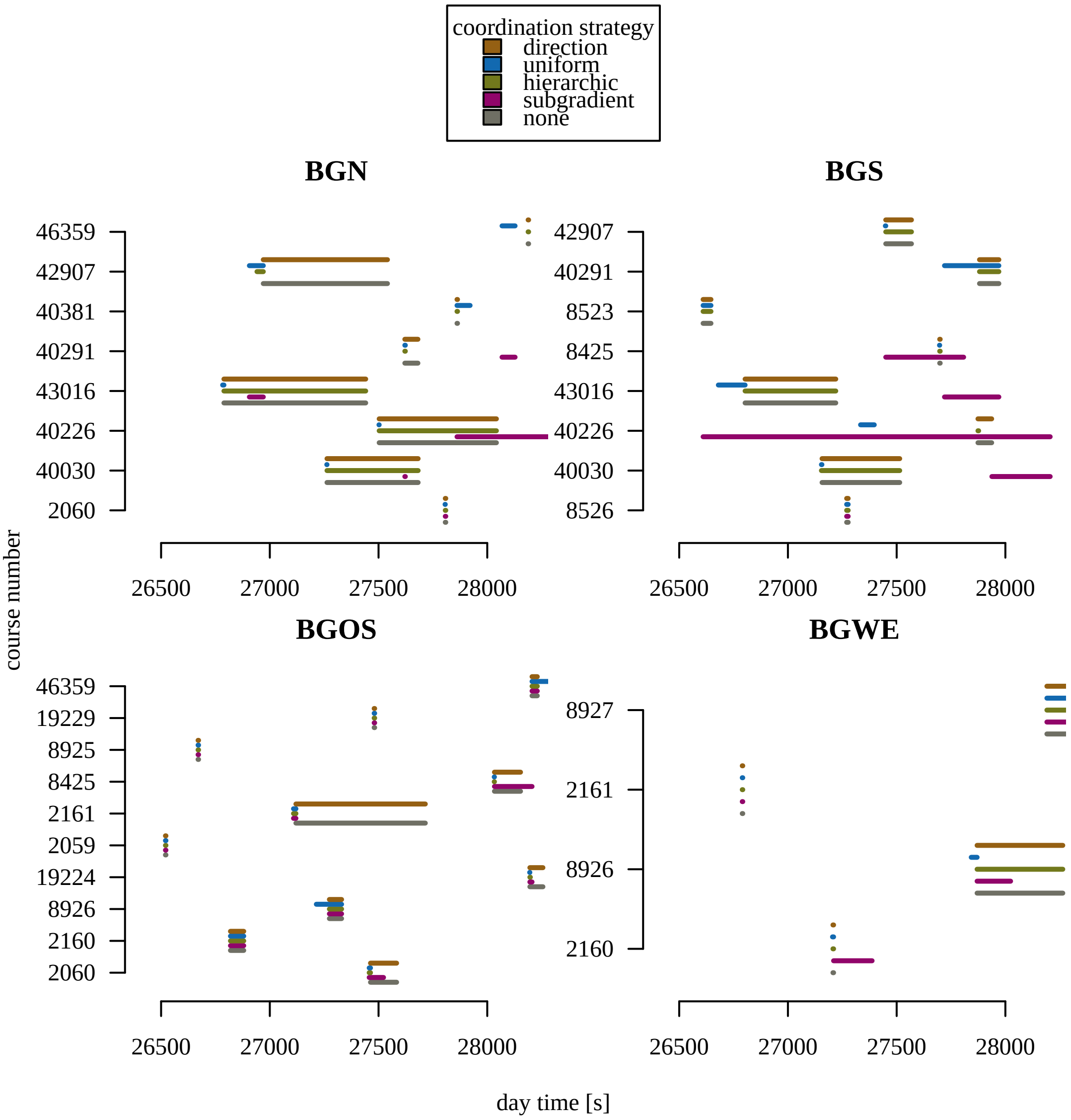


Figure D.5: Intervals between arrival at and departure from portals. Different colors correspond to different coordination strategies. Delay set 5.



