Railway Track Allocation: Models and Algorithms

vorgelegt von
Dipl.-Math. oec. Thomas Schlechte
aus Halle an der Saale

Von der Fakultät II – Mathematik und Naturwissenschaften
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften
– Dr. rer. nat. –

genehmigte Dissertation

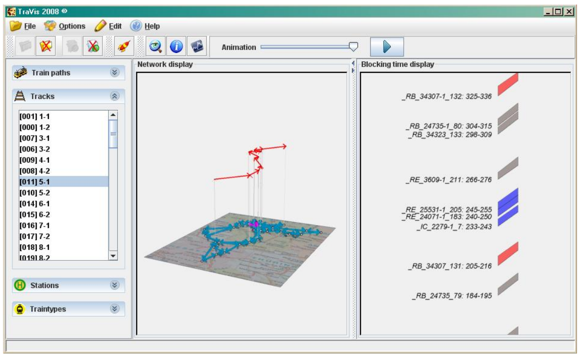
Promotionsausschuss

Berichter:	Prof. Dr. Dr. h. c. mult. Martin Grötschel
	PD Dr. habil. Ralf Borndörfer
Vorsitzender:	Prof. Dr. Fredi Tröltzsch

Tag der wissenschaftlichen Aussprache: 21.12.2011

Berlin 2012
D 83

Railway Track Allocation: Models and Algorithms



Thomas Schlechte

Preface

The "heart" of a railway system is the timetable. Each railway operator has to decide on the timetable to offer and on the rolling stock to operate the trips of the trains. For the railway infrastructure manager the picture is slightly different – trains have to be allocated to railway tracks and times, called *slots* such that all passenger and freight transport operators are satisfied and all train movements can be carried out safely. This problem is called the *track allocation problem*. My thesis deals with integer programming models and algorithmic solution methods for the track allocation problem in real world railway systems.

My work on this topic has been initiated and motivated by the interdisciplinary research project "railway slot allocation" or in German "Trassenbörse".[1] This project investigated the question whether a competitive marketing of a railway infrastructure can be achieved using an auction-based allocation of railway slots. The idea is that competing train operating companies (TOCs) can bid for any imaginable use of the infrastructure. Possible conflicts will be resolved in favor of the party with the higher willingness to pay, which leads directly to the question of finding revenue maximal track allocations. Moreover a fair and transparent mechanism "cries" out for exact optimization approaches, because otherwise the resulting allocation is hardly acceptable and applicable in practice. This leads to challenging questions in economics, railway engineering, and mathematical optimization. In particular, developing models that build a bridge between the abstract world of mathematics and the technical world of railway operations was an exciting task.

I worked on the "Trassenbörse" project with partners from different areas, namely, on economic problems with the Workgroup for Economic and Infrastructure Policy (WIP) at the Technical University of Berlin (TU Berlin), on railway aspects with the Chair of Track and Railway Operations (SFWBB) at TU Berlin, the Institute of Transport, Railway Construction and Operation (IVE) at the Leibniz Universität Hannover, and the Management Consultants Ilgmann Miethner Partner (IMP).

[1]This project was funded by the Federal Ministry of Education and Research (BMBF), Grant number 19M2019 and the Federal Ministry of Economics and Technology (BMWi), Grant number 19M4031A and Grant number 19M7015B.

This thesis is written from the common perspective of all persons I worked closely with, especially the project heads Ralf Borndörfer and Martin Grötschel, project partners Gottfried Ilgmann and Klemens Polatschek, and the ZIB colleagues Berkan Erol, Elmar Swarat, and Steffen Weider.

The highlight of the project was a cooperation with the Schweizerische Bundesbahnen (SBB) on optimizing the cargo traffic through the Simplon tunnel, one of the major transit routes in the Alps. This real world application was challenging in many ways. It provides the opportunity to verify the usefulness of our methods and algorithms by computing high quality solutions in a fully automatic way.

The material covered in this thesis has been presented at several international conferences, e.g., European Conference on Operational Research (EURO 2009, 2010), Conference on Transportation Scheduling and Disruption Handling, Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and System (ATMOS 2007, 2010), International Seminar on Railway Operations Modeling and Analysis (ISROR 2007, 2009, 2011), Symposium on Operations Research (OR 2005, 2006, 2007, 2008), International Conference on Computer System Design and Operation in the Railway and other Transit Systems (COMPRAIL), International Conference on Multiple Criteria Decision Making (MCDM), World Conference on Transport Research (WCTR). Significant parts have already been published in various refereed conference proceedings and journals:

- ▷ Borndörfer et al. (2006) [34],
- ▷ Borndörfer et al. (2005) [33],
- ▷ Borndörfer & Schlechte (2007) [31],
- ▷ Borndörfer & Schlechte (2007) [30],
- ▷ Erol et al. (2008) [85],
- ▷ Schlechte & Borndörfer (2008) [188],
- ▷ Borndörfer, Mura & Schlechte (2009) [40],
- ▷ Borndörfer, Erol & Schlechte (2009) [38],
- ▷ Schlechte & Tanner (2010) [189][3],
- ▷ Borndörfer, Schlechte & Weider (2010) [43],
- ▷ Schlechte et al. (2011) [190],[1]
- ▷ and Borndörfer et al. (2010) [42][2].

[1] accepted by Journal of Rail Transport Planning & Management.
[2] accepted by Annals of Operations Research.
[3] submitted to Research in Transportation Economics.

Research Goals and Contributions

The goal of the thesis is to solve real world track allocation problems by exact integer programming methods. In order to establish a fair and transparent railway slot allocation, exact optimization approaches are required, as well as accurate and reliable railway models. Integer programming based methods can provide excellent guarantees in practice. We successfully identified and tackled several tasks to achieve these ambitious goals:

1. applying a novel modeling approach to the track allocation problem called "configuration" models and providing a mathematical analysis of the associated polyhedron,

2. developing a sophisticated integer programming approach called "rapid branching" that highly utilizes the column generation technique and the bundle method to tackle large scale track allocation instances,

3. developing a Micro-Macro Transformation, i.e., a bottom-up aggregation, approach to railway models of different scale to produce a reliable macroscopic problem formulation of the track allocation problem,

4. providing a study comparing the proposed methodology to former approaches, and,

5. carrying out a comprehensive real world data study for the Simplon corridor in Switzerland of the "entire" optimal railway track allocation framework.

In addition, we present extensions to incorporate aspects of robustness and we provide an integration and empirical analysis of railway slot allocation in an auction based framework.

Thesis Structure

A rough outline of the thesis is shown in Figure 1. It follows the "solution cycle of applied mathematics". In a first step the real world problem is analyzed, then the track allocation problem is translated into a suitable mathematical model, then a method to solve the models in an efficient way is developed, followed by applying the developed methodology in practice to evaluate its performance. Finally, the loop is closed by re-translating the results back to the real world application and analyze them together with experts and practitioners.

Main concepts on planning problems in railway transportation are presented in Chapter I. Railway modeling and infrastructure capacity is the main topic of Chapter II. Chapter III focuses on the mathematical modeling and the solution of the track allocation problem. Finally, Chapter IV presents results for real world data as well as for ambitious hypothetical auctioning instances.

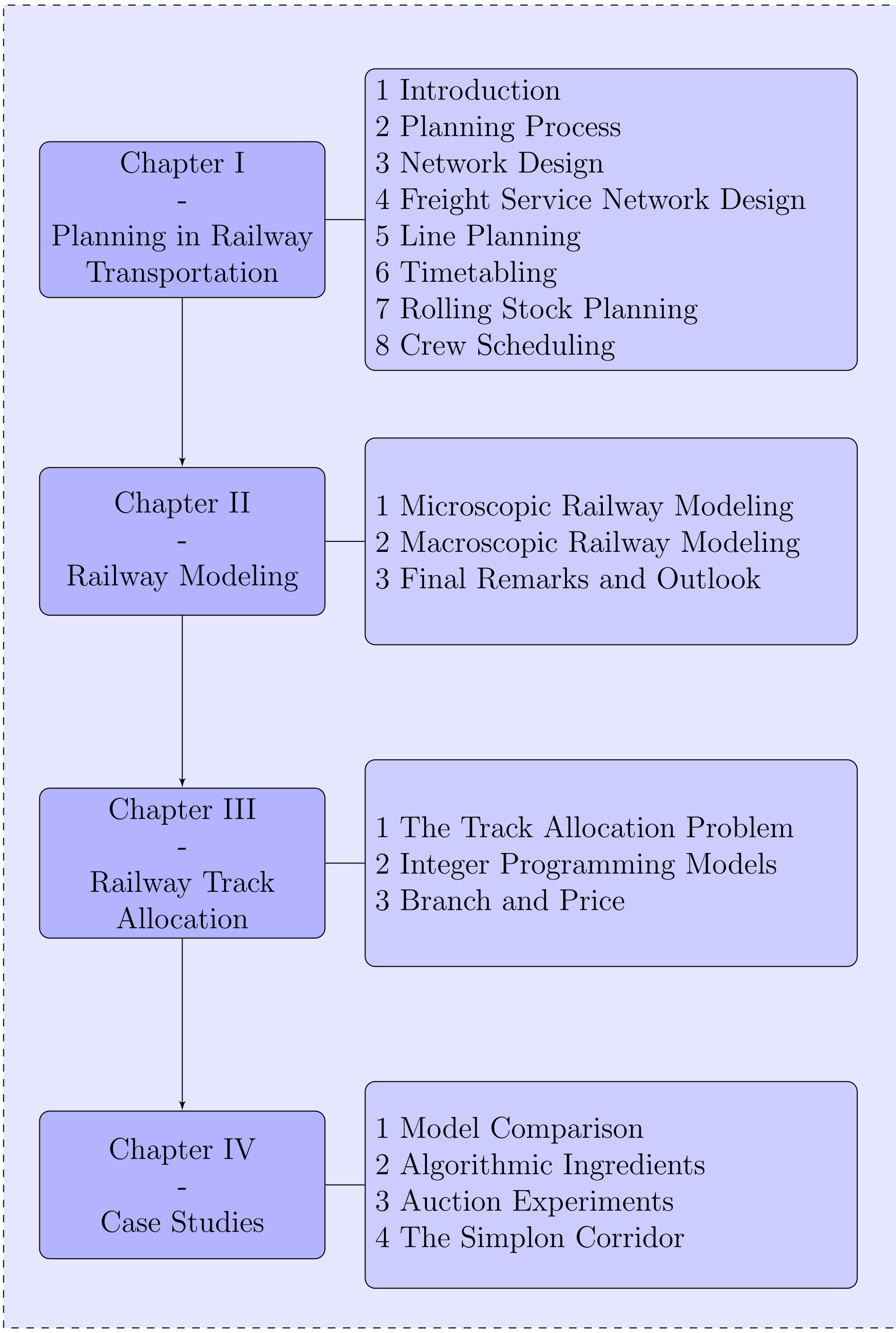


Figure 1: Structure of the thesis.

Abstract

This thesis is about mathematical optimization for the efficient use of railway infrastructure. We address the optimal allocation of the available railway track capacity – *the track allocation problem*. This track allocation problem is a major challenge for a railway company, independent of whether a free market, a private monopoly, or a public monopoly is given. Planning and operating railway transportation systems is extremely hard due to the combinatorial complexity of the underlying discrete optimization problems, the technical intricacies, and the immense sizes of the problem instances. Mathematical models and optimization techniques can result in huge gains for both railway customers and operators, e.g., in terms of cost reductions or service quality improvements. We tackle this challenge by developing novel mathematical models and associated innovative algorithmic solution methods for large scale instances. This allows us to produce for the first time reliable solutions for a real world instance, i.e., the *Simplon* corridor in Switzerland.

The opening chapter gives a comprehensive overview on railway planning problems. This provides insights into the regulatory and technical framework, it discusses the interaction of several planning steps, and identifies optimization potentials in railway transportation. The remainder of the thesis is comprised of two major parts.

The first part (Chapter II) is concerned with modeling railway systems to allow for resource and capacity analysis. Railway capacity has basically two dimensions, a space dimension which are the physical infrastructure elements as well as a time dimension that refers to the train movements, i.e., occupation or blocking times, on the physical infrastructure. Railway safety systems operate on the same principle all over the world. A train has to reserve infrastructure blocks for some time to pass through. Two trains reserving the same block of the infrastructure within the same point in time is called *block conflict*. Therefore, models for railway capacity involve the definition and calculation of reasonable running and associated reservation and blocking times to allow for a conflict free allocation.

There are *microscopic* models that describe the railway system extremely detailed and thorough. Microscopic models have the advantage

that the calculation of the running times and the energy consumption of the trains is very accurate. A major strength of microscopic models is that almost all technical details and local peculiarities are adjustable and are taken into account. We describe the railway system on a microscopic scale that covers the behavior of trains and the safety system completely and correctly. Those models of the railway infrastructure are already very large even for very small parts of the network. The reason is that all signals, incline changes, and switches around a railway station have to be modeled to allow for precise running time calculations of trains. In general microscopic models are used in simulation tools which are nowadays present at almost all railway companies all over the world. The most important field of application is to validate a single timetable and to decide whether a timetable is operable and realizable in practice. However, microscopic models are inappropriate for mathematical optimization because of the size and the high level of detail. Hence, most optimization approaches consider simplified, so called *macroscopic*, models. The challenging part is to construct a reliable macroscopic model for the associated microscopic model and to facilitate the transition between both models of different scale.

In order to allocate railway capacity significant parts of the microscopic model can be transformed into aggregated resource consumption in space and time. We develop a general macroscopic representation of railway systems which is based on minimal headway times for entering tracks of train routes and which is able to cope with all relevant railway safety systems. We introduce a novel bottom-up approach to generate a macroscopic model by an automatic aggregation of simulation data produced by any microscopic model. The transformation aggregates and shrinks the infrastructure network to a smaller representation, i.e., it conserves all resource and capacity aspects of the results of the microscopic simulation by conservative rounding of all times. The main advantage of our approach is that we can guarantee that our macroscopic results, i.e., train routes, are feasible after re-transformation for the original microscopic model. Because of the conservative rounding macroscopic models tend to underestimate the capacity. We can control the accuracy of our macroscopic model by changing the used time discretization. Finally, we provide a priori error estimations of our transformation algorithm, i.e., in terms of exceeding of running and headway times.

In the second and main part (Chapter III) of the thesis, the optimal track allocation problem for macroscopic models of the railway system is considered. The literature for related problems is surveyed. A graph-theoretic model for the track allocation problem is developed. In that model optimal track allocations correspond to conflict-free paths in special time-expanded graphs. Furthermore, we made considerable progress on solving track allocation problems by two main features – a novel modeling approach for the macroscopic track allocation problem and algorithmic improvements based on the utilization of the bundle method.

More specifically, we study four types of integer programming model formulations for the track allocation problem: two standard formulations that model resource or block conflicts in terms of packing constraints, and two novel coupling or "configuration" formulations. In both cases variants with either arc variables or with path variables will be presented. The key idea of the new formulation is to use additional "configuration" variables that are appropriately coupled with the standard "train" flow variables to ensure feasibility. We show that these models are a so called "extended" formulations of the standard packing models.

The success of an integer programming approach usually depends on the strength of the linear programming (LP) relaxation. Hence, we analyze the LP relaxations of our model formulations. We show, that in case of block conflicts, the packing constraints in the standard formulation stem from cliques of an interval graph and can therefore be separated in polynomial time. It follows that the LP relaxation of a strong version of this model, including all clique inequalities from block conflicts, can be solved in polynomial time. We prove that the LP relaxation of the extended formulation for which the number of variables can be exponential, can also be solved in polynomial time, and that it produces the same LP bound. Furthermore, we prove that certain constraints of the extended model are facets of the polytope associated with the integer programing formulation. To incorporate robustness aspects and further combinatorial requirements we present suitable extensions of our coupling models.

The path variant of the coupling model provides a strong LP bound, is amenable to standard column generation techniques, and therefore suited for large-scale computation. Furthermore, we present a sophisticated solution approach that is able to compute high-quality integer solutions for large-scale railway track allocation problems in practice. Our algorithm is a further development of the rapid branching method introduced in Borndörfer, Löbel & Weider (2008) [37] (see also the thesis Weider (2007) [213]) for integrated vehicle and duty scheduling in public transport. The method solves a Lagrangean relaxation of the track allocation problem as a basis for a branch-and-generate procedure that is guided by approximate LP solutions computed by the bundle method. This successful second application in public transportation provides evidence that the rapid branching heuristic guided by the bundle method is a general heuristic method for large-scale path packing and covering problems. All models and algorithms are implemented in a software module `TS-OPT`.

Finally, we go back to practice and present in the last chapter several case studies using the tools `NETCAST` and `TS-OPT`. We provide a computational comparison of our new models and standard packing models used in the literature. Our computational experience indicates that our approach, i.e., "configuration models", outperforms other models. Moreover, the rapid branching heuristic and the bundle method enable us to produce high quality solutions for very large scale instances, which has not been possible before. In addition, we present results for a theoretical and rather visionary auction framework for track allocation. We discuss several auction design questions and analyze experiments of various auction simulations.

The highlights are results for the Simplon corridor in Switzerland. We optimized the train traffic through this tunnel using our models and software tools. To the best knowledge of the author and confirmed by several railway practitioners this was the first time that fully automatically produced track allocations on a macroscopic scale fulfill the requirements of the originating microscopic model, withstand the evaluation in the microscopic simulation tool `OpenTrack`, and exploit the infrastructure capacity. This documents the success of our approach in practice and the usefulness and applicability of mathematical optimization to railway track allocation.