Curriculum Vitae

AMBRA TOLETTI

Born:	June 7, 1990 in Locarno (TI), Switzerland
Citizenship:	Swiss
Civil Status:	Single

EDUCATION

Doctor of Sciences in Civil Engineering, 2018
Swiss Federal Institute of Technology (ETH), Zurich, Switzerland

Focus:	Railway operations and railway traffic rescheduling
Dissertation:	"Automated railway traffic rescheduling and customer information"
Advisors:	Prof. Dr. Ulrich A. Weidmann, Prof. Dr. Anita Schöbel (Universität Göttingen), Dr. Marco Laumanns (IBM Research), Dr. Andreas Dietsch (SBB)

Master of Science in Applied Mathematics, 2014
Swiss Federal Institute of Technology (ETH), Zurich, Switzerland

Focus:	Statistics
Application Field:	Transportation
Master thesis:	"Tree-based variational methods for parameter estimation, inference and denoising on Markov Random Fields" (Advisor: Prof. Dr. Sara Van de Geer)
Semester paper:	"Wie zufrieden sind Pendler? (in German)" (Advisors: Prof. Dr. Kay W. Axhausen, Prof. Dr. Werner Stahel, Veronika Killer)

Bachelor of Sciences in Mathematics, 2012
Swiss Federal Institute of Technology (ETH), Zurich, Switzerland

Bachelor thesis: "Modulated Fourier expansions for ODEs with oscillatory solutions" (Advisor: Prof. Dr. Ralf Hiptmair)

Swiss matura with scientific focus, 2009
Liceo di Lugano 1 (Lilu1), Lugano(TI), Switzerland

EXPERIENCE

Graduate Research (Full time), 05/2014 - 04/2018
Department of Civil Engineering, ETH Zurich, Switzerland

- Research on railway operations topics in collaboration with SBB and RBS
- Teaching support
- Organizing committee of international conference "IT15.rail"
- Railway Operations Laboratory team

Teaching Assistant (Part time), 09/2012 - 12/2013
Department of Mathematics, ETH, Zurich, Switzerland

- Lead of exercise classes and check of homework
- Preparation and lead of exam preparation classes

Schriftenreihe des IVT

Herausgegeben vom Institut für Verkehrsplanung und Transportsysteme der Eidgenössischen Technischen Hochschule ETH Zürich

Bis auf wenige Ausnahmen sind sämtliche Titel der Schriftenreihe des IVT über die Research Collection der ETH unter www.research-collection.ethz.ch frei zugänglich.

Nr.	Titel	Autor	Jahr	Bezug
59	Aufteilung von Erholungsaktivitäten im Raum und in der Zeit	A. Deloukas	1986	online
60	Baulich integrierte Strassen	M. Rotach, F. Hoppler, M. Burgherr, M. Grieder	1986	online
61	Unterhaltskosten von Trolley- und Dieselbussen in der Schweiz	H. Brändli, B. Albrecht, H. Müller, E. Schmid	1986	online
62	Eichung und Validation eines Umlegungsmodelles für den Strassengüterverkehr	C. Hidber, E. Meier	1986	online
63	Fahrpläne für die Zürcher S-Bahn	G. Rey	1986	online
64	Quergefälle in Geraden und Kurven	P. Spacek	1987	online
65	Simulation von Eisenbahnsystemen mit RWS-1	P. Giger	1987	online
66	Siedlung - Verkehrsangebot - Verkehrsnachfrage	M. Rotach, F. Hoppler, H. Bruderer, M. Mötteli	1987	online
67	N 13, Au–Haag: Auswirkungen der Sofortmassnahmen vom Sommer 1984 auf das Unfallgeschehen	K. Dietrich, P. Spacek	1987	online
68	Entwicklung des Schweizerischen Personenverkehrs 1960–1990	C. Hidber	1987	online
69	MacTrac – interaktives Programm für Zuglaufrechnungen Benutzerhandbuch	P. Brunner	1988	online
70	Mehrdimensionale Bewertungsverfahren und UVP im Verkehr	C. Hidber u.a.	1988	online
71	Ein Beitrag zur Umlegung: Ausgewählte Probleme und Lösungsansätze	C. Hidber, M. Keller	1988	online
72	Flexible Betriebsweise: Die Kombination von Linien- und Bedarfsbetrieb auf einer Buslinie	H. Brändli, B. Albrecht, K. Bareiss	1988	online
73	Von der Bahn 2000 zum System OeV 2000	H. Brändli, B. Albrecht, W. Glünkin	1988	online
74	Planung des öffentlichen Verkehrs in nichtstädtischen Gebieten	H. Brändli, H. Amacker	1988	online
75	Simulation of railway networks with RWS-1	P. Giger	1989	online
76	Einfluss des Mischprozesses auf die Qualität bituminöser Mischungen	M. Kronig	1989	online
77	Regionale Arbeitsmobilität	W. Dietrich	1989	online
78	Zur Bewertung der Wirkung sicherheitsorientierter Massnahmen im Eisenbahnbetrieb	R. Röttinger	1989	online
79	Bewertung der offiziellen NEAT-Varianten	W. Schurter, N. Bischofberger	1989	online
80	DQM-2: Ein Gerät zur dynamischen Querprofilmessung auf Strassen	U. Scheifele	1989	online
81	Neuverkehr infolge Ausbau und Veränderung des Verkehrssystems	E. Meier	1989	online
82	Entwicklung von Verhaltensmodellen als Grundlage eines programmierten Erhaltungskonzeptes Teil 1: Modelle für bleibende Verformungen	J.-D. Zufferey	1989	online
83	Moderne EDV-Anwendungen zur Verkehrsbeeinflussung	C. Hidber, W. Schurter	1989	online
84	Berufspendlerverkehr 1950–1990 Entwicklung des Berufspendlerverkehrs der schweizerischen Agglomerationen	C. Hidber, N. Bischofberger	1989	online
85	Drainasphalt Beobachtungen des Verhaltens von hohlraumreichen Verschleissschichten unter Verkehr	H. Köster	1990	online