Deutsche Zusammenfassung

Diese Arbeit befasst sich mit der mathematischen Optimierung zur effizienten Nutzung der Eisenbahninfrastruktur. Wir behandeln die optimale Allokation der zur Verfügung stehenden Kapazität eines Eisenbahnschienennetzes – das *Trassenallokationsproblem*. Das Trassenallokationsproblem stellt eine wesentliche Herausforderung für jedes Bahnunternehmen dar, unabhängig, ob ein freier Markt, ein privates Monopol, oder ein öffentliches Monopol vorherrscht. Die Planung und der Betrieb eines Schienenverkehrssystems ist extrem schwierig aufgrund der kombinatorischen Komplexität der zugrundeliegenden diskreten Optimierungsprobleme, der technischen Besonderheiten, und der immensen Größen der Probleminstanzen. Mathematische Modelle und Optimierungstechniken können zu enormen Nutzen führen, sowohl für die Kunden der Bahn als auch für die Betreiber, z.B. in Bezug auf Kosteneinsparungen und Verbesserungen der Servicequalität. Wir lösen diese Herausforderung durch die Entwicklung neuartiger mathematischer Modelle und der dazughörigen innovativen algorithmischen Lösungsmethoden für sehr große Instanzen. Dadurch waren wir erstmals in der Lage zuverlässige Lösungen für Instanzen der realen Welt, d.h. für den *Simplon* Korridor in der Schweiz, zu produzieren.

Das einführende Kapitel gibt einen umfangreichen Überblick zum Planungsprozeß im Eisenbahnwesen. Es liefert Einblicke in den ordnungspolitischen und technischen Rahmen, diskutiert die Beziehung zwischen den verschiedenen Planungsschritten und identifiziert Optimierungspotentiale in Eisenbahnverkehrssystemen. Der restliche Teil der Arbeit gliedert sich in zwei Hauptteile.

Der erste Teil (Kapitel II) beschäftigt sich mit der Modellierung des Schienenbahnsystems unter Berücksichtigung von Kapazität und Ressourcen. Kapazität im Schienenverkehr hat grundsätzlich zwei Dimensionen, eine räumliche, welche der physischen Infrastruktur entspricht, und eine zeitliche, die sich auf die Zugbewegungen innerhalb dieser bezieht, d.h. die Belegung- und Blockierungszeiten. Sicherungssysteme im Schienenverkehr beruhen überall auf der Welt auf demselben Prinzip. Ein Zug muss Blöcke der Infrastruktur für die Durchfahrt reservieren. Das gleichzeitige Belegen eines Blockes durch zwei Züge wird Blockkonflikt genannt. Um eine konfliktfreie Belegung zu erreichen, beinhalten Modelle zur Kapazität im Schienenverkehr daher die Definition und Berechnung von angemessenen Fahrzeiten und dementsprechenden Reservierungs- oder Blockierungszeiten.

Es gibt *mikroskopische* Modelle, die das Bahnsystem sehr ausführlich und genau beschreiben. Mikroskopische Modelle haben den Vorteil, dass die Berechnung der Fahrzeiten und des Energieverbrauchs der Züge sehr genau ist. Eine große Stärke von mikroskopischen Modellen ist, dass nahezu alle technischen Details und lokalen Besonderheiten einstellbar sind und bei den Berechnungen berücksichtigt werden. Wir beschreiben das Bahnsystem auf einer mikroskopischen Ebene, so dass das Verhalten der Züge und das Sicherheitssystem korrekt und vollständig abgebildet sind. Diese Modelle der Schieneninfrastruktur sind bereits für sehr kleine Netzausschnitte sehr groß. Der Grund ist, dass alle Signale, Neigungswechsel und Weichen im Vorfeld eines Bahnhofes modelliert werden müssen, um präzise Fahrzeitrechnungen zu erlauben. Im Allgemeinen wird diese Art der Modellierung in Simulationssystemen benutzt, die nahezu bei jedem Bahnunternehmen rund um die Welt im Einsatz sind. Die bedeutenste Anwendung dieser Systeme ist einen einzelnen Fahrplan zu validieren und zu entscheiden, ob ein Fahrplan betrieblich umsetzbar und in der Realität durchführbar ist. Mikroskopische Modelle sind jedoch aufgrund ihrer Größe und ihrer hohen Detailtiefe ungeeignet für eine mathematischen Optimierung. Dementsprechend betrachten die meisten Optimierungsansätze vereinfachte, so genannte *makroskopische*, Modelle. Die Herausforderung besteht hierbei darin, ein zuverlässiges makroskopisches Modell für ein entsprechendes mikroskopisches Modell zu konstruieren und den Übergang zwischen beiden Modellen verschiedener Detailstufen zu erleichtern.

Zur Belelgung von Kapazität im Bahnsystem können signifikante Teile der mikroskopischen Infrastruktur zu einem aggregierten Ressourcenverbrauch in Raum und Zeit transformiert werden. Wir entwickeln eine allgemeine makroskopischen Darstellung des Schienensystems, die auf minimalen Zugfolgezeiten für das Einbrechen von Zügen auf Gleisabschnitten basiert und welche damit in der Lage ist, alle relevante Sicherungssyteme im Schienenverkehr zu bewältigen. Wir führen einen neuartigen “Bottom-up”-Ansatz ein, um ein makroskopisches Modell durch eine automatische Aggregation von Simulationsdaten eines mikroskopischen Modells zu generieren. Diese Transformation aggregiert und schrumpft das Infrastrukturnetz auf eine kleinere Darstellung, wobei alle Ressourcen- und Kapazitätsaspekte durch konservatives Runden aller Zeiten erhalten bleiben. Der Hauptvorteil unseres Ansatzes ist, dass wir garantieren können, dass unsere makroskopischen Resultate, d.h. die Trassen der Züge, nach der Rücktransformation auch im mikroskopischen Modell zulässig sind. Durch das konservative Runden tendieren makroskopische Modelle die Kapazität zu unterschätzen. Die Genauigkeit des makroskopischen Modells können wir durch die gewählte Zeitdiskretisierung steuern. Schließlich liefern wir eine a priori Fehlerabschätzung unseres Transformationsalgorithmus, d.h. in der Beurteilung der Überschreitungen der Fahr- und Mindestzugfolgezeiten.

Im zweiten und Hauptteil (Kapitel III) der Dissertation wird das Problem des Bestimmens optimaler Trassenallokationen für makroskopische Bahnmodelle betrachtet. Ein Literaturüberblick zu verwandten Problemen wird gegeben. Für das Trassenallokationsproblem wird ein graphentheoretisches Modell entwickelt, in dem optimale Lösungen als maximal gewichtete konfliktfreie Menge von Pfaden in speziellen zeitexpandierten Graphen dargestellt werden können. Des Weiteren erreichen wir wesentliche Fortschritte beim Lösen von Trassenallokationsprobleme durch zwei Hauptbeiträge - die Entwickling einer neuartigen Modellformulierung des makroskopischen Trassenallokationsproblemes und algorithmische Verbesserungen basierend auf der Nutzung des Bündelverfahrens.

Im Detail studieren wir vier verschiedene Typen von ganzzahligen Modellformulierungen für das Trassenallokationsproblem: zwei Standardformulierungen, die Ressourcen- oder Blockkonflikte mit Hilfe von Packungsungleichungen modellieren, und zwei neuartige Kopplungs- oder "Konfigurationsmodelle". In beiden Fällen werden Varianten mit entweder Bogen- oder Pfadvariablen präsentiert. Die Kernidee dieser neuen Modelle besteht darin, zusätzliche "Konfigurationsvariablen" zu nutzen, die, um Zulässigkeit zu sichern, mit den Standard "Flussvariablen" der Züge entsprechend gekoppelt werden. Wir zeigen, dass diese Modelle eine spezielle Formulierung, eine sogenannte "extended formulation", der Standard Packungsmodelle sind.

Der Erfolg eines ganzzahligen Programmierungsansatzes hängt üblicherweise von der Stärke der LP Relaxierung ab. Infolgedessen analysieren wir die LP Relaxierungen unserer Modellformulierungen. Wir zeigen, dass sich im Falle von Blockkonflikten die Packungsbedingungen der Standardformulierung aus den Cliquen eines Intervallgraphen ergeben und diese sich deswegen in polynomieller Zeit bestimmen lassen. Wir beweisen, dass die LP Relaxierung der "extended formulation" bei der die Anzahl der Variablen exponentiell sein kann, ebenso in polynomieller Zeit gelöst werden kann und dass diese Relaxierung diesselbe LP Schranke liefert. Des Weiteren beweisen wir, dass bestimmte Bedingungen der "extended formulation" Facetten des Polytops der entsprechenden ganzzahligen Modellformulierung sind.

Die Pfadvariante des Konfigurationsmodells besitzt eine starke LP - Schranke, ist geeignet für Spaltenerzeugungstechniken und ist somit verwendbar zum Lösen sehr großer Probleme. Des Weiteren präsentieren wir ein fortgeschrittenen Lösungsansatz, der in der Lage ist Lösungen hoher Qualität für große Trassenallokationsprobleme zu berechnen. Unser Algorithmus ist eine Weiterentwicklung der "rapid branching"-Methode von Borndörfer, Löbel & Weider (2008) [37] (siehe ebenso Weider (2007) [213]) zur Lösung von integrierten Umlauf- und Dienstplanungsproblemen im öffentlichen Personenverkehr. Die Methode löst eine Lagrange-Relaxierung des Trassenallokationsproblems als Grundlage für einen branch-and-generate Algorithmus, der durch approximative Lösungen des Bündelverfahrens für das LP geleitet wird. Diese erfolgreiche zweite Verkehrsanwendung liefert den Beleg, daß die "rapid branching"-Methode ein vielversprechender allgemeiner Ansatz zum Lösen großer Pfadpackungs- und Pfadüberdeckungsprobleme ist. Die neuen Modelle und Algorithmen sind im Software-Tool `TS-OPT` implementiert.

Abschließend blicken wir zurück zur praktischen Anwendung und präsentieren im letzten Kapitel mehrere Fallstudien unter Verwendung der entwickelten Werkzeuge NETCAST und `TS-OPT`. Wir liefern einen ausführlichen Vergleich der Rechnungen unserer neuartigen Modelle mit bekannten Standardmodellen aus der Literatur. Unsere Rechenresultate zeigen, dass der neuartige Ansatz, d.h. die "Konfigurationsmodelle", andere Modelle in den meisten Fällen übertrifft. Zudem ermöglichen uns die "rapid branching"-Heuristik und die Bündelmethode qualitativ hochwertige Lösungen für sehr große Probleminstanzen zu produzieren, was bisher nicht möglich war. Daneben präsentieren wir theoretische und eher visionäre Resultate für die Vergabe von Trassen innerhalb eines Auktionsrahmens. Wir diskutieren verschiedene Fragestellungen zur Auktionsform und analyzieren Simulationsexperimente verschiedenener Auktionen.

Den Höhepunkt bilden Resultate für Praxisszenarios zum Simplon Korridor in der Schweiz. Nach bestem Wissen des Autors und bestätigt durch zahlreiche Eisenbahnpraktiker ist dies das erste Mal, dass auf einer makroskopischen Ebene automatisch erstellte Trassenallokationen die Bedingungen des ursprünglichen mikroskopischen Modells erfüllen und der Evaluierung innerhalb des mikroskopischen Simulationstools `OpenTrack` standhalten. Das dokumentiert den Erfolg unseres Ansatzes und den Nutzen and die Anwendbarkeit mathematischer Optimierung zur Allokation von Trassen im Schienenverkehr.

Acknowledgements

First of all, I am very grateful to Prof. Dr. Dr. h. c. mult. M. Grötschel for having given me the possibility to stay at the Zuse Institute Berlin after writing my diploma thesis. Thank you for the trust and the freedom during these past years.

A fundamental person all through the thesis work was my supervisor Dr. habil. Ralf Borndörfer. You always had time for me even if you were acquiring and heading thousands of projects. You always trusted me, taught me how to structure a project, how to get the big picture, how to identify open questions, where contributions are still needed without getting lost in all technical details, and many more. Special thanks goes also to Dr. Steffen Weider who provided me his code of the bundle method and supported my adaption and further development of the rapid branching heuristic.

Applied research is really applied only if it is done and evaluated in close collaboration with an industrial and operating partner. Therefore, I am very thankful for all discussions with external experts from Lufthansa Systems Berlin, DB Schenker, DB GSU, and in particular from Swiss Federal Railways (SBB). Special thanks go to Thomas Graffagnino and Martin Balser for explaining various technical details from railway systems and discussing several results. In addition, I want to thank Daniel Hürlimann for his support for the simulation tool `OpenTrack`. I also greatly appreciated the contact with international colleagues from Aachen, Rotterdam, Delft, Bologna, Zürich, Chemnitz, Kaiserslautern and Darmstadt during several fruitful conferences.

I would like to thank also all my colleagues at the department Optimization that made my time as a PhD student so enjoyable. The vivid atmosphere of the Optimization group was also very enriching. In particular, the daily coffee breaks with - Kati, Stefan H., Stefan V., Christian, Timo, Ambros, Jonas and all the others - has become a kind of institution for reflection and motivation. Furthermore, I would like to thank Marika Neumann, Markus Reuther, Rüdiger Stephan, Elmar Swarat, Steffen Weider, and Axel Werner for proof-reading and discussing parts of my thesis. Last but not least I want to thank my girlfriend Ina and my family for their patience and support.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Chapter I

Planning in Railway Transportation

The purpose of our work is to develop mathematical optimization models and solution methods to increase the efficiency of future railway transportation systems. The reasons for this is manifold; liberalization, cost pressure, environmental and energy considerations, and the expected increase of the transportation demand are all important factors to consider. Every day, millions of people are transported by trains in Germany. Public transport in general is a major factor for the productivity of entire regions and decides on the quality of life of people.

Figure 1 shows the expected development of freight transportation in Germany from 2003 to 2015 as estimated by the Deutsche Bahn AG (DB AG). This estimate was the basis of the last German Federal Transport Infrastructure Plan 2003 (Bundesverkehrswegeplan 2003), see Federal Transport Infrastructure Planning Project Group (2003) [87]. It is a framework investment plan and a planning instrument that follows the guiding principle of “development of Eastern Germany and upgrading in Western Germany”. The total funding available for road, rail and waterway construction for the period from 2001 to 2015 is around 150 billion euros.

The railway industry has to solve challenging tasks to guarantee or even increase their quality of service and their efficiency. Besides the need to implement adequate technologies (information, control, and booking systems) and latest technology of equipment and resources (trains, railway infrastructure elements), developing mathematical support systems to tackle decision, planning, and in particular optimization problems will be of major importance.

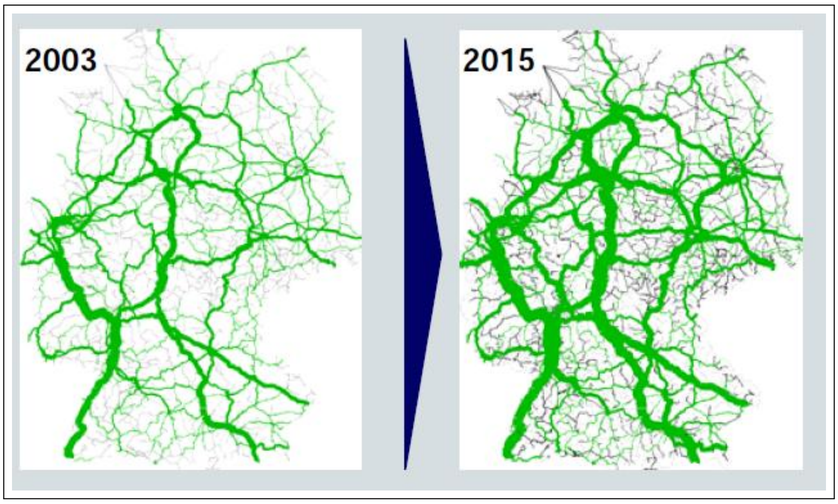


Figure 1: Estimated demand for (freight) railway transportation in Germany, source: Federal Transport Infrastructure Planning Project Group (2003) [87].

In Section 1 we will give a comprehensive introduction on the political environment and organizational structures because both directly affect the planning and operation of railway transport. In addition, we will refurbish an early publication from Charnes & Miller (1956) [67] that demonstrates prominently that railway transport is one of the initial application areas for mathematics in particular for discrete and linear optimization.

Only recently railway success stories of optimization models are reported from Liebchen (2008) [149], Kroon et al. (2009) [140], and Caimi (2009) [57] in the area of periodic timetabling by using enhanced integer programming techniques. This thesis focuses on a related planning problem – *the track allocation problem*. Thus, Section 2 gives a general overview of an idealized planning process in railway transportation. We will further describe several other planning problems shortly including line planning in Section 5 and crew scheduling in Section 8 in more detail. Mathematical models and state of the art solution approaches will be discussed, as well as the differences to and similarities with equivalent planning tasks of other public transportation systems. Moreover, in Section 6 we will depict the requirements and the process of railway capacity allocation in Europe to motivate and establish a general formulation for the track allocation problem.

We will show how to establish a general framework that is able to handle almost all technical details and the gigantic size of the railway infrastructure network by a novel aggregation approach. Therefore, and to build a bridge to railway engineering, we explain the most important microscopic technical details in Chapter II. Furthermore, we introduce a general standard for macroscopic railway models which is publicly available TTPlib [208] and develop a multi-scale approach that automatically transforms microscopic railway models from real world data to general macroscopic models with certain error estimations.

Nevertheless, the resulting macroscopic track allocation problems are still very large and complex mathematical problems. From a complexity point of view track allocation problems belongs to the class of $\mathcal{NP}$-hard problems. In order to produce high quality solutions in reasonable time for real world instances, we develop a strong novel model formulation and adapt a sophisticated solution approach. We believe that this modeling technique can be also very successful for other problems – in particular if the problem is an integration of several combinatorial problems which are coupled by several constraints. Chapter III will introduce and analyze this novel model formulation called "configuration" model in case of the the track allocation problem. Furthermore, we will generalize and adapt the rapid branching heuristic of Weider (2007) [213]. We will see that we could significantly speed up our column generation approach by utilizing the bundle method to solve the Lagrangean relaxation instead of using standard solvers for the LP relaxations.

Finally, to verify our contributions on modeling and solving track allocation problems in Chapter IV, we implemented several software tools that are needed to establish a track allocation framework:

- ▷ a transformation module that automatically analyses and simplifies data from microscopic simulation tools and provides reliable macroscopic railway models (NETCAST),
- ▷ an optimization module that produces high quality solutions (together with guaranteed optimality gaps) for real world track allocation problems in reasonable time (TS-OPT),
- ▷ and a $3d$-visualization module to illustrate the track allocation problem, to discuss the solutions with practitioners, and to automatically provide macroscopic statistics (TraVis).

1 Introduction

Railway systems can be categorized as either public or private. Private railway systems are owned by private companies and are with a few exceptions exclusively planned, built, and operated by this single owner. Prominent examples are the railway systems in Japan and the US, see Gorman (2009) [102]; Harrod (2010) [112]; White & Krug (2005) [215]. In contrast, public railway systems are generally funded by public institutions or governments. In the past an integrated railway company was usually appointed to plan, build, and operate the railway system. Now the efforts of the European Commission to segregate the integrated railway companies into a railway infrastructure manager (network provider) and railway undertakings (train operating companies) shall ensure open access to railway capacity for any licensed railway undertaking. The idea is that competition leads to a more efficient use of the railway infrastructure capacity, which in the long run shall increase the share of railway transportation within the European member states. However, even in case of an absolute monopoly the planning of railway systems is very complex because of the technicalities and operational rules. This complexity is further increased by the varying requirements and objectives of different participating railway undertakings in public railway systems.

The focus of this work is capacity allocation in an arbitrary railway system. In a nutshell, the question is to decide which train can use which part of the railway infrastructure at which time. Chapter I aims to build an integrated picture of the railway system and railway planning process, i.e., we will illuminate the requirements of passenger and freight railway transportation. In Chapter II resource models will be developed that allow for capacity considerations. Based on one of these railway models, i.e., an aggregated macroscopic one, we will formulate a general optimization model for private and public railway systems in Chapter III, which meets the requirements of passenger and freight railway transportation to a large extent.

Several railway reforms in Europe were intended to promote on-rail competition leading to more attractive services in the timetable. However, even after the reforms were implemented, the railways continued to allocate train paths on their own networks themselves. Discrimination was thus still theoretically possible. However, competition can only bring benefits if all railway undertakings are treated equally when seeking access to the infrastructure.

Switzerland has been pioneer in introducing competition in the use of the rail networks. The three different Swiss railway network providers, SBB, BLS and SOB, outsourced the allocation of their train paths to a joint independent body. Accordingly, at the beginning of 2006, and in conjunction with the Swiss Public Transport Operators' Association, these railways together founded the `Trasse Schweiz AG` (trasse.ch).

By outsourcing train path allocation to a body which is legally independent and independent in its decision making, the three largest Swiss standard gauge railways together with the Swiss Public Transport Operators Association reinforced their commitment to fair on-rail competition. This institution ensures that the processes to prepare for the timetable are free of discrimination. `Trasse Schweiz AG` coordinates the resolution of conflicts between applications and allocate train paths in accordance with the legislation. One of their principles is:

> "We increase the attractiveness of the rail mode by making the best use of the network and optimizing the application processes."

That statement essentially summarizes the main motivation of this thesis.

An initial publication on applying linear optimization techniques comes from railway freight transportation. Charnes & Miller (1956) [67] discussed the scheduling problem of satisfying freight demand by train circulations. The setting is described by a small example in Figure 2. In a graph with nodes $\#1, \#2$, and $\#3$ a directed demand which has to be satisfied is shown on each arc. The goal is to determine directed cycles in that graph that cover all demands with minimal cost, i.e., each cycle represents a train rotation. For example choosing four times the rotation $(\#1, \#2, \#1)$ would cover all required freight movements between $\#1$ and $\#2$. However the demand from $\#2$ to $\#1$ is only one and therefore that would be an inefficient partial solution with three empty trips called "light moves" in the original work. Charnes and Miller proposed a linear programming formulation for the problem enumerating all possible rotations, i.e., five directed cycles $(\#1, \#2, \#1)$, $(\#1, \#3, \#1)$, $(\#2, \#3, \#2)$ in Figure 2. Multiple choices of cycles that satisfy all demands represent a solution. Thus, for each rotation an integer variable with crew and engine cost was introduced. The opti-

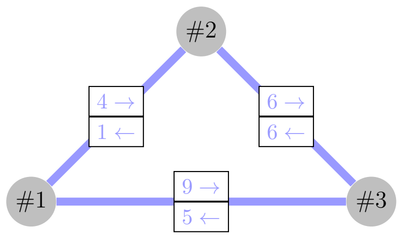


Figure 2: Simplified routing network of Charnes & Miller (1956) [67].

mization model states that the chosen subset has to fulfill all demands. This was one of the first approaches to solve real applications by means of a set partitioning problem, i.e., to represent a solution as a set of sub-solutions, here cycles. Finally, they manually solved the instance by applying the simplex tableau method.

After that pioneering work on modeling it took many years of improvement in the solution techniques to go a step further and to support more complex planning challenges in public transportation and in particular in railway transportation by optimization.

In fact the airline industry became the driving force of the development. One reason is the competitive market structure which leads to a higher cost pressure for aviation companies. Therefore the airline industry has a healthy margin in the implementation of automated processes and the evaluation of operations. Integrated data handling, measuring the quality of service and controlling the planning and operation by several *key performance indicators* (KPI) are anchored in almost all aviation companies over the world. Nowadays in the airline industry the classical individual planning problems of almost all practical problem sizes can be solved by optimization tools. Integration of different planning steps and the incorporation of uncertainty in the input data can be tackled. A prominent example for such *robust optimization* approaches is the tail assignment problem which is the classical problem of assigning flights to individual aircraft. Nowadays robust versions can be tackled by stochastic optimization, see Lan, Clarke & Barnhart (2006) [144], or a novel probability of delay propagation approach by Borndörfer et al. (2010) [41]. Suhl, Dück & Kliewer (2009) [205] use similar ideas and extensions to increase the stability of crew schedules.

An astonishing situation happened in Berlin which somehow documents the challenges and problems that might result from the deregulation. The British Financial Times wrote on 27th of July 2009:

"Concrete walls, watch-towers, barbed wire and armed border guards for decades prevented Germans travelling across Berlin from the east to the west. But, as the German capital gears up to celebrate 20 years since the fall of the Berlin Wall, leftwing commentators are claiming that capitalism, not communism, is now keeping the two apart. For the S-Bahn - the suburban commuter railway running into and around Berlin that became a symbol of the cold war divide - has come grinding to a halt.

More than two-thirds of the network's 550 trains were withdrawn from service last week and the main east-west line closed after safety checks following a derailment showed that about 4,000 wheels needed replacing. Hundreds of thousands of Berliners have been forced to get on their bikes or use alternative, overcrowded routes to work, while tourists weaned on stereotypical notions of German punctuality and efficiency have been left inconvenienced and bemused by the chaos. Deutsche Bahn, the national railway operator, is under fire for cutting staff and closing repair workshops at its S-Bahn subsidiary in an attempt to boost profitability ahead of an initial public offering, that has since been postponed.

For businesses dependent on the custom of S-Bahn passengers, the partial -suspension of services is no joke. "For the past two or three days it's been really bad. Customers are down by more than half," said an employee at a clothing-alteration service situated below the deserted S-Bahn platform at Friedrichstrasse station, in the former East Berlin. "German trains are world famous. I didn't think -something like this could happen."

A columnist for Tagesspiegel, a Berlin-based newspaper, drolely observed that the number of S-Bahn carriages rendered unusable by management incompetence was only slightly less than the total number damaged by the Red Army in 1945. Others note that even the Berlin Wall itself did not prevent S-Bahn passengers traveling between west and east, so long as they held a West German passport. The East German authorities continued to operate the S-Bahn in West Berlin after the partition of the city following the second world war until the 1980s. West Berliners eventually boycotted this service in protest of the communist

> regime. But now it is being claimed that capitalism is driving passengers away.
>
> “The chaos in the Berliner S-Bahn is a lesson in the consequences of capitalism. It is a graphic depiction of where subservience to financial markets greedy pursuit of profit ultimately leads,” Ulrich Maurer, chief whip of the radical Left party, said. Deutsche Bahn has apologized for the inconvenience but insists that cost-cutting was not the problem and blames the train manufacturer instead. ”Even if we had had twice as many employees and three times as many workshops . . . it would not have prevented these wheels from breaking,” a Deutsche Bahn spokesman said. Nevertheless, S-Bahn-Berlin’s entire senior management was forced to resign this month after it emerged that they had not ordered sufficient safety checks. The repairs, refunds, and lost fares could leave Deutsche Bahn up to 100 million euros out of the pocket, according to one estimate. A full service is not expected to resume until December.”

The described situation documents that the railway system in Europe has to face huge challenges in implementing the liberalization. In addition central topics of the railway system are often politically and socially sensitive subjects. A detailed characterization of the recent political situation of the German railway system, future perspectives, the role of the infrastructure, and other controversial issues can be found in G.Ilgmann (2007) [99]. All in all we hope and we believe that an innovation process in the railway system in Europe is going to start. Major railway planning decisions can be supported by mathematical models and optimization tools in the near future, in particular the almost manual construction of the timetables and track allocations which is often seen as the “heart” of the railway system.

Due to the deregulation and the segregation of national railway companies in Europe the transfer of mathematical optimization techniques to railway operations will proceed. In the future, competition will hopefully give rise to efficiency and will lead to an increasing use of information technology and mathematical models. Algorithmic decision support to solve the complex and large scale planning problems may become necessary tools for railway transportation companies. In the future state of the art planning systems with optimization inside will replace the “manual” solution. The key message is that optimization, i.e., mathematical models and solution methods, are predestined to support railway planning challenges now and in the future.

In the following section we will briefly highlight several of these planning problems from different transportation modes. We will present mathematical models and discuss state of the art solution approaches to tackle real world applications, see Barnhart & Laporte (2007) [17] for an overview on optimization in transportation in general. We use in this thesis the definitions and notation of Grötschel, Lovász & Schrijver (1988) [104] and Nemhauser & Wolsey (1988) [167] for graphs, linear programs (LPs), and mixed integer programs (MIPs). Furthermore, we use the algorithmic terminology to LP and MIP solving of Achterberg (2007) [3].

2 Planning Process

Bussieck, Winter & Zimmermann (1997) [50] divide the planning process in public transport into three major steps - strategic, tactical and operational planning. Table 1 shows the goals and time horizon of all steps. Public transport, especially railway transportation, is such a technically complex and large system that it is impossible to consider the entire system at once. Also, the different planning horizons of certain decisions enforce a decomposition. Therefore a sequence of hierarchical planning steps has emerged over the years. However, in reality there is no such standardization as we will explain it theoretically.

Two important parties are involved in the railway transportation planning process, i.e., train operating companies and railway infrastructure providers. Following the terminology of the European commission, we will use the terms railway undertaking (RU) and infrastructure manager (IM), respectively. Furthermore, several national and international institutions have a huge political influence on railway transportation, which is on the borderline between a social or public good and a product that can be traded on a free liberalized market. The special case of the changing railway environment in Europe will be discussed in detail in Section 6.1.

In contrast to railway undertakings fully private aviation or independent urban public transport companies can perform the complete planning process almost internally. In the airline industry the needed infrastructure capacity, i.e., the slots at the airports, are granted by grandfa-

level	time horizon	goal
strategic	5-15 years	resource acquisition
tactical	1-5 years	resource allocation
operational	24h - 1 year	resource consumption

Table 1: Planning steps in railroad traffic, source: Bussieck, Winter & Zimmermann (1997) [50].

ther rights, see Barnier et al. (2001) [21]; Castelli, Pellegrini & Pesenti (2010) [66]. Borndörfer, Grötschel & Jaeger (2008) [36]; Borndörfer, Grötschel & Jaeger (2009) [39] and Hanne & Dornberger (2010) [108] give recent surveys about the potential of optimization for transportation systems and the differences between the planning process in the airline industry, urban public transport, and the railway industry. In the case of urban public transport the planning process is discussed in Weider (2007) [213] and Borndörfer, Grötschel & Pfetsch (2007) [35]. A detailed description of the process in the airline industry can be found in Grönkvist (2005) [103] and Barnhart & Laporte (2007) [17]. Bussieck (1997) [49] describes the use of discrete optimization in the planning process of public rail transport in the case of an integrated system. Analogous considerations can be found in Liebchen (2006) [148] and Lusby et al. (2009) [159]. There the planning steps are classified with respect to the time horizon and their general purposes.

Strategic or long-term part concerns the issues of network design and line planning (resource acquisition), see Sections 3 and 5. On the tactical stage the level of services, usually a timetable has to be created, as well as the schedules for the needed resources (resource allocation). Finally, on the operational stage the resources, e.g., rolling stock, vehicles, aircraft and crews, are monitored in real operations (resource consumption).

On the day of operation re-scheduling and dispatching problems have to be faced. These kind of problems have a different flavor than pure planning tasks. Decisions must be made very quickly in the real-time setting, but only limited information on the "scenario" is available. Usually data has to be taken into consideration in a so called online fashion. More details about this kind of problem can be found in Grötschel, Krumke & Rambau (2001) [105], Albers & Leonardi (1999) [9], and Albers (2003) [8]. Recent approaches are to establish fast methods which bring the "real" situation back to the "planned" one when

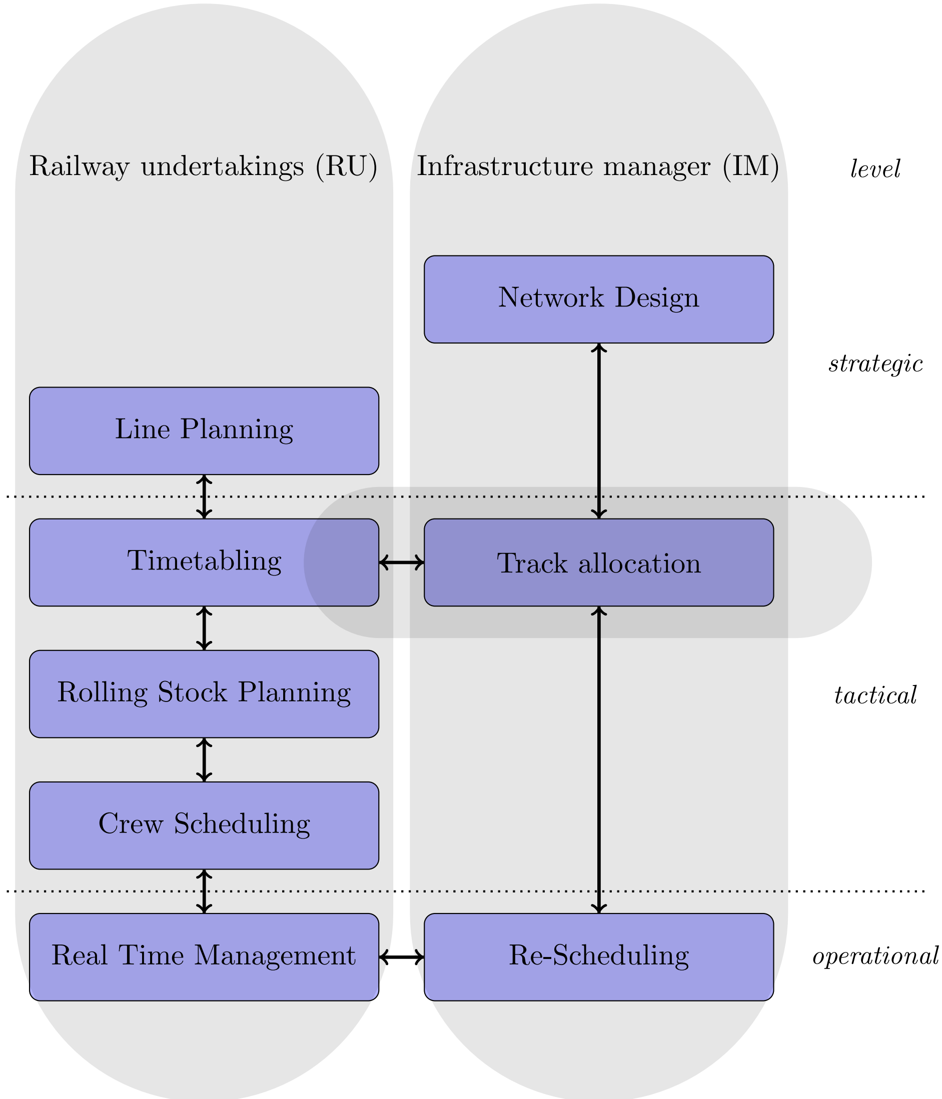


Figure 3: Idealized planning process for railway transportation in Europe.

possible, see Potthoff, Huisman & Desaulniers (2008) [177], Rezanova & Ryan (2010) [182], and Jespersen-Groth et al. (2009) [123].

In Klabes (2010) [129] the planning process is newly considered for the case of the segregated European railway system. In Figure 3 the novel process is illustrated for the segregated railway industry in Europe.

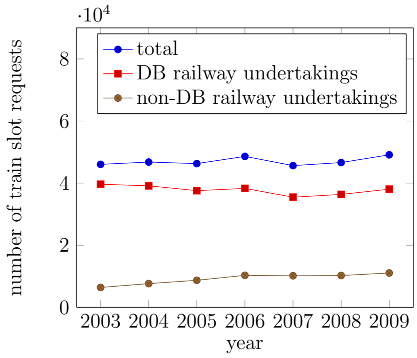
2.1 Strategic Planning

The responsibilities of the planning steps refer directly to either the railway undertaking or the infrastructure manager on behalf of the state. Nevertheless the long-term decisions in up- or downgrading the network are highly influenced by the railway undertakings and their demands. In case of passenger railway undertakings the desired timetable aims to implement a given line plan. The timetable itself induces train slots requests which is one input for the track allocation problem. These are naturally very strict with respect to departure and arrival times in order to offer and operate a concrete and reliable timetable. Further details on line planning and periodic timetabling are given in Section 5 and Section 6.2, respectively.