Nr.	Titel	Autor	Jahr	Bezug
86	Güterverkehrsaufkommen in Industriegebieten	P. Schirato, C. Hidber	1991	online
87	Langzeitverhalten von bituminösen Drainbelägen Teil 1: Lärmverhalten von Drainbelägen	T. Isenring	1991	online
88	EDV-Anwendungen im Verkehrswesen	C. Hidber, W. Schurter	1991	online
89	Sichtweiten	F. Bühlmann, P. Spacek, H. P. Lindenmann,	1991	online
90	Transporttechnik der Fussgänger	U. Weidmann	1992	online
91	Optimierung in Verkehrsplanung, Transporttechnik und Logistik	C. Hidber u.a.	1992	online
92	Elemente eines computergestützten Werkzeugs zur Entwicklung von Eisenbahnsicherungsanlagen mit Petri-Netzen	M. Montigel	1993	online
93	Verkehrsangebot Schweiz 1960–1992	C. Hidber, N. Bischofberger	1992	online
94	Simulationsmodell für Tramnetze	P. Brunner	1993	online
95	Desserte ferroviaire de l'aeroport de Geneve-Cointrin	C. Hidber, G. Abay, J.-P. Widmer	1993	online
96	Kostenproblematik des Schienenverkehrs: Ansätze zur Reduktion der Produktionskosten	H. Brändli, J. Wichser	1993	online
97	Think Trac: ein einfach zu portierendes Traktionsprogramm für die Berechnung von Fahrzeiten im Eisenbahnverkehr	H. Brändli, J. Hoessly	1993	online
98	Pioniere des Verkehrs. Eine Auswahl von Kurzbiographien zur Einführung in die Verkehrsgeschichte	C. Hidber u.a.	1993	online
99	Der Fahrgastwechsel im öffentlichen Personenverkehr.	U. Weidmann	1994	online
100/1	Optimierung des Oberbaus bei Meterspurbahnen: Teil 1: Berechnung und Beurteilung	H. Brändli, J. Wichser, S. Rangosch, M. Kohler	1994	online
100/2	Optimierung des Oberbaus bei Meterspurbahnen: Teil 2: Grundlagen und Methodik	H. Brändli, J. Wichser, S. Rangosch	1994	online
100/3	Optimierung des Oberbaus bei Meterspurbahnen: Teil 3: Literaturkatalog mit Kommentar	H. Brändli, J. Wichser, S. Rangosch	1994	online
101	Zur Gestaltungsaufgabe des Bauingenieurs: Systemtheoretische Grundlagen und Folgerungen für Planung und Ausführung ein Beitrag zur Ingenieurwissenschaft	B. Meyer	1994	online
102	Modellierung und Gewährleistung von Abhängigkeiten in Eisenbahnsicherungsanlagen	M. Montigel	1994	online
103	Simulation von Eisenbahnsystemen mit RWS-1: 3. Auflage RWS Version 1.7 für Apple Macintosh	P. Giger	1994	online
104	Nationalstrasse N2, Basel–Chiasso Kapazitätsuntersuchung: Verkehrstechnische Studie zur Beurteilung der heutigen Kapazitätsverhältnisse	K. Dietrich, P. Spacek	1994	online
105	Anwendungsbeispiele zur Optimierung in Verkehrsplanung, Transporttechnik und Logistik	C. Hidber, Z. Oblozinska	1994	online
106	Grundlagen zur Berechnung der Fahrgastwechselzeit	U. Weidmann	1995	online
107	Umweltbilanz der Warenverteilung (Non Food) des Migros-Genossenschafts-Bundes	C. Hidber, E. Meier, U. Koller	1995	online
108	Lagestabilität lückenloser Meterspurgleise in kleinen Bogenradien	S. Rangosch	1995	online
109	Pioniere des Verkehrs. Eine Auswahl von Kurzbiographien zur Einführung in die Verkehrsgeschichte. Band 2.	C. Hidber u.a.	1995	online
110	Qualitätsmanagement von Eisenbahnstrecken.	E. Hediger	1996	online
111	Abschätzung des Zonen-Binnenverkehrs in Städten; Teil1	C. Hidber, J.-P. Widmer, J.-L. Frossard, G. Schwartz, J. Stintzing, J. van Dijk	1996	online