The requirements of train slot requests for cargo or freight railway operators differ significantly from slot requests for passenger trains because they usually have more flexibility, i.e., arrival and departure are only important at stations where loading has to be performed. Section 3 will describe the network design problem of the major European single wagon railway transportation system. In general freight railway operators need a mixture of annual and ad hoc train slots. The demand is of course highly influenced by the industry customers and the freight concept of the operating railway undertaking. We collected such data for the German subnetwork hakafu_simple to estimate the demand of the railway freight transportation, see Chapter IV Section 1 and Schlechte & Tanner (2010) [189].

2.2 Tactical Planning

The essential connection between all train slot requests is the step to determine the complete track allocation, which is the focus of this work. However, we primarily consider the point of view of a railway infrastructure provider, which is interested in optimizing the utilization of the network. That is to determine optimal track allocations. This is in contrast to timetabling where one asks for the ideal arrival and departure times to realize a timetable concept or a line plan. A timetable can be seen as a set of train slot requests without flexibility. Railway optimization from a railway undertaking's point of view for passenger traffic is discussed in Caprara et al. (2007) [64]. State of the art modeling and optimization approaches to periodic timetabling, which is the



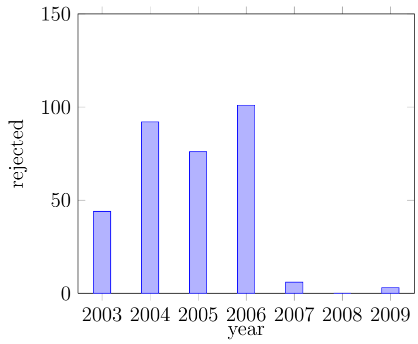


Figure 4: Requested train paths at DB, source: Klabes (2010) [129].

usual type of schedule for passenger railway traffic, is at length studied by Liebchen (2006) [148].

The induced competition for railway capacity allocation in public railway systems in Europe has a several impacts on the allocation procedure. In the past, a single integrated railway company performed the complete planning. Its segregation reduces the ability of the railway infrastructure manager to only perform network planning, capacity allocation, and re-scheduling with respect to infrastructure aspects. Thus, the infrastructure manager only has limited information during the planning process and needs to respect the confidential information of the railway undertakings. Moreover, new railway undertakings enter the market which increases the complexity of the planning process. Klabes (2010) [129] collected the relevant numbers from the DB Netz reports. On the left hand of Figure 4 the changing environment is illustrated by listing the growing number of train slot requests from railway undertakings independent from the former integrated railway company "Deutsche Bahn". On the right hand of Figure 4 the number of rejected train slot requests for the same periods are shown. It can be seen that at the start of the segregation from 2003 until 2006 a lot of requests had to be rejected by DB Netz. Efforts to decrease these numbers by providing alternative slots were apparently successful in the following years.

The business report for the year 2009 Trasse Schweiz AG [207] of the `Trasse Schweiz AG` documents the new challenges for constructing track allocations as well. In the Swiss network a lot of different railway undertakings are operating, e.g., in 2009 there were 29 train operating companies which submitted train slot requests. The geographical position in central Europe and the limited transportation possibilities through the Alps causes that. The future challenge for Switzerland will be to handle the complex track allocation process as the following extract from the report 2009 already highlights:

> "The regulation of the conflicts arising in train slot orders of the annual timetable 2010 was despite or even less because of the financial or economic crisis in comparison to the last years extensive and time-consuming. Indeed the number of submitted train slot requests by cargo operators for the annual timetable 2010 decreased up to 10 percent in comparison to the last year. However, railway undertakings (RM) concentrated her orders due to the cost pressure and competitive market conditions on the most attractive time windows and stick much longer to their original requests. Nevertheless, we managed together with all infrastructure providers[1] to find for all conflicts alternative train slots, which were accepted by the railway undertakings. No train slot request had to be rejected." (translation by the author).

The competing railway undertakings should interact in a transparent and free market. The creation of such a market for railway capacity is a key target of the European Commission, hoping that it will lead to a more economic utilization of the railway infrastructure. Even more liberalization of the railway system should lead to a growing market and allow for innovative trends like in other old-established industries, i.e., aviation industry, telecommunication or energy market. After the acceptance of train slots each railway undertaking determines his partial operating timetable, which acts as input for the planning of the needed resources. In case of a railway operator the rolling stock rotations have to be constructed, which is very complex problem due to several regularities and maintenance requirements, see Fioole et al. (2006) [88],Anderegg et al. (2003) [12], Eidenbenz, Pagourtzis & Widmayer (2003) [80], and Peeters & Kroon (2008) [176].

In public transport and in airline industry vehicle scheduling and aircraft rotation planning are the analogous tasks, see Löbel (1997) [155] and Grönkvist (2005) [103]. The major objective is to operate a reliable timetable with minimum cost, which is in general minimizing the number of engines, wagons, vehicles, aircrafts, etc. Another key requirement for planning railway rolling stock rotations is to provide

[1]There are three different railway infrastructure providers in Switzerland, i.e., BLS, SBB, and SOB.

regularity of the solutions. This means that a train that runs in the same way every day of the week, will also be composed in the same way every day of the week, always using the same cars from the same preceding trains. Such a regime simplifies the operation of a railway significantly. However, the rule can not always be followed. Trains may run later on weekends, or not at all on certain days, e.g., in order to perform a maintenance operation. Although it is intuitively clear, it is not easy to give a precise definition what regularity actually means.

The output of rolling stock planning is to assign trains, i.e., specific train configurations, to each passenger trip, to select deadhead trips, i.e., "empty" movements of the trains given by the constructed rolling stock rotation, and to schedule maintenances and turn around activities of trains. Passenger trips, that are trips of the published timetable, and deadhead trips need to be assigned to crews, which have to execute them. We will describe this planning step in more detail in Section 8 in case of an aviation company. This demonstrates the power of general mathematical modeling and methodology to different applications and that the authors experience about that planning step comes from airline crew scheduling, i.e., pairing optimization. However, recent work on railway crew scheduling can be found in Abbink et al. (2005) [1] and Bengtsson et al. (2007) [24].

2.3 Operational Planning

As already mentioned real time problems on the day of operation have quite different requirements, even if these problems can be formulated very similar from a mathematical modeling point. In railway transportation disruption and delay management is very difficult because local decisions have a huge influence on the complete timetable system. Nevertheless, easy and fast rules of thumb are used to decide, which trains have to be re-routed, have to wait, or even have to be canceled. D'Ariano et al. (2008) [72] and Corman, Goverde & D'Ariano (2009) [71] presented a real-time traffic management system to support local dispatching in practice. On the basis of this renewed timetable rolling stock rosters and crew schedules have to be adopted, see Clausen et al. (2010) [69]; Jespersen-Groth et al. (2007) [122]; Potthoff, Huisman & Desaulniers (2008) [177]; Rezanova & Ryan (2010) [182].

Every single step in this idealized sequential planning process is a difficult task by itself or even more has to be further divided and simplified into subproblems. We will discuss several of them in the following subsections, see how they can be modeled as combinatorial optimization problems, and solved by state of the art solution approaches.

The main application of track allocation is to determine the best operational implementable realization of a requested timetable, which is the main focus of this work. But, we want to mention that in a segregated railway system the track allocation process directly gives information about the infrastructure capacity. Imaging the case that two trains of a certain type, i.e., two train slots, are only in conflict in one station. A potential upgrade of the capacity of that station allows for allocating both trains. This kind of feedback to the department concerning network design is very important. Even more long-term infrastructure decisions could be evaluated by applying automatically the track allocation process, i.e., without full details on a coarse macroscopic level but with different demand expectations. Even if we did not developed our models for this purpose, it is clear that suitable extensions or simplifications the other way around of our models could support infrastructure decisions in a quantifiable way. For example major upgrades of the German railway system like the high-speed route from Erfurt to Nürnberg or the extension of the main station of Stuttgart can be evaluated from a reliable resource perspective. The billions of euros for such large projects can then be justify or sorted by reasonable quantifications of the real capacity benefit with respect to the given expected demand.

An obvious disadvantage of the decomposition is that the in some sense "optimal" solution for one step serves as fixed input for the subsequent problem. Therefore one cannot expect an overall "optimal" solution for the entire system. In the end, not even a feasible one is guaranteed. In that case former decisions have to be changed, and a part or the complete process has to be repeated. Prominent examples are regional scenarios for urban public transportation where traditional sequential approaches are not able to produce feasible schedules. Weider (2007) [213] demonstrates in case of vehicle and duty scheduling how integrated models can cope with that and even more can increase the overall planning efficiency. Nevertheless, hierarchic planning partitions the traffic planning problem into manageable tasks. Tasks lead directly to quantifiable optimization problems and can be solved by linear and integer programming to optimality or at least with proven optimality gaps. Problem standardization, automatization, organizing data, computational capabilities, mathematical modeling, and sophisticated algorithmic approaches on a problem specific but also on a general level form the basis of optimization success stories in practice. As a prominent example for this we refer to the dutch railway timetable - the first railway timetable which was almost constructed from scratch. In fact the entire planning process was decomposed and each planning problem at Netherlands Railways (NS) was solved by the support of exact or heuristic mathematical approaches and sophisticated techniques, in particular linear, integer and constraint programming. More details can be found in the prizewinning work Kroon et al. (2009) [140], which was honored with the *Franz Edelman Award* 2008. A prize which is rewarded to outstanding examples of management science and operations research practice in the world.

3 Network Design

Network design is the question of construction or modification of existing railway infrastructure. Railway infrastructure managers take the responsibility for that planning step in close cooperation with public authorities.

Infrastructure decisions are long term and very cost intensive, especially in railway systems. Typically an existing infrastructure has to be modified due to changes of the travel demand, capacity requirements, and new technologies. The usual objective is to minimize the construction cost while still ensuring the expected travel demand. Nevertheless this is a highly political planning step relying on uncertain future demand estimations. The resolution of such problems is carried out in close cooperation with senior management of the infrastructure owner due to the obviously high capital investment and the long lasting implications not only for the entire company even for the (national) railway system and for the affected cities as well. Prominent example is the recent project Stuttgart 21 that remains a subject of dispute in the public's view, see Kopper (2010) [137].

Standard approaches for the travel demand estimations are interviews of customers, evaluation of ticket sales, and various statistical methods based on automated passenger counts. All these methods are very costly and time consuming. But of course in the future more and more of these data will be collected automatically and available for analysis. However this can only be done for passenger traffic, the estimation of future demand of cargo traffic is even more difficult and needs different approaches. Furthermore, in a segregated railway system, this is confidential information of the railway undertakings, see Figure 3. Nevertheless, the information that a railway infrastructure manager collects during the allocation process for the annual timetable can be used to identify congested parts of the network, or downsizing potential.

A somehow exceptional and remarkable approach to railway network design was realized in the project *Rail2000* in Switzerland, see Kräuchi & Stöckli (2004) [138] and Caimi (2009) [57]. There, the sequential approach was re-ordered, the initial step was to define a service intention, i.e., finish line planning and passenger timetabling at first, to determine the required infrastructure. The major advantage is of course that the railway infrastructure matches perfectly to the explicit given service intention and is not based on coarse and aggregated demand forecast. The logical drawback is that the Swiss railway timetable, at least for the passenger traffic, is a very stable entity for the future years. The crucial assumption is that the demand is almost constant and the given service intention will change only slightly.

To the best of the authors knowledge only network design approaches to integrated railway systems can be found in the literature. The complex situation for a segregated railway system, i.e., for an infrastructure manager dealing with a lot of railway undertakings using the same infrastructure, is not considered on a general optimization level. Only several individual cases are discussed and analyzed as in Niekerk & Voogd (1999) [168] and Romein, Trip & de Vries (2003) [184]. Basic approaches are using simulation tools to evaluate, to analyze, and to compare some infrastructure possibilities as in Middelkoop & Bouwman (2000) [161] and Klima & Kavicka (2000) [133].

A framework for a general class of network design problems is presented in Kim & Barnhart (1997) [126] and applied to the blocking problem in railroad traffic in the US, see Barnhart, Jin & Vance (2000) [19]. Integrated service network design for rail freight transportation in the US is considered in Ahuja, Jha & Liu (2007) [6]; Jha, Ahuja & Şahin (2008) [124]; Zhu, Crainic & Gendreau (2009) [218]. In the next section, we will explain and discuss the network design problem for freight transportation for the German case in more detail.

Concluding, we want to point out that future developments and requirements of a railway infrastructure network, i.e., passenger or freight service networks, are very difficult to anticipate and highly political driven. A huge system knowledge and experience are preconditions for these crucial long-term design decisions, which are hard to quantify a priori, as well as a posteriori. However, the models developed in this work can support railway companies in evaluating possible network modifications and measure their impact from a quantifiable capacity point of view - even if this is not the main focus of our work.

4 Freight Service Network Design

Deutsche Bahn, the largest German railway company, primarily offers two products to industrial customers that want to transport freight via rail. Typically large customers order block-trains of about 20 to 40 cars. In this case, Deutsche Bahn, i.e., DB Schenker, as the operator can pull such a complete train by a locomotive from origin to destination. That is a direct freight transportation offer with a fixed train composition. Small customers on the other hand order only 1 to 5 cars. In such case it is too expensive to pull this group of cars by a single locomotive through the network. Instead the cars are only pulled to the next classification yard. There they are grouped with the cars from other customers, and then as new trains pulled to the next classification yard. There the trains are disassembled, and the cars are again re-grouped with others until each car has reached its final destination. This second freight transportation product of DB gives rise to a natural network design question, i.e., where are the classification yards located and how to route between them. Fügenschuh et al. (2008) [95] and Fügenschuh, Homfeld & Schülldorf (2009) [96] discuss the whole system of *single wagon freight transportation*, show the positive effect of bundling cars, and compare the problem to other freight transportation concepts mentioned in the literature, e.g., the *railroad blocking problem* in the US or Canada.

The railroad blocking problem can be formulated as a very large-scale, multi-commodity, flow-network-design and routing problem with billions of decision variables, see Jha, Ahuja & Şahin (2008) [124] and Barnhart, Jin & Vance (2000) [19]. Ahuja, Jha & Liu (2007) [6] presented an algorithm using an emerging technique known as very large-scale neighborhood search to support major US railway companies that transfers millions of cars over its network annually. The authors report that their heuristic approach is able to solve the problem to near optimality using one to two hours of computer time on a standard workstation computer.

Due to some similarities to our modeling approach for railway track allocation we want to explain the whole problem in more detail. The version which we will present in the next paragraphs describes the operational situation faced at DB Schenker Rail, the largest European cargo railway transportation company. We want to thank Alexander Below and Christian Liebchen for several discussions on that topic and system.

4.1 Single Wagon Freight Transportation

The *single wagon network* $N = (B, R)$ is a graph that describes the local transport possibilities of single wagons in a railway system. All inbound tracks and sorting sidings on satellite terminals, junction stations, and classification yards induce a node $b \in B$. An arc $r = (u, v)$ with $u, v \in B$ exists if a train trip from u to v is possible.

A *shipment* is an accepted order that consists of a number of single wagons (with different weight, length, type, etc.), departure station and interval (freight pickup definition), arrival station and interval (freight delivery definition), and a measure of the service quality of the transshipment in terms of penalties for the deviation of the requirements. The set of all shipments is denoted by S.

A *routing* is an unique path in N for each origin and destination pair given as a *routing matrix*, i.e., in some places depending on the wagon types or time of the day. The routing can equivalently be characterized by a set of in-trees. An in-tree is a directed graph with a so-called *root* node such that there exists exactly one directed path from each node to the root.

A *train slot* denotes a concrete temporal allocation of an arc in N by a standard freight train with a given number of wagons, maximum length, and maximum weight, i.e., each slot f has a discrete departure time d_f and an arrival time a_f. T denote the set of all given slots. In the German case we have to distinguish between three different types of slots:

1. safe slots with fixed timing, e.g., by master contracts
2. optional slots with relatively safe timing, e.g., system slots

3. (vague) requested slots with desired timing, e.g., chartered or extra train (slots).

The network design part at DB Schenker consist of deciding which of these timed slots should be requested from the network provider in order to run the system with a certain shipment quality and with minimal cost.

A *freight train trip*, or shortly trip, denotes an allocation of a slot with an ordered set of at most k shipments. Z denotes the set of all feasible trips. The set of all trips for slot f is denoted by Z_f.

In *classification yards* all single wagons will be rearranged with respect to the routing matrices, i.e., they will be sorted and shelfed in the corresponding siding. Classification yards are made of three parts; entry tracks, sorting tracks, and exit tracks. There the freight train is disassembled, and the individual shipments are pushed over the hump, entering the sorting tracks behind. Each sorting track is assigned to an unique successor $b \in B$. As soon as enough shipments are gathered on one sorting track, this new train is pulled into the exit group. There, it waits, until it can leave the yard and re-enter the network.

The nodes of N represent a simplified model of these yards, e.g., with a maximum shunting capacity per time interval. In practice the shunting procedure at the special yards is more restricted, e.g., minimum transition times, minimum distances between arrivals and departures, fixed downtimes, maximum operations per periods etc.

A *production schedule* is an assignment of all shipments to feasible trips such that the pickup and delivery definitions of all shipments can be guaranteed. In addition the production schedule, i.e., the set of trips, has to respect the routing principles and all operation rules and capacities at the classification yards.

4.2 An Integrated Coupling Approach

The problem of finding a production schedule can be modeled as an integer program with an exact representation of the given degrees of freedom. The main challenge is to adhere to the FIFO principle. In fact each trip that arrives in a yard has to be disassemble immediately. Each shipment will arrive as fast as possible at their unique sorting yard and will depart directly with the next trip.

The model belongs to a broad class of integer programs where a set of path systems are meaningfully coupled. In that application transportation paths of the shipments are linked with additional "configuration" variables, i.e., variables for trip construction in the yards.

The model is based on a *trip scheduling digraph* $D = (V, A)$ induced by N that describes the transportation of the individual shipments in place, time, and position within a trip. Each classification yard b induces an arrival track that models a waiting queue in front of the shunting hump.

For each yard $b \in B$ we associate an additional node b^+ and several additional nodes b^-_* that represents the different directions and sorting tracks to control the queue in front of the humping yard b. Each arc $r = (b, x) \in R$ of the single wagon network N is also considered as two arcs to handle sorting, i.e., an arc from $(b, x) \in R$ induces (b^+, b^-_x) and (b^-_x, x^+). Let $G = B^+ \cup B^-$ the set of all those expanded nodes associated with sorting on railway tracks. $[T] = \{0, \ldots, T-1\}$ denotes a set of discrete times and $[m] = \{0, \ldots, m-1\}$ a set of possible positions of shipments within a trip.

Thus, a node

$$v = (g, t, i) \in V \subseteq G \times [T] \times [m]$$

is a possible event modeling that a shipment arrives at track g, time t, and position i within a trip. Moreover it is an arrival event if $g \in B^+$ or otherwise a departure event. The position of a shipment is relevant due to the fact that we have to follow the FIFO principle at the classification yards. A larger position in a trip could result in a later departure from this classification yard. The set V contains all these events as well as the pickup and delivery of a shipment.

Arcs of D model the transport of shipments at precise positions within the trip and the transition of shipments from the incoming track of a yard to the sorting yards with all potential position changes. In addition all local rules, e.g., time restrictions, can be incorporated in that arc construction as well as the routing requirements.

Figure 5 shows a possible block (train) composition q for slot $f = (b, 14, y, 20, 4)$, i.e., a train slot that departs at b^-_x (b) and time 14 and arrives at y^+ (y) at time 20 with a maximum of 4 shipments. Two trains arrive from x^-_b at b^+ within the considered interval and reach the siding

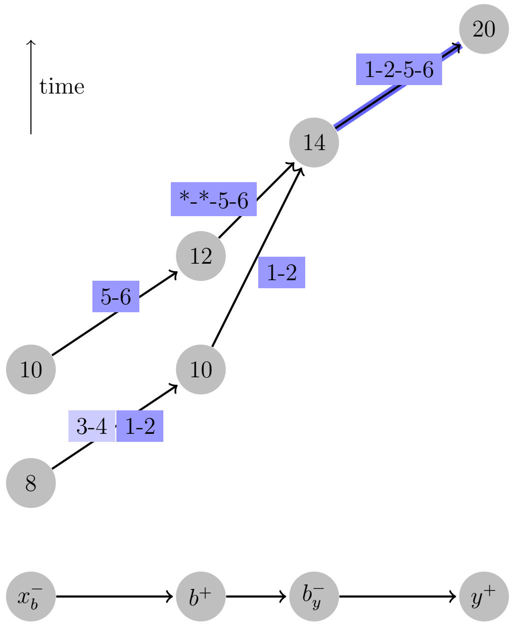


Figure 5: Possible train composition for track $f = (v_r, 14, w_b, 20, 4)$.

to y via b_y^-. In the course of this the position of shipments changes, e.g., shipments 1 and 2 from position 3 and 4 in the first train trip to 1 and 2 in the second. The arcs associated with b^+ and b_y^- control the sorting with respect to the routing matrix and the potential position changes of the shipments, i.e., $*$ denotes wild cards for first positions. The shipments 3 and 4 are not routed via y and therefore are not sorted on (b^+, b_y^-). The proposed trip composition networks can obviously become very large due to the ordering. However, the degree of freedom is somehow limited due to the fixed slots and routing principles, i.e., only certain positions are possible for the shipments.

The optimization task is to minimize the cost of the slots and the cost of the trip construction at the yards. Any production schedule can be represented in D by a set of feasible paths, i.e., one for each shipment. In the integer programming model the paths of the shipments are coupled with the construction of trips at the yards to respect the operational rules and the shipment positions. We will briefly explain the formulation. First we use trivial 0/1 variables $x_{f,t}$ to determine which trip t is used for slot f. The idea of the modeling technique is to introduce 0/1 variables y_q to control the creation of trips and to force the "real" operational routing of the shipments at the classification yards by means of inequalities

$$\sum_{t \in Z_f} x_{f,t} - \sum_{q \in Q_f} y_q = 0, \quad \forall f \in F.$$

The set Q_f can be interpreted as a certain subset of arcs in an auxiliary graph that represents the construction of trip t in the departure yard of slot f. On the hand if some trip t is selected for slot f by setting $x_{f,t} = 1$, then the construction of that train in the departure yard must be feasible which is ensured by setting the "right" variables y_q to one. On the other hand if trip t is not used on slot f all corresponding configuration variables y_q have to be zero. If no degrees of freedom for selecting slots are given, then this model only propagates the operational rules at the classification yards. In addition, an optimized selection of slots is a strategic question that can be answered by those models using a reasonable set of slots.

That example serves only for motivational purposes of a general modeling technique that couples and integrates problems appropriately. In addition it should give the reader some insights in the source of the particular train slot requirements of a freight railway operator. Since train slots defined and used by single wagon freight service operators serve as direct input for track allocation problems.

5 Line Planning

Once the infrastructure of the passenger transportation system is determined, lines have to be defined and associated with individual frequencies. A *line* is a transportation route between two designated, but not necessarily different, terminal stations in the transportation network. Usually there are some intermediate stops, but especially in long distance passenger railway transportation direct lines, i.e., in Germany called *Sprinter*, are used to offer very fast connections between major cities. A train line also includes the specification of the train type, i.e., type of engine, number of wagons, and its frequency, in case of regular periodic services. For example this can be four times an hour during peak-hour traffic and two times an hour in off-hour traffic. The *Line Planning Problem* is to select a set of feasible lines and their frequencies subject to certain constraints and pursuing given objectives.

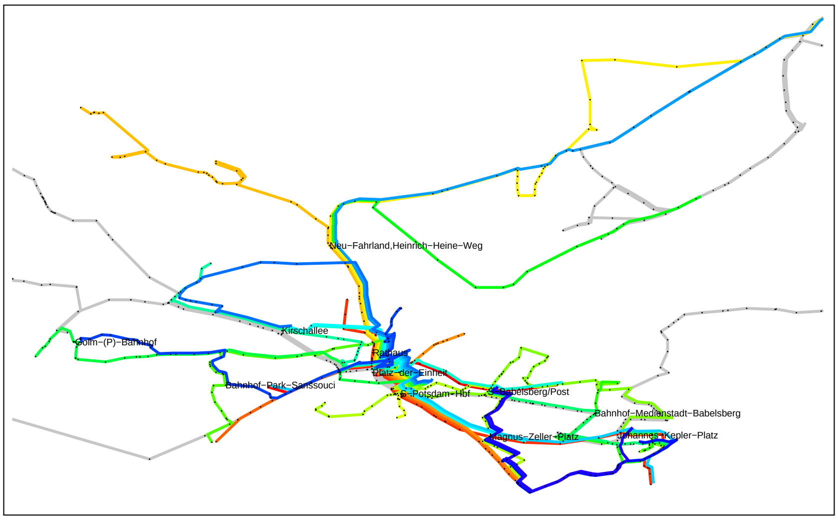


Figure 6: Visualization of line plan for Potsdam.

In particular, the line plan tries to meet the passenger travel demand and respect existing simplified network capacities and properties. Common, but obviously contradictory objectives of a line plan are the minimization of operating costs and the maximization of the service or travel quality. Travel quality or attractiveness of a line plan can be measured by the number of direct connections and travel times for passengers. But of course, the passenger satisfaction of a line plan mainly depends on the operated and experienced timetable implementing the line plan, see Schittenhelm (2009) [186].

Significant work on line planning can be found, for example, in Bussieck (1997) [49], and Goossens, van Hoesel & Kroon (2006) [101]. Later, novel multi-commodity flow models for line planning were proposed by Schöbel & Scholl (2006) [192] and Borndörfer, Grötschel & Pfetsch (2007) [35]. Its main features, in comparison to existing models, are that the passenger paths can be freely routed and lines are generated dynamically. From a general perspective these models are also "coupling" models. The line variables provide "capacities" that passenger flow variables utilized for transfers.

Properties of this model, its complexity, and a column-generation algorithm for its solution are presented and tested on real-world data for the city of Potsdam, Germany. A recent research field is the incorporation and handling of transfers, e.g., the change-and-go model of Schöbel & Scholl (2006) [192]. However, for large scale instances the model is hardly computational tractable.

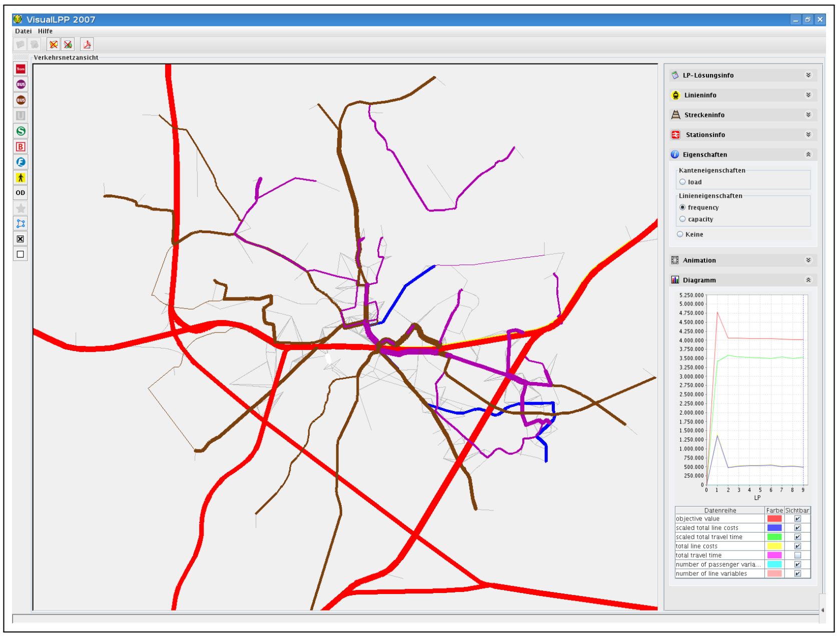


Figure 7: Screenshot of visualization tool for public transport networks.

Therefore, Borndörfer & Neumann (2010) [29] propose a novel "compact" integer programming approach to deal with transfer minimization for line planning problems even for larger instances. Therein they incorporate penalties for transfers that are induced by "connection capacities" and compare a direct connection capacity model with a change-and-go model. In Figure 6 a line plan for the city of Potsdam can be seen, each color represents one line.

Finally, the resulting line plan serves as a direct input for the periodic train timetabling problem, where valid arrival and departure times for the given lines and frequencies have to be found. However, the final decision of which transport mode a user chooses depends on the available options provided by the public transport network. Figure 7 shows the complete public transport network of the city of Potsdam, i.e., bus, tram, subway, and city railway.

6 Timetabling

The train timetabling problem has many names - such as train scheduling problem, train routing problem, or sometimes track allocation problem. The timetable, which is the solution of the train timetabling problem, is the heart of a public transportation system. In the end, this is the offer a railway undertaking presents to the passengers. In the case of a freight train operator, the corresponding train slots are the basis to implement and operate the transportation service.

It is a main problem of the planning process of railway traffic - simply because it asks for the efficient utilization of the railway infrastructure, which obviously is a rare good. In addition the service quality of an offered timetable depends directly on the concrete allocation. In a segregated railway system additionally the crucial interconnection between railway undertakings and infrastructure managers has to be taken into consideration.

Nevertheless optimization models and techniques are not that widely used for timetabling in practice in contrast to the subsequent resource planning problems, i.e., vehicle and crew scheduling. Most timetables are minor modifications of their predecessors, so that basically timetables are historically grown. One reason is that a timetable is, not only in Germany, a huge political issue. Whether a German city will get access to the system of long-distance passenger trains – high-speed trains that are connecting important cities – will be decided in elongated negotiations between the railway operator DB Fernverkehr, the federal state, and the German government, i.e., the Federal Ministry of Transport, Building and Urban Development (www.bmvbs.de). A prominent subject of dispute in the recent years was the rather small city Montabaur that got access to the ICE transportation network. In an idealized world network design planning for long-distance passenger trains would answer such questions and provide the input for the timetabling. In addition decisions on the service quality of an urban rapid transit system, e.g., the Berlin S-Bahn, will be preassigned and is mainly subsidized. Lobbying swayed the decisions more than the results of quantified analysis.

In the following sections we will focus on three different aspects of timetabling in more detail. Section 6.1 will discuss the ongoing deregulation of the European railway market. We give a brief literature review on periodic and individual trip train timetabling in Section 6.2. Finally, Section 6.3 will briefly discuss standard railway models of different scale.

6.1 European Railway Environment

Railway transportation services require very accurate planning of operation in contrast to other modes. This is due to the fact that railway undertakings have to promote their railway transportation services for passengers far prior to the actual railway operation. A published and only rarely annually changed train timetable allows the customer to use railway transportation services efficiently. Moreover, uncontrolled railway operation is particularly prone to deadlocks. Train drivers need to obtain the moving authority for a certain part of the railway infrastructure from a centrally authorized controlling instance, which assures a high level of safety. An annual initial schedule helps to control railway operation, since it reduces the vast complexity of real time operational planning. Nevertheless the liberalization and introduction of competition in the European railway system will break down these old-established and rigid structures in the near future. However, in comparison to airline transportation and urban bus transport the railway system is very rigid and hardly innovative.

Furthermore, railway systems consist of very expensive assets. In order to make best use of these valuable infrastructure and to ensure economic operation, efficient planning of the railway operation is indispensable. Mathematical optimization models and algorithmic methodology can help to automatize and tackle these challenges.

In 2009 there were 300 railway undertakings operating in the German secondary railway market, 60 of them do request railway capacity for passenger trains. From an economic perspective railway undertakings offer transportation services on the primary railway market. Thus, the market where railway capacity is traded is called secondary railway market.

However, DB Regio is still the biggest railway undertaking requesting railway capacity for passenger trains. In 2002 Deutsche Bahn AG established a “Competition officer”, in order to guarantee the correct implementation of the European framework for railway capacity allocation.

Within a competitive railway market, the train slot requests submitted by concurrent railway undertakings are more likely to conflict. This assumption is backed by current statistics of the competition reports of the German railway system. The number of conflicting trains slot requests climbs from 10.000 up to 12.000 from 2008 to 2009, i.e., that is an impressive increase of 20 %. In the same period the conflicts reported by the Trasse Schweiz AG for the allocation process in Switzerland increase from 103 to 127.

A detailed discussion of the legal environment of the European railway market can be found in Mura (2006) [164] and Klabes (2010) [129]. In there, all European directives and legal definitions are given as well as various references to the discussed statistics. We will summarize the most important facts; Article 18 of the EU Directive 2001/14/EC contains all relevant deadlines for the capacity allocation process in the European railway system. Of course, some flexibility is given to the national infrastructure managers. They can determine these deadlines within certain tolerances. However, they have to publish them so that they are available to all licensed railway undertakings to establish a fair and open-access market. The main regulations are listed in the following:

- ▷ The working train timetable shall be established annually.
- ▷ Infrastructure managers have to declare a specific date and time when the shift of one train timetable to the new one takes places.
- ▷ The final date for receipt of annual train slot requests must not be earlier than 12 months before the new timetable is operated.
- ▷ Not later than 11 month before the new timetable is operated the infrastructure managers shall ensure that the international train slot requests have been allocated provisionally.[2]
- ▷ Four months after the deadline for submission of the annual train slot requests by railway undertakings, a draft timetable shall be prepared.

Furthermore, four types of slot request are to be distinguished:

- ▷ long term train slot requests,
- ▷ international train slot requests,
- ▷ annual train slot requests,
- ▷ and ad hoc train slot requests.

The planning time horizon, which is the time period between the date when a train request is submitted and the date when the train path request is included into the working timetable, are from 5 up to 15

[2]The allocation of international train slot requests should be adhered to as far as possible, because at least two different national railway infrastructure managers and one railway undertaking are involved.

years in case of long term slot requests. This shall insure reliability for the future planning of railway infrastructure managers and railway undertakings by so called framework agreements. International train slot requests require capacity from at least two different international railway infrastructure providers. Annual train path requests have to be submitted annually to be included into the annual timetable. They can be requested until a deadline that can be determined by the infrastructure manager, usually 8 months before the new timetable is operated. Due to the necessary cooperation between the concerned national infrastructure managers an independent organization, `RailNetEurope` (`www.railneteurope.com`), was set up. International train slot requests are directly submitted to `RailNetEurope`, which is responsible for the coordination between the involved national infrastructure managers.

Ad hoc train slot request are, as the name already suggest, submitted at short notice. In particular this applies to cargo trains which are planned in a much more flexible way than passenger trains. Such train slots are requested from two weeks to 24 hours in advance. In Figure 8 only the beginning of ad hoc requests concerning the new annual timetable is shown. Ad hoc requests for the actual timetable are of course possible at any time.

Most infrastructure managers already plan suitable train slots, sometimes called system slots, in advance without binding them to a specific railway undertaking. In case of ad hoc slot requests or individual slot requests in the course of the year such anticipated system slots can be assigned. Deciding how much capacity should be reserved a priori for those ad hoc requests is by no means trivial. Of course, this is also done due to the complex planning even for the case of only one additional single slot. We see a huge potential to support this task by optimization models and algorithms. A reliable track allocation model and solver could easily analyze the effect of adding another slot without the price of time-consuming simulation runs. Moreover we will present a general approach that guarantees the re-transformation of the optimization results into the simulation frameworks.