Nr.	Titel	Autor	Jahr	Bezug
112	Vergleich: Schotterloser Oberbau / Schotteroberbau	H. Brändli, P. Pingoud	1997	online
113	Leistungsfähigkeit von Verkehrssystemen	N. Bischofberger	1997	online
114	Do rail stations at airports allow a better distribution of air passenger transport demand among airports	R. Schilling, J.-P. Widmer	1997	online
115	Korridor-Leistungsfähigkeit - Zusammenwirken mehrerer Verkehrsträger bei Überlastung	C. Hidber, N. Bischofberger, J. Greutert, Z. Oblozinska, C. de Rahm	1997	online
116/1	Bahnerschliessung Flughäfen Stuttgart und Frankfurt a. M. Kosten-Nutzen-Analyse	J.-P. Widmer, R. Schilling, R. Gottwald	1997	online
116/2	Bahnerschliessung Flughäfen Brüssel. Kosten-Nutzen-Analyse	J.-P. Widmer, O. Hintermeister	1997	online
116/3	Bahnerschliessung Flughäfen Paris Roissy-CDG2 und Orly. Kosten-Nutzen-Analyse	J.-P. Widmer, C. Dasen-Sender	1997	online
116/4	Bahnerschliessung Flughafen Zürich. Kosten-Nutzen-Analyse	J.-P. Widmer, R. Gottwald	1998	online
117	Berufspendlerverkehr 1980–1990–2000	C. Hidber, C. Dasen-Sender, N. Bischofberger	1997	online
118	25 Jahre IVT-Messungen zum Verkehrsablauf auf Autobahnen	K. Dietrich, H. P. Lindenmann, Y. Chabot-Zhang	1998	online
119	Qualitätsfaktor – Vergleich der Angebotsqualität im Regionalverkehr	U. Widmer, M. Neumeister	1998	online
120	Einsatz von zementstabilisiertem Asphaltgranulat in Fundationsschichten	M. Shojaati	1998	online
121	Carpools im Spannungsfeld mit dem öffentlichen Personenverkehr	S. Dasen	1999	online
122	Bonus/Malus-System; System zur Gewährleistung der Betriebsqualität auf dem für den freien Zugang geöffneten SBB-Netz.	M. Hofer	1999	online
123	Beiträge zur Soziologie und Politologie im Verkehr	C. Hidber u.a.	1999	online
124	Stabilitätsprobleme lückenloser Meterspurgleise in engen Radien und in Uebergangsbögen	F. Gallati	2001	online
125	Objektorientierte Modellierung von Infrastrukturelementen und Betriebsvorgängen im Eisenbahnwesen	D. Hürlimann	2002	online
126	Der Bettungsmodul für den Schotteroberbau von Meterspurbahnen	M. Kohler	2002	online
127	Verwendung von Eisenbahnbetriebsdaten für die Schwachstellen- und Risikoanalyse zur Verbesserung der Angebots- und Betriebsqualität	M. Ullius	2005	online
128	Netzgestaltungsgrundsätze für den öffentlichen Personennahverkehr in Verdichtungsräumen	U. Schäffeler	2005	online
129	GIS-basiertes Konzept zur Modellierung von Einzugsbereichen auf Bahn-Haltestellen	J. Jermann	2005	online
130	Fachbegriffe des öffentlichen Verkehrs	J. Wichser, H. Schneebeli, S. Bollinger	2005	online
131	Regionalisierung des Schienenverkehrs in der Schweiz	M. Rieder	2005	online
132	Parameters of pedestrians, pedestrian traffic and walking facilities	S. Buchmüller, U. Weidmann	2006	online
133	Strategies for increasing intermodal transport between Eastern and Western Europe	J. Wichser, S. Besters, B. Bojanic, S. Bolliger, N. Fries	2006	online
134	Europäische Marktstudie für das System Swissmetro	U. Weidmann, S. Buchmüller, M. Rieder, A. Nash, A. Erath	2006	online
135	Studie zu einem neuen schweizerischen Trassepreissystem	U. Weidmann, H. Schneebeli, N. Fries, J. Wichser, P. Schmidt,	2007	online
136	Regionen im Umbruch! – Regionalverkehr im Aufbruch	Tagungsband	2007	online