The procedure of capacity allocation is illustrated in Figure 8. The deadlines denoted by $x - 11$ and y, as well as the interaction between railway undertakings (RU) and infrastructure managers (IM) can be seen. The first month of operation of the timetable is denoted by x. In addition we highlight the stage where the infrastructure managers

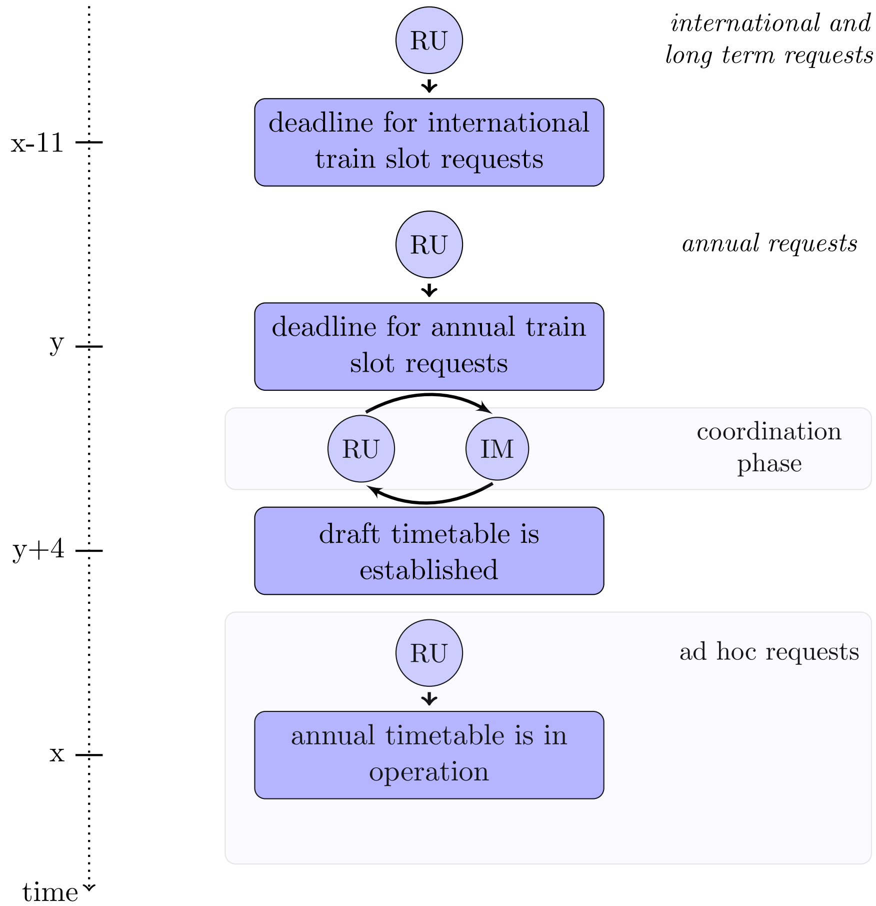


Figure 8: Timeline for railway capacity allocation in Europe, source: Klabes (2010) [129].

have to solve track allocation problems. Of course the international, long term and the annual requests can also be planned at the point of submission, but conflicts at that time are very rare. In the end of the process a working (annual) timetable or track allocation is determined. Therefore, the names train timetabling and track allocation problem are used for essentially the same problem, only the point of view differs. On the one hand, railway undertakings are interested in their accepted slots to offer a suitable timetable for their various purposes. On the other hand, infrastructure managers are interested in a high and stable utilization of the network by the complete allocation of all railway undertakings. Finally, long term, international and annual requests are considered in a draft train timetable at $y + 4$. Due to the limited railway infrastructure capacity the occurrence of conflicts is very likely, especially in highly frequented parts or bottlenecks. However, in the coordination phase of the railway capacity allocation process all conflicts have to be resolved. This is were optimization can significantly support the planning process. Even more is required by most European directives and laws: In Germany §9 passage 5 of the Regulation for the use of railway infrastructure, see Federal Ministry of Transport & Housing (2005) [86], states:

> "The network provider has to compare the charges to decide between equally ranked types of traffic under the terms of passage 4. In case of a conflict between two train slot requests the one with the higher charge takes or has priority, in case conflicts between more than two train slot requests the allocation or choice with the highest charge in total takes or has priority." (translation by the author).

In a first step, the infrastructure managers try to resolve the occurring conflicts as best as they can. In particular, slot requests that are involved in conflicts are altered. Of course, when realizing an exact optimization approach with all "degrees of freedom" it can occur that the best decision affects also slots that are not directly in conflict before. In Figure 9 a trivial situation is shown. Each line represents a train run on track j from left to right, i.e., the boxes on the sides represent the connecting stations. Imagine that the first and the last train (blue) are already scheduled, and the other train (red) requested to run on j at the depicted time. On the left hand side one can see that only the last two trains are in conflict on j, i.e., the crossing of both lines symbolizes a "crash" at that time. As a result, sticking exactly to the requested times leads to a schedule with maximal two trains. However, on the right hand side one can see a solution that allows to run all trains by choosing slightly earlier departure times for the first ones. In fact, we assume that the slot contracts for the train slots allow for the propagted departure shift, i.e., we choose an arbitrary safety distance to avoid crossings.

This requires the coordination and cooperation between railway infrastructure managers and all those railway undertakings whose train paths need to be altered. Usually at the end of this process a conflict free draft timetable is determined. However, in some cases train slot requests are rejected in the coordination phase. It is clear that there is some discrimination potential and therefore independent agencies are in charge of controlling these procedures, e.g., in Germany

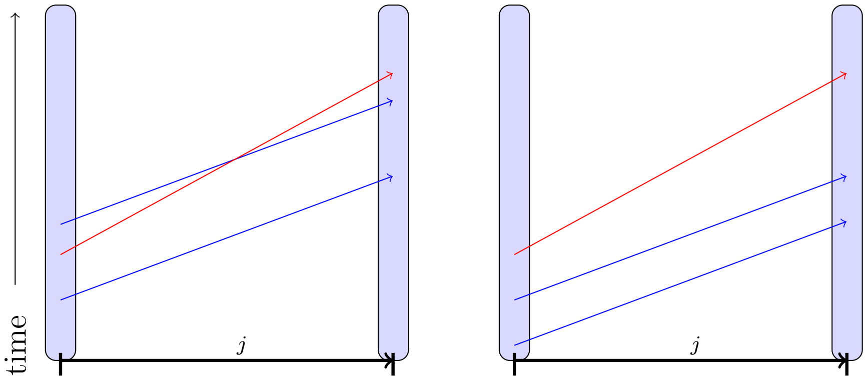


Figure 9: Simple conflict example and re-solution for track allocation.

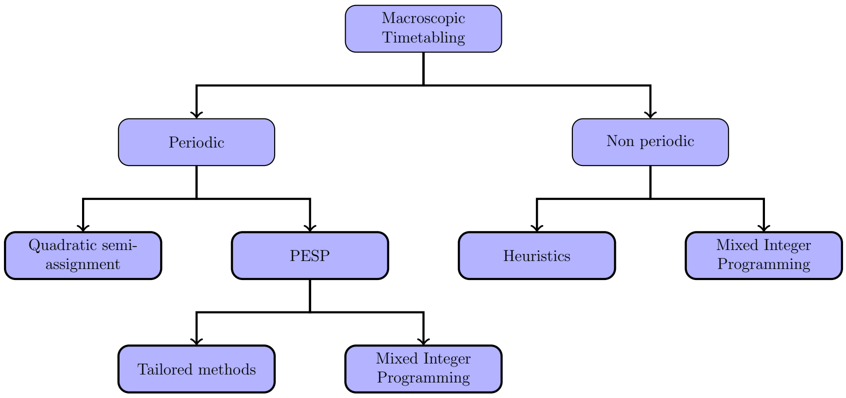


Figure 10: Principal methods in the literature for macroscopic timetabling by Caimi (2009) [57].

the Federal Network Agency (Bundesnetzagentur), see http://www.bundesnetzagentur.de.

6.2 Periodic versus Trip Timetabling

Lusby et al. (2009) [159] give a recent survey on the track allocation problem and railway timetabling. Nevertheless we want to enlighten some aspects and present a general classification according to solution methods used by Liebchen (2006) [148] and Caimi (2009) [57]. In Figure 10 the approaches on macroscopic railway timetabling are basically divided into two categories, periodic and non-periodic scheduling.

6.2.1 Periodic Timetabling

Periodic timetables are first and foremost used for passenger traffic. Even if there are some works on quadratic semi-assignment models, e.g., Klemt & Stemme (1988) [131], most authors consider another model, the Periodic Event Scheduling Problem (PESP). It is a powerful and well-studied model for macroscopic scheduling. Serafini & Ukovich (1989) [199] introduced a general version and Schrijver & Steenbeck (1994) [194] applied it at first to train scheduling. Since that time the PESP has been intensively studied and many extensions and variants were presented, see Odijk (1997) [169], Lindner (2000) [154], Kroon & Peeters (2003) [141], Kroon, Dekker & Vromans (2004) [142], and Liebchen & Möhring (2004) [150]. The PESP model was successfully applied as the core method for the generation of the 2005 timetable of the Berlin underground, see Liebchen (2006) [148] and Liebchen (2008) [149], and for the generation of the 2007 railway timetable in the Netherlands Kroon et al. (2009) [140]. Furthermore commercial software, e.g., TAKT, see Nachtigall & Opitz (2008) [165], based on the PESP model was developed and entered the market. The degrees of freedom for PESP are on a global interacting level between the trains. It is always assumed that the route or path is already decided, i.e., all headway parameters are calculated under this fixed assumption, as well as the connection times inside the stations. Furthermore it is expected that all trains can be scheduled with respect to their frequencies, otherwise the complete problem is stated to be infeasible. This disadvantage of the model formulation was for a long time negligible due to sufficient capacity for appropriate scenarios. Obviously from an optimization point of view this has to be revisited and at least feedback on locals conflicts has to be given, which is one particularity of TAKT.

Recent research work focuses on the integration of robustness aspects, see Odijk, Romeijn & van Maaren (2006) [170], Kroon et al. (2006) [139], Cacchiani et al. (2008) [53], Liebchen et al. (2009) [152], Liebchen et al. (2010) [153], and Fischetti, Salvagnin & Zanette (2009) [91] as well as integration of flexibilities to improve the interaction between macroscopic and microscopic scheduling, see Caimi (2009) [57] and Caimi et al. (2007) [59]. The contributions of Caimi (2009) [57] are mainly in the area of integrating and improving the interaction between microscopic and macroscopic models for planning passenger traffic. The idea and goals can be found in Burkolter, Herrmann & Caimi (2005) [48]. For example the extension of the PESP to flexible event times (FPESP) allows for more degrees of freedom in the subsequent microscopic scheduling.

The (passenger) timetable itself is the core of all railway activities. From a historical and from a customer point of view national railway operators offer almost exclusively periodic timetables for passenger traffic. On the one hand this is much easier to remember and recall for passengers and on the other hand the whole process of determining a valid timetable becomes much easier, i.e., the planning of all system-oriented components like infrastructure, rolling stock and crews. Furthermore most people expect symmetric transport chains, if they make a round trip. An historical overview is given in Figure **??**, which demonstrates the dominance in European subway and railway systems today. Summarizing, a periodic timetable is easy to use, easy to understand, and easier to operate.

However, Borndörfer & Liebchen (2007) [28] showed in a theoretical work that periodic timetables can become inefficient compared to trip timetables from an operator point of view. Sub-optimality and inefficiency of periodic timetables are accepted and well known. Even more specializations such as *synchronized periodic timetables* (ITF) are popular in practice and usually used for passenger traffic. A synchronized periodic timetable is a periodic timetable that additionally provides reasonable transfer times at certain stations.

In our rapid growing information society the reasons for periodicity could become negligible in the future. The development in traffic engineering of traffic management systems will bring more and more helpful real-time information to the passengers as well as to the operators. The necessity of easy manageable timetables will then cease to apply in the future. If an acceptance for non-periodic and fully individual or demand dependent timetables increases, railway operators could offer much more efficient timetables. A trend which can already be observed for large public events in sports, music and so on. Deregulation and competition will assist this development as well.

In a future world of full and real-time available information passengers will not be insistent that trains have to be scheduled with a fixed cycle period. More important will be that the timetable covers the demand efficiently and reliably. The frequency in peak hours has to be higher, but it will not be mandatory that departure and arrival times will follow an exact periodic pattern as long as enough connections are provided. The service quality experienced by the passenger depends more on the reliability of the service, i.e., the deviation between expected waiting times and real waiting times.

Let us discuss timetabling from a passenger traffic perspective. The line planning determines passenger lines with their frequencies for different demand periods, i.e., the lines can be different in peak hours or on weekends. The task of timetabling is now to define exact arrival and departure times, e.g., in minutes, at each station of the lines. It is clear that the requirements and constraints are somehow different to the ones of freight traffic, especially in contrast to long-distance railway services. Passenger trains have, in general, a fixed stopping pattern with respect to the line definition and of course a tight dwell time interval to fulfill. One the one hand maximum dwell times are needed to offer passengers fast services. On the other hand they have to be at least large enough to allow for transfers, i.e., desired and favorite connections of different lines at certain major stations. For freight railway traffic the situation is different and other aspects mainly affect the service quality, e.g., required arrival times at certain stations and long possession times are needed to perform shunting and loading activities. The costs for a freight train are much more unpredictable due to the fact that braking, unforeseeable stops and acceleration have a huge effect on the energy consumption and the total running time.

6.2.2 Non periodic Timetabling

For networks where freight traffic is predominant and for freight traffic in general non-periodic macroscopic timetables are broadly used. Already in the 1970s Szpigel (1973) [206] studied this problem and proposed a mixed integer programming formulation. Later many techniques like constraint programming by Silva de Oliveira (2001) [201], Oliveira & Smith (2001) [171], and Rodriguez (2007) [183], artificial intelligence approaches by Abril, Salido & Barber (2008) [2], and resource constrained formulations by Zhou & Zhong (2007) [217] were applied. Problem or even case specific heuristic approaches were developed, e.g., Cai & Goh (1994) [55], Cai, Goh & Mees (1998) [56], Higgins, Kozan & Ferreira (1997) [115], Dorfman & Medanic (2004) [76], Ghoseiri, Szidarovszky & Asgharpour (2004) [98], Semet & Schoenauer (2005) [198], Lee & Chen (2009) [146], and Zheng, Kin & Hua (2009) [216]. However, the most popular and successful solution approaches are integer programming based formulations as proposed in the seminal works of Brännlund et al. (1998) [44] and Caprara et al. (2006) [63]. The most important advantage of exact optimization approaches is that in addition to solutions also a guarantee on the solution quality is given. This allows for precise estimations on optimization potential for the various planning challenges.

Freight transportation is innately non-periodic – a large number of operated freight or cargo trains are even not known at the beginning of the timetable planning process. Only for some standardized types of cargo trains slots will be allocated or reserved - later these slots will be assigned to the real operating trains and an adaption of the schedule has to be done. The reason is that the exact weight and length of a train, which is committed only a short period before the day of operation, is needed to compute realistic running times. Thus, this can lead to some minor changes of the scheduled departure and arrival times of these trains and probably also for other trains due to safety margins and headway times. Modeling the railway safety system will be described in detail in Chapter II.

One of the earliest publications on the optimization of trip train schedules is from Szpigel (1973) [206]. The focus of his work is a long single track railroad in eastern Brazil which is used by trains to transport iron ore in both directions. The line is divided into a number of track sections, with each track section linking two stations. In stations additional tracks are available to allow trains to stop or overtake each other. The main contribution of the author is to identify strong similarities between train scheduling problems and the well known job-shop scheduling problem. In the train scheduling context trains can be seen as jobs. They require the use of several track sections, that are the machines, to complete their designated route. To prevent track sections from hosting more than one train operation at any given time *ordering constraints* are introduced. Finally, he solves the problem with a branch and bound approach until reaching a feasible meet and pass plan. Nowadays we would call this method a lazy constraints approach that ignores the ordering constraints in the linear relaxation and then branch if the solution contains trains in conflict. However, models and techniques presented in that work for a simple single line are the basis of considering complicated routing situations.

Later enumeration based methods were used by Sauder & Westerman (1983) [185] and Jovanovic & Harker (1991) [125] to construct feasible meet and pass plans based on a MIP approach. To the best of our knowledge the model and algorithm of Jovanovic & Harker (1991)

[125] was the first one, which leads to a software system that already includes a simulation modul to work with reasonable times for the train movements.

Carey & Lockwood (1995) [65] consider an almost identical network to that of Szpigel (1973) [206], but propose a different modeling and solution approach. The authors present a large MIP formulation similar to that of Jovanovic & Harker (1991) [125]. Each binary decision variable controls the order of a pair of trains on a given track section.

Cai & Goh (1994) [55] propose a simple greedy heuristic for the same problem. The heuristic considers trains in chronological order and assumes that the start time and location are known. Later in Cai, Goh & Mees (1998) [56] the authors extend their work to the case that the initial location of a train is fixed. A successful implementation of the algorithm is reported for an Asian railway company, where up to 400 trains run per day with as many as 60 trains in the system at any given time.

Brännlund et al. (1998) [44] introduce the notion of *packing* constraints to restrict the number of trains using any track or block section to at most one instead of control the order explicitly. This work can be seen as the first resource based model approach to the track allocation problem. The authors propose a set packing integer programming formulation to solve the problem for a bidirectional single line connecting 17 stations in Sweden. An acyclic time-space network consisting of different arc types is use to model each train's movement. Paths in the time-space network reflect different strategies for the associated train to complete its itinerary. The scheduling horizon is discretized into intervals of one minute each. The objective is to maximize the profits of the scheduled trains with a penalty for unnecessary waiting times. The author suggests to solve the problem with Lagrangian relaxation techniques. After relaxing all packing constraints the problem decomposes into n independent subproblems where n is the number of trains. To construct integral solutions a train priority based heuristic is used and performs well for the considered instances, i.e., solutions with an optimality gap of only a few percent are reported. A comprehensive survey of optimization models for train routing and scheduling up to the year 1998 is given by Cordeau, Toth & Vigo (1998) [70].

Caprara et al. (2001) [61] and Caprara, Fischetti & Toth (2002) [62] further developed the graph theoretical formulation using an event activity digraph. In addition the authors proved that the classical stable set problem can be reduced to TTP, such that the problem is $\mathcal{NP}$ hard. Indeed, the optimal track allocation problem can be seen as a problem to a find a maximum weight packing with respect to block conflicts of train routes in a time-expanded digraph. This framework is fairly general, see further articles by Cacchiani, Caprara & Toth (2007) [52], Cacchiani, Caprara & Toth (2010) [54], Fischer et al. (2008) [90] and Cacchiani (2007) [51] for comprehensive discussions how such a model can be used to deal with various kind of technical constraints.

Finally, Table 2 lists the sizes of the largest instances solved so far by the various authors. The research of Fischer et al. (2008) [90] and Fischer & Helmberg (2010) [89] focus primarily on solution techniques for relaxations of the problem, i.e., we marked scenarios for which only heuristic solutions are reported. However, a fair comparison is not only complicated by the different scale of the models. In particular, Lusby (2008) [158] and Klabes (2010) [129] consider microscopic railway models. In fact, several additional parameters determine the degrees of freedom and the computational tractability of any TTP instance. Here is a short list of the most important ones:

- ▷ routing possibilities within the network,
- ▷ discretization of time,
- ▷ selection of train types,
- ▷ options for running times,
- ▷ time windows of arrival and departure events,
- ▷ complexity of the objective function,
- ▷ and flexibility to let trains stop and wait.

6.2.3 Conclusion

We conclude with the vision that train schedules will be become more and more flexible in the near future. Information systems and state of the art optimization techniques will allow track allocation problems to be solved for real world application. Hence, infrastructure managers will be able to improve the solutions of the coordination phase. More scenarios can be handled and additional cargo requests or ad hoc request will be answered much faster. That will lead to a more efficient utilization of the infrastructure. Even a completely different handling and marketing process of ad hoc requests is imaginable to take advantage of the new allocation possibilities. Furthermore, railway operators will be able to react faster on major demand changes in passenger

reference	#stations	#tracks	#trains
Szpigel (1973) [206]	6	5	10
Brännlund et al. (1998) [44]	17	16	26
Caprara, Fischetti & Toth (2002) [62]	17	16	221
	102	101	41
Cacchiani, Caprara & Toth (2007) [52]	17	16	221
	102	101	41
Cacchiani, Caprara & Toth (2010) [54]	65	64	775
Fischer et al. (2008) [90]	104	193	251
Fischer & Helmberg (2010) [89]	104(445)	193(744)	137
Fischer & Helmberg (2010) [89]	*1776*	*3852*	*3388*
Lusby (2008) [158] (microscopic)	≈ 120	524	66
Klabes (2010) [129] (microscopic)	*2255*	*2392*	*32*
Chapter IV, Section 4 (microscopic)	1154	1831	390
Chapter IV, Section 4	18	40	390
Chapter IV, Section 1	37	120	1140

Table 2: Sizes of the solved instances in the literature for the TTP instance.

transportation, i.e., the offered timetable will be more flexible. One prediction, for instance, is that innovative railway infrastructure managers will be able to construct creative solutions and hence, will be able allocate “more” train slots. As a result railway operators will more and more rely on ad hoc slots and also become more flexible in designing their timetables and their operations. However, we propose that the railway system needs some time to implement this flexibility. We rather assume that primarily railway infrastructure managers will use mathematical optimization models to evaluate more strategic and tactical planning questions concerning track allocations.

The highly dynamic aviation environment is the perfect role model of a free market, where the competitors have to satisfy the customers demands and have to anticipate innovation potential - otherwise the competition will squeeze them out of the market. The ongoing European liberalization of railway traffic will support this process. It is not clear that this process can be successfully finished and “real” competition will be introduced – however railway transportation has to find its way to establish efficient offers to compete with the other transportation modes. The integration of state of the art mathematical modeling and optimization techniques can immediately support the allocation process of railway capacity.

6.3 Microscopic versus Macroscopic Models

The level of detail of a railway infrastructure or operation model depends on the quality and accuracy requirements for generating appropriate results and, of course, on the availability and reliability of the data. For long term and strategic planning problems high accuracy data is often not manageable, might not exist or can not be provided on time without causing expenditure, e.g., Sewcyk (2004) [200]. In addition it makes no sense to deal with highly detailed railway models, if the question to answer will relate only on some parameters. A prominent example is timetable information where neither the railway infrastructure or the rolling stock have to be observed precisely. Moreover formal and legal reasons might prohibit free access to highly detailed infrastructure data that are classified as essential facilities by some European railway infrastructure managers. These are reasons why models of different scale has been established:

- *Microscopic models* require high detailed data to produce reliable and high quality results, i.e., for running time calculation and the simulation of timetables and railway operations.
- *Mesoscopic models* are produced if no microscopic data is available, standard assumptions are made for missing microscopic elements. They are used in most eastern European countries that do not want to put a lot of effort in generating and maintaining a microscopic database.
- *Macroscopic models* embrace coarse and aggregated structures; real-world applications are vehicle circulation, long term traffic planning, strategic infrastructure planning, and travel information systems.

Obviously, optimization on a microscopic level is still inconceivable due to the enormous size and granularity of the data. Even more, it is not necessary because the decision to run a train or let a train wait can be done on a macroscopic level that is based on microscopic evaluations. For example, all macroscopic running times are deduced by microscopic simulation data, assuming a standard acceleration and braking behavior of the standard train compositions. Thus all relevant switches, inclines, curves or other velocity impacts are considered implicitly.

The literature has suggested a number of top-down approaches, e.g., Klemenz & S.Schultz (2007) [130] and Caimi (2009) [57]. In a top-down approach to model railway systems an overview of the entire system is first formulated, specifying but not detailing any "real" sub-systems. A top-down model is often specified with the assistance of "black boxes". However, black boxes may fail to elucidate elementary mechanisms to realistically validate the model. Solving track allocation problems is only useful if the railway system is modeled precisely with respect to resource consumptions, i.e., the calculation of running and headway times must be incorporated in detail.

The focus of Chapter II will be to develop a novel bottom-up approach for automatic construction of reliable macroscopic railway models based on very detailed microscopic ones. We will start with a realistic microscopic railway model that indeed might be too large to be solved in reasonable time to optimality. However, this model could be simplified and aggregated by well defined rules and error estimations, i.e., running and headway times are incorporated almost exactly. This approach turns out to be more reliable and thus more convincing than contrary top-down approaches that try to integrate more and more details in weak and questionable base models.

7 Rolling Stock Planning

The goal of the rolling stock planning, the vehicle scheduling problem or the aircraft rotation problem is to find a cost minimal assignment of rolling stock, vehicles or aircrafts to the trips stemming from the timetabling. Input for the rolling stock planning are the timetabled trips and the possible deadhead trips of the vehicles, the rolling stock or the aircrafts. The timetabled trips are the trips that transport passengers. Deadhead trips give the possible concatenation of timetabled trips into rotations. The set of timetabled trips and deadhead trips together is simply called *trips*. Each trip has a start- and end-time and a start- and end-location, further we need to know the length and the driving time of each trip. The problem naturally give rise to a rolling stock scheduling graph. That is a standard event activity digraph representing space and time. In the following we want to discuss the special problem of vehicle scheduling (VSP) in urban public transport. The cost of a vehicle schedule is composed of a fixed cost per used vehicle, cost per driven distance, and cost per time away from a depot of a vehicle.

An extensive literature survey of the VSP until 1997 can be found in Kokott & Löbel (1997) [135], Kliewer, Mellouli & Suhl (2006) [132], and Steinzen et al. (2010) [203].

The set of available vehicles is called a fleet. The maximum number of vehicles used can be a constraint of the VSP or be part of its result. Each vehicle has a unique vehicle type. Typical vehicle types in case of bus traffic are standard bus, double decker, or articulated bus. Each vehicle type has a set of characteristics which is relevant for the planning process, such as the number of seats, an average speed, minimum maintenance intervals, or maximum length of covered distance without refueling. Not all vehicle types are able to service all trips. For instance, long buses cannot go around narrow curves, double deckers may not pass low bridges, or a larger bus is preferred for trips with high passenger volume. Each vehicle of a fleet is associated with a unique garage at a certain location. Each garage contains vehicles of varying types in certain quantities. We call a vehicle type/garage combination a depot. We may have a maximum number of vehicles of certain types per garage or in total. These numbers are called capacities of the depots or vehicle type capacities. Obviously, similar restrictions are given in case of planning aircraft rotations or rolling stock rotations.

A rotation, sometimes also called block, is an alternating sequence of deadhead and timetabled trips that begins and ends in the same depot. Rotations can be combined to courses. A course is a set of rotations that can be driven by a single vehicle. We call a set of courses that covers all timetabled trips a vehicle schedule.

State of the art solution methods for large real-world instances of the vehicle scheduling problem are either based on Lagrangian relaxation heuristics, see Kokott & Löbel (1997) [135], or by heuristic preprocessing and solving the resulting problem by standard MIP solvers as proposed by Kliewer, Mellouli & Suhl (2006) [132]. Finally, Figure 11 shows a partial vehicle scheduling graph for a rolling stock scenario, i.e., only the passenger trips are visualized as arcs in a standard week.

8 Crew Scheduling

The crew scheduling problem arises not only in railway traffic, but also in urban public transport and airline transportation. From a practical point of view these problems may all differ in their structure, needs,

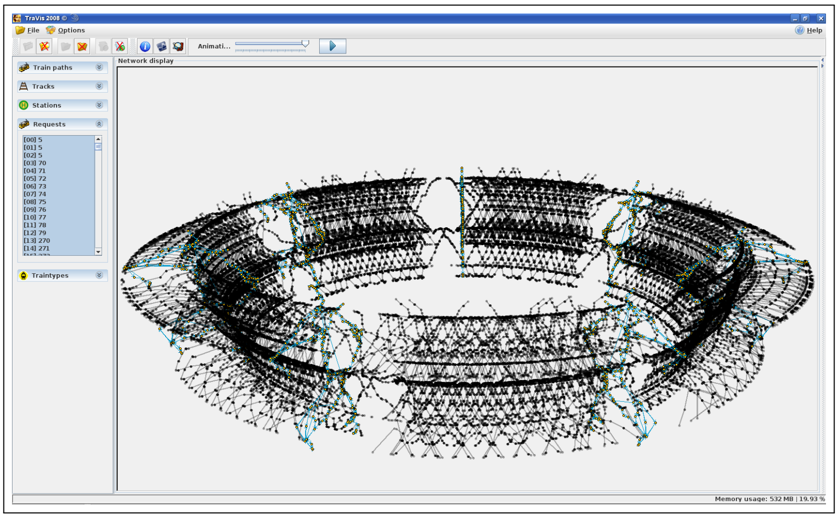


Figure 11: A partial cyclic rolling stock rotation graph visualized in our 3D visualization Tool TRAVIS using a torus to deal with the periodicity.

rules and especially their sizes. From a theoretical mathematical point of view they can be formulated as a general model and solved by equivalent techniques with a proven optimality gap for almost all practical relevant sizes - even for very large scale instances.

That is one reason why we will discuss this problem in the following paragraphs. Another one is that the author gathered many valuable experiences in solving large-scale airline crew scheduling problems in practice. The corresponding mathematical optimization model and some key constructions are shown in detail. Finally, the general algorithmic solution approach is presented.

8.1 Airline Crew Scheduling

We refer to Barnhart, Belobaba & Odoni (2003) [20] for an overview on airline optimization in general and on airline crew scheduling. Operational cost for crews are a huge cost factor for every aviation company in the world. Complex rule systems by the government as well as by specific labor unions, home-base capacities and balancing requirements to support the subsequent rostering process lead to very large scale combinatorial optimization problems. The goal is to find a cost minimal set of duties which cover all relevant *legs*, i.e., the planned flights of the airline, and fulfills all *home-base* capacities.

We denote the set of relevant legs by $\mathcal{T}$ and the set of home-bases, that are locations of available crews, by $\mathcal{H}$. We partition all possible duties or *crew pairings*, as it is called in the airline industry, with respect to their home-bases, i.e., the start and end location of a pairing must be the same. Let $\mathcal{P}$ be set of all pairings with $\mathcal{P} = \cup_{h \in \mathcal{H}} \mathcal{P}_h$.

8.2 Crew Scheduling Graph

The crew scheduling problem can be described in terms of an acyclic directed network $G = (V, A)$. The nodes of G are induced by the set of *timetabled flights*, in railway or bus application by the set of timetabled trips. These are tasks $t \in \mathcal{T}$ that has to be performed by personnel in a feasible crew schedule. Additionally there are nodes s and t, which mark the beginning and the end of pairings; called sink and source nodes of G. Supplementary tasks can also be considered in G, such as flight transport, also called deadheads, or ground transport. We will later discuss how to handle them implicitly a posteriori.

The arcs A of G are called links; they correspond to possible direct concatenations of tasks within pairings. In addition there are artificial links that model valid beginnings or endings of pairings. An arc $(u, v) \in A$ represents the consecutive processing of task v after u by a pairing, therefore local rules with respect to time and location, e.g., minimal transfer times or ground times, can be handled by the construction of the graph, i.e., by the definition of the arc set. However, most of the pairing construction rules concern the complete pairing, such as maximal landings per pairing, minimal and maximal flight time, minimal number of meal breaks and many more. We denote by $\mathcal{R}$ the set of consumption rules and U_r the upper limit. An easy example for such a graph is given in Figure 12.

Each feasible pairing corresponds to a path in G. Unfortunately, some paths may violate the construction rules, i.e., assume in example graph shown in Figure 12 a maximal number of landings of at most two, then the path $p = \{(s, A-B), (A-B, B-C), (B-C, C-A), (C-A, t)\}$ is infeasible. We will come back to details on pairing generation in Section 8.5 after formulating the crew scheduling problem as an set partitioning problem.

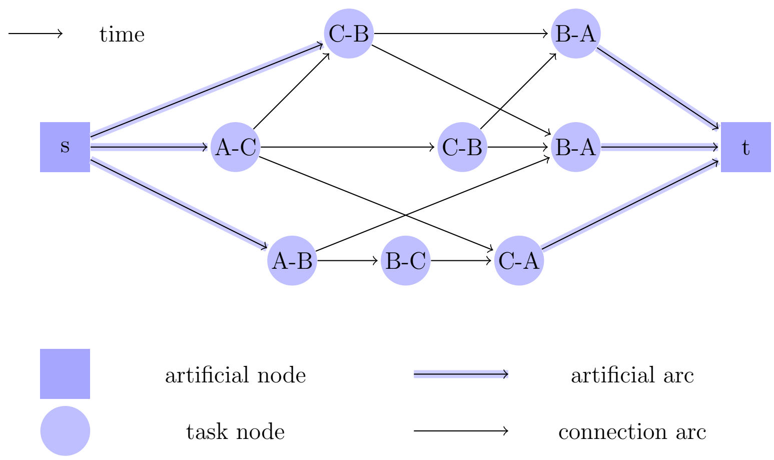


Figure 12: Crew Scheduling Graph.

8.3 Set Partitioning

We introduce a binary decision variable x_p for each pairing $p \in \mathcal{P}$, which is 1 if pairing p is chosen or 0 otherwise. To each pairing, which is nothing other than a sequence of tasks (and additional elements like deadheads, ground transports, meal breaks etc.) We denote by c_p a cost value. If we have restrictions on the number of available crews on a home-base h, we introduce a so called base constraint and an upper bound κ_h. Obviously, this is the most simple case of a base constraint. There are much more complex rules per day and per pairing type or even balancing requirements which can be handle in reality. Although this leads to base constraints, we skipped the details on that for simplification. We refer to Borndörfer et al. (2005) [33], there the definition of general linear base constraints with arbitrary coefficients is shown in detail to synchronize crews by using base constraints. In addition, we report in that paper on the solution of real world instances for crew scheduling with some thousands tasks. Moreover, our algorithmic kernel has been integrated in the planning system NetLine/Crew of the software company Lufthansa Systems GmbH. In Figure 13 a screenshot of the planning tool NetLine/Crew of Lufthansa Systems GmbH can be seen.

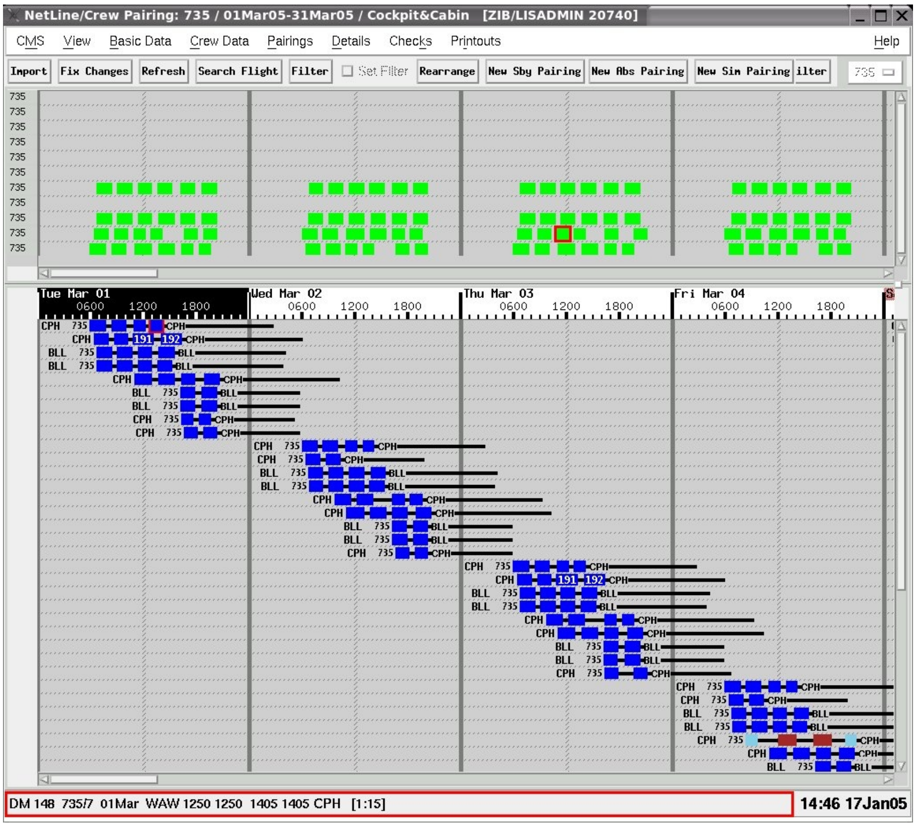


Figure 13: Set of legs (above) and a set of covering pairings (below) show as a Gant chart in the planning tool NetLine.

$$
\begin{array}{llllll}
\text{(SPP)} & \text{(i)} & \min & \sum\limits_{p\in\mathcal{P}} c_p x_p & & \\
 & \text{(ii)} & & \sum\limits_{p\in\mathcal{P},t\in p} x_p & = \mathbf{1}, & \forall t\in\mathcal{T} \\
 & \text{(iii)} & & \sum\limits_{p\in\mathcal{P}_h} x_p & \leq \kappa_h, & \forall h\in\mathcal{H} \\
 & \text{(iv)} & & x_p & \in \{0,1\}, & \forall p\in\mathcal{P}
\end{array}
$$

The objective function (i) minimizes the sum of pairing costs. Constraints (ii) ensure that each task $t \in \mathcal{T}$ is covered by exactly one pairing p. To ensure feasibility, we can assume that there is a "slack" pairing type with single-leg parings of high cost M.