Nr.	Titel	Autor	Jahr	Bezug
137	Systemvorschlag für ein neues schweizerisches Trassenpreissystem	U. Weidmann, J. Wichser, P. Schmidt	2008	online
138	Betriebsstabilität bei Buslinien mit Fahrausweisverkauf durch Fahrer	U. Weidmann, M. Lüthi, S. Buchmüller, R. Dorbritz	2008	online
139	Leistungsfähigkeitsbestimmung öffentlicher Verkehrssysteme	U. Weidmann, G. Anderhub, R. Dorbritz	2008	online
140	Peripherer Verkehr – Verkehrte Peripherie?	Tagungsband	2008	online
141	Mobilitätsplan Hochschulgebiet Zürich	U. Weidmann, P. Spacek, K. W. Axhausen, B. Alt, G. Anderhub, R. Dorbritz, A. Frei, M. Laube, M. Scherer, C. Weis	2008	online
142	Gesamterschliessungskonzept Science City Synthesebericht	U. Weidmann, P. Frank	2008	online
143	Ein Trassenpreissystem aus Umweltsicht unter besonderem Augenmerk des Lärms	U. Weidmann, St. Moll, P. Schmidt	2009	online
144	Validierung eines semi-quantitativen Ansatzes zur Risikobeurteilung in der Eisenbahntechnik	S.-L. Bepperling	2009	online
145	Verlässliche Finanzierung des öffentlichen Verkehrs in der Schweiz Konzeptstudie	U. Weidmann, J. Wichser	2009	online
146	Die Revolution der Automation – Verkehrsautomatisierung und Gesellschaft im 20. und 21. Jahrhundert	Tagungsband	2009	online
147	Improving the efficiency of heavily used railway networks through integated real-time rescheduling	M. Lüthi	2009	online
148	Market potential and value of sustainable freight transport chains	N. Fries	2009	online
149	100 Jahre Gesellschaft der Ingenieure des öffentlichen Verkehrs 1910–2010, (k)ein Wunder, dass es uns noch gibt	Gesellschaft der Ingenieure des öffentlichen Verkehrs	2009	online
150	Investigation of space-time structures in public transport networks and their optimisation	B. Alt	2010	online
151	Nachhaltige Güterfeinverteilung	U. Weidmann, N. Fries, B. Bopp, W. Stölzle, E. Hofmann, K. Gebert	2010	online
152	Ein lärmabhängiges Trassenpreissystem für die Schweiz	U. Weidmann, St. Moll	2010	online
153	Laterales Fahrverhalten	G. Santel	2011	online
154	Stabil Mobil – Komplexe Verkehrssysteme als Herausforderung unserer Gesellschaft	Tagungsband	2011	online
155	Methodology for assessing the structural and operational robustness of railway networks	R. Dorbritz	2012	online
156	Operational stability and reliability of urban bus routes in Zurich, Switzerland	U. Weidmann, N. Carrasco, O. Fink	2012	online
157	Metro/tram – Eine Perspektive für Zürich	U. Weidmann, W. Huber	2012	online
158	Optimale Netznutzung und Wirksamkeit der Instrumente zu deren Lenkung	U. Weidmann, P. Frank, T. Fumasoli, St. Moll	2012	online
159	Europäische Eisenbahnregulierung im Wandel: Organisationsformen und Bestimmungsgrössen vom 19. bis zum 21. Jahrhundert	U. Weidmann, M. Rieder	2012	online
160	Productivity improvements for freight railways through collaborative transport planning	St. Moll	2012	online
161	Concept des transports et de mobilité dans des communes de Clos du Doubs et Soubey	U. Weidmann, M. Rieder, S. Herrigel-Wiedersheim	2015	online

Nr.	Titel	Autor	Jahr	Bezug
162	Differences in cognition of public transport systems: Image and behavior towards urban public transport	M. Scherer	2013	online
163	Methodik zur Effizienzbeurteilung der Kapazitätsnutzung und -entwicklung von Bahnnetzen	P. Frank	2013	online
164	Prognose der Dauer von Störungen des Bahnbetriebs	St. Schranil	2013	online
165	Failure and degradation prediction by artificial neural networks: Applications to railway systems	O. Fink	2014	auf Anfrage via IVT Sekretariat (CHF 50.-)
166	Das Verhalten von lückenlos verschweissten Gleisen (LVG) in engen Radien der Meterspur	B. Bopp	2014	online
167	How international borders affect local public transport: Analyses and evaluations of cross-border agglomerations in Switzerland, France and Germany	E. Barth	2014	online
168	Spurverhalten von Motorfahrzeugen in Kurven: Zusammenhänge zwischen Spurtypen, Unfallgeschehen und Kurvengeometrie	F. Baumgartner	2015	online
169	Planning, operation, and the influence of national culture in public transport service reliability	N. Carrasco	2015	online
170	Algorithmic decision support for the construction of periodic railway timetables	S. Herrigel-Wiedersheim	2015	online
171	Generische Beschreibung von Eisenbahnbetriebsprozessen	S. Höppner	2015	online
172	Holistic rail network operation by integration of train automation and traffic management	X. Rao	2015	online
173	Innovationen im Bahnsystem	U. Weidmann, St. Schranil, D. Bruckmann, T. Fumasoli, S. Herrigel-Wiedersheim	2015	online
174	Stand des Wissens zu Schweizer Tramsystemen im Mischverkehr Forschungsprojekt im Rahmen der COST Action TU1103	U. Weidmann, N. Carrasco, J. Kupferschmid, C. Marti, M. Schwertner	2015	online
175	Impacts of non-Aeronautical Development at Airports on Public Transport	H. Orth	2016	online
176	Measuring Service Quality in Freight Transport Networks	A. Mancera Sugrañes	2018	online
177	Energetischer Systemvergleich von Diesel-, Hybrid- und Elektrobussen	M. Schwertner	2017	online
178	Estimations of the Potential and the Impact of Rail Freight in Urban Areas	T. Fumasoli	2018	online
179	Automated railway traffic rescheduling and customer information	A. Toletti	2018	online
180	Quantitative assessment of public transport and built environment integration at the neighborhood scale (vorläufiger Titel)	C. Marti	2018	erscheint in Kürze
181	Forschungsbericht Rahmensteifigkeitsmessungen	P. Braess, M. Zimmermann, U. Weidmann	2018	online
182	Sicherstellung einer langfristig guten Gleislage in atmenden Bögen	P. Braess	2018	online
183	New generic approach to the pedestrian fundamental diagram (vorläufiger Titel)	E. Bosina	2018	erscheint in Kürze
184	Consequences of Automated Transport Systems as Feeder Services to Rail (project funded by the SBB Fund for Research into Management in the Field of Transport)	M. Sinner, P. Khaligh, U Weidmann	2018	online

Schriftenreihe wird fortgeführt

Bestellung nicht online verfügbarer Titel

Nr. Titel Anzahl

Meine Anschrift
Firma

Name

Adresse

PLZ/Ort Land

Telefon Fax

E-Mail

Lieferung erfolgt gegen Rechnung

Bestellungen sind zu richten an:
Sekretariat IVT, Institut für Verkehrsplanung und Transportsysteme
Eidgenössische Technische Hochschule ETH Zürich
Stefano-Franscini-Platz 5, HIL F 36.2
CH-8093 Zürich
Fax: +41 44 633 10 57
www.ivt.ethz.ch