Sometimes it is also possible to relax these to covering constraints. This allows more than one pairing to contain each task. Then in a post-processing step the decision of which crew really processes the task and which is only using it as a flight transport has to be taken.

But we want to point out that this can only be done if this change does not violate the pairing construction rules, e.g., a number of maximal flight transports can not be controlled anymore and may be violated. That no homebase capacity κ_h will be exceeded, is guaranteed by constraints (iii). Finally, we require that each variable x_p is integer to get an implementable crew schedule.

8.4 Branch and Bound

Ignoring the integrality constraints (SPP) (iv) will lead to a well known, linear programming relaxation, which we denote by (MLP). This model is used to derive a strong lower bound on the optimal value. Unfortunately, the solution of the relaxation can and will probably be fractional so that we have to divide the problem into several subproblems. The construction of the branches has to ensure that the optimal solution of (SPP) will be feasible in at least one new subproblem. The linear relaxation bound of the subproblems can only increase due to the new domain restrictions. A good branching decision is a crucial point in solving integer programs, i.e., for (SPP) constraint branching, proposed by Foster & Ryan (1991) [92] is much more effective than single variable branching. Another successful branching rule for these kind of problems is to choose a large subset of variables to fix to one based on perturbation techniques, see Marsten (1994) [160], Wedelin (1995) [211] and Borndörfer, Löbel & Weider (2008) [37]. This can be seen as *diving heuristic* trying to evaluate different parts of the branch and bound tree in a strong branching flavor to detect a so called *main branch*. In Chapter III and Section 3, we will highlight this idea in more detail and utilize it to solve large scale track allocation instances.

8.5 Column Generation

Unfortunately, the number of possible pairings $p \in \mathcal{P}$ is too large, even to write down the model (MLP). Only for a small number of tasks to cover it may be possible to enumerate all pairings. However, we can solve this optimization model by using a sophisticated technique called column generation. The idea was first applied to the crew pairing problem by Barnhart et al. (1998) [18] and is as simple as effective. Let us therefore recapitulate the main steps of the simplex algorithm to solve linear programs. During the simplex algorithm a solution of a

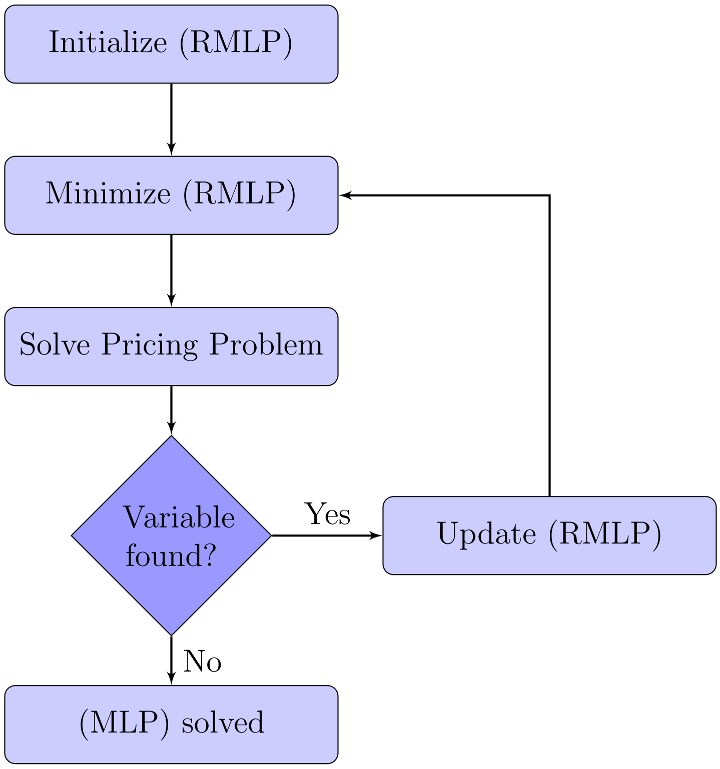


Figure 14: General column generation approach to solve LPs with a large column set.

linear program will only be improved if a non-basic variable with negative reduced cost can be added to the basis (in case of a minimization problem). This pricing step can also be done without constructing all variables or columns explicitly. Let us start with an appropriate subset of variables, then the linear relaxation, denoted by restricted master (RMLP), is solved to optimality. Only a non-considered variable can improve the current solution of the relaxation - if we can show that there is no variable left with negative reduced cost; we have proven optimality for (MLP) without even looking at all variables explicitly. Due to the fact that we add the necessary variables, columns of (RMLP), step by step this procedure is called dynamic column generation. The success and efficiency of such an approach is closely related to the complexity and capability of solving the pricing step in an implicit manner.

Denoting by (π, μ) a given dual solution to (RMLP), where π is associated with the partitioning (MLP) (ii) and μ with the (home-)base constraints (MLP) (iii), the pricing question arising for the master problem (MLP) is:

$$(PRICE)\ \exists h \in \mathcal{H}, \overline{p} \in \mathcal{P}_h : \overline{c_{\overline{p}}} = c_{\overline{p}} - \sum_{t \in \overline{p}} \pi_t + \mu_h < 0.$$

We assume that $c_p = \sum_{a \in p} c_a$. As all pairings end in the non-leg task t, we can define the reduced cost of an arc $(u, v) \in A$ w.r.t. (π, μ) as

$$c_{(u,v)} := \begin{cases} c_{(u,v)} - \pi_v & v \in \mathcal{T} \\ c_{(u,v)} + \mu_h & v = t. \end{cases}$$

The pricing problem to construct a pairing of homebase h (and type k) of negative reduced cost becomes a constrained shortest path problem in the acyclic digraph $G = (V, A)$ (restricted to homebase h and rule set of type k):

$$\begin{array}{lllrcll}
\text{(RCSP)} & \text{(i)} & & \min \sum_{a \in A} \overline{c_a} x_a & & & \\
& \text{(ii)} & \sum_{a \in \delta_{\text{out}}(v)} x_a - & \sum_{a \in \delta_{\text{in}}(v)} x_a & = & \delta_{st}(v), & \forall v \in V \\
& \text{(iii)} & & \sum_{a \in A} w_{ar} x_a & \leq & U_r, & \forall r \in \mathcal{R} \\
& \text{(iv)} & & x_a & \in & \{0, 1\}, & \forall a \in A
\end{array}$$

Here, $\delta_{st}(v) = 1$ if $v = s$, $\delta_{st}(v) = -1$ if $v = t$ and $\delta_{st}(v) = 0$ otherwise. We solve this problem using a branch-and-bound algorithm similar to Beasley & Christofides (1989) [22], using lower bounds derived from a Lagrangean relaxation of the resource constraints (RCSP) (iii), see Borndörfer, Grötschel & Löbel (2003) [32] for more details on the dynamic program. In addition, we used "configurable" classes of classical linear resource constraints and cumulative resource constraints with replenishment arcs. We can handle most pairing construction rules directly by multi-label methods. Irnich & Desaulniers (2005) [120] and Irnich et al. (2010) [121] gives a recent survey on resource constrained shortest path problem and how to tackle them in a column generation framework. Some rules, however, are so complex that these techniques would become unwieldy or require too much customization. For such cases we used a callback mechanism, that is, we ignore the rule in our pricing model, construct a pairing, and send it to a general rule verification oracle that either accepts or rejects the pairing. This can be seen as adding additional resource constraints for infeasible paths in an dynamic cutting plane manner. Let $|P|$ be length of p and $\mathcal{P}$ a set of forbidden paths, then

$$\text{(iii-b)} \quad \sum_{a \in p} x_a \leq |P| - 1, \forall p \in \mathcal{P}.$$

ensures feasibility of the paths, so that a one to one correspondence to pairings is reached. Even if this allows for a general application, we want to point out explicitly that such rules slow down the pricing routine. Therefore, we recommend to avoid such unstructured rules if possible.

8.6 Branch and Price

The optimal solution value of (MLP) is a global bound on the optimal value of the model (SPP). If we unfortunately get a fractional solution variable x_p, we must branch and apply a divide and conquer technique to ensure integrality. This is the state of art and standard technique to solve mixed integer programs (MIPs), see once again Achterberg (2007) [3]. In addition to the standard preprocessing techniques, branching rules, node selections, heuristics, and cutting plane procedures, we have to resolve the LP-relaxation of the subproblems induced by the branching or in other words fixing decisions. In contrast to standard or static MIP solving we have to keep in mind that in our new branches some non-generated variables are possibly required to solve these subproblems to optimality. In addition, we have to ensure that the branching decisions so far are respected. Hence, we have to enrich the standard pricing of variables with a dynamic procedure that respects the fixing decisions as well, i.e., the branch on $x_p = 0$.

Added together this leads to an exact approach, so called branch and price algorithm, to solve large scale MIPs to optimality. For practical instances this may be too time consuming and even not appropriate because getting a feasible good solution in acceptable time is more important in practice than proving optimality. Solving the restricted variant of the (SPP) via branch and bound only will lead to poor solutions. Therefore, pricing is required in some branch and bound nodes to "complete" the solution and to generate "undesirable" pairings, i.e., from a cost or dual perspective, in the end. This real-world requirement can be achieved by powerful problem adaptive heuristics, which only perform pricing in several promising nodes of the branch and bound tree. Hence, a global guaranteed bound and optimality gap can still persist.

8.7 Crew Composition

A main difference to duty scheduling in public transport or railway transport is that for airline crew scheduling complete crews must be considered, i.e., each leg has to be covered by at least two pairings. However, the rules and costs are quite different due to varying contracts and responsibilities, i.e., cockpit crews are paid higher than the cabin crews. Furthermore the number of required members of the cabin crew can differ from flight to flight. This could lead to noteworthy savings but also to inhomogeneous pairings. Of course an aviation company wants to have homogeneous pairings to increase the stability of the schedule. In case of unavoidable disturbances and cancellations a schedule with constant crew compositions seems to be more stable and recoverable, because only this crew is affected from disturbances.

To handle this “regularity” requirement, we did some preliminary computational experiments for an straight forward sequential approach by using the introduced standard model (SPP), see Borndörfer et al. (2005) [33]. In a first step the major cost component which is the cockpit crew is minimized. After this, these pairings were set as “desired ones”, if they are still valid for the other crew part, or at least new ones are preferred to be as similar as possible to the fixed one of the cockpit. In a second step we re-optimize the cabin pairings using model (SPP) with respect to the adapted cost function and cabin rules. This sequential approach produces homogeneous solutions for cockpit and cabin crew very fast. Potthoff, Huisman & Desaulniers (2008) [177] successfully used similar ideas and models for re-scheduling of crews at the operational stage. From our point of view, an integrated model for cabin and cockpit crew is only required if the cost structure changes significantly.

Chapter II

Railway Modeling

In this chapter we describe techniques to model railway systems with different granularities of the underlying railway infrastructure. In a so-called microscopic representation of the railway system almost all technical details are considered. The analysis of very detailed models can lead to more reliable conclusions about the railway system. Therefore microscopic models are basically used to evaluate timetables via railway simulation systems, i.e., to respect the safety system exactly. The disadvantage of very detailed models is the vast amount of data that needs to be acquired and processed. Even more computational capabilities and data management reach their limits.

M. Soukup wrote in a Swiss newspaper article, in the Sonntagszeitung from 24.08.2008, about the new planning system `NeTS`:

> "Since 21. July 2008 the first 50 SBB schedulers have been developing the timetable for 2010 using the new system. By the date of the changeover to the new timetable on 12. December 2009, 500 more people will be working with `NeTS`. Huge amounts of information are currently being entered into the system. For example, when the IC828 train leaves Zurich at 3pm heading for Bern, the timetable schedulers must first take into account around 200 parameters, including the time of day, the rolling stock, the type of train, the length of the train, the length of the route and conflicts when entering and leaving stations. Extrapolated up to cover the whole timetable, this means that `NeTS` processes around 3.6 billion pieces of information and needs between 500 and 700 gigabytes of storage space."

To approach this problem, macroscopic models are developed that simplify and aggregate the railway infrastructure representation. Main

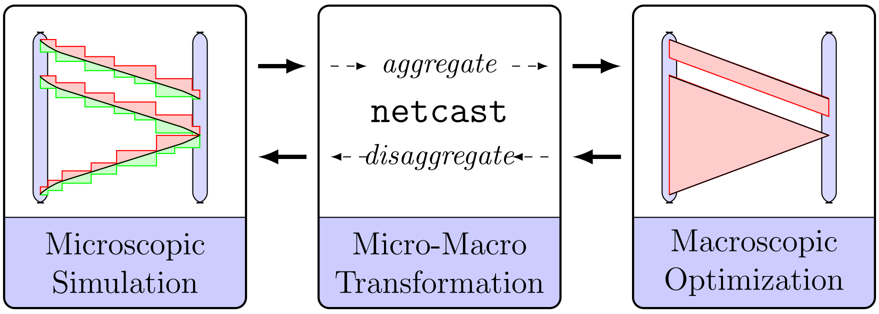


Figure 1: Idealized closed loop between railway models of different scale for railway track allocation.

application of macroscopic models are timetable information systems. One goal of this work is to extend the usage of macroscopic models to capacity allocation. Therefore we define microscopic railway infrastructure resources and their macroscopic counterparts. The challenge is to specify a reduced and manageable model which sustains the core of the system at the same time. A classification and comparative discussion of railway infrastructure models can be found in Radtke (2008) [180].

The major contribution of this chapter will be the development a bottom-up approach to construct a macroscopic model which conserves resource and capacity aspects of the considered microscopic railway system, i.e., resulting in the tool NETCAST. Such formalized and aggregated models can be tackled by optimization methods, especially integer programming. The main concept of this Micro-Macro Transformation is shown in Figure 1.

This will be the topic of the next chapter. A highlight will be the evaluation of the proposed network simplification and an aggregation method on real world data as presented in Borndörfer et al. (2010) [42]. Furthermore, we establish the theoretical background in Schlechte et al. (2011) [190] to quantify the quality of the resulting macroscopic model. The essential task is here to analyze the information loss and to control the error caused by the Micro-Macro Transformation.

Most that will be presented in this chapter is joint work with Ralf Borndörfer, Berkan Erol and Elmar Swarat. It is based on several discussions with researchers from institutes on railway transport, railway operations and operations research, as well as railway experts from different railway undertakings and infrastructure providers.

Let us name some of them here: Sören Schultz, Christian Weise, Thomas Graffagnino, Andreas Gille, Marc Klemenz, Sebastian Klabes, Richard Lusby, Gabrio Caimi, Frank Fischer, Martin Fuchsberger, and Holger Flier. In particular we want to thank: Thomas Graffagnino from SBB (Schweizerische Bundesbahnen) who provided us real world data and explained us a lot of technical issues, Martin Balser who points out and contributed to the rounding and discretization aspects, and Daniel Hürlimann and his excellent support to the simulation tool `OpenTrack`.

To establish an optimization process to the allocation of "railway capacity", we first have to define capacity and derive a resource based model for a railway system in an appropriate way. Railway capacity has basically two dimensions, a space dimension which are the physical infrastructure elements as well as a time dimension that refers to the train movements, i.e., occupation or blocking times, on the physical infrastructure.

A major challenge of both dimensions is the granularity, the potential size, and the arbitrary smooth variation of time. Figure 2 shows the the rather small German station Altenbeken in full microscopic detail, i.e., with all segments, signals, switches, crossovers, etc.

Railway efficiency and the capacity of railway networks are important research topics in engineering, operations research, and mathematics for several decades. The main challenge is to master the trade-off between accuracy and complexity in the planning, optimization, and simulation models. Radtke (2008) [180] and Gille, Klemenz & Siefer (2010) [100] proposed the use of both *microscopic* and *macroscopic models*. They applied microscopic models for running time calculations and the accurate simulation of railway operations, and macroscopic models for long term traffic and strategic infrastructure planning. In a similar vein, Schultze (1985) [195] suggested a procedure to insert train slots according to predefined priorities in a first step, and to test the reliability of this timetable in a second step by simulating stochastic disturbances. An alternative approach to determine the capacity of a network are *analytical methods*. They aim at expressing the railway efficiency by appropriate statistics, e.g., the occupancy rate. There exist two different approaches: The first is the *handicap theory* by Potthoff (1980) [178]; it is based on queuing models. The second uses *probabilistic models* to compute follow-on delays; it is mainly based on the work of Schwanhäußer (1974) [196]. He also introduced the important concept of *section route nodes* to analyze the performance of route nodes or stations. Hansen (2010) [109] presents a probabilistic model as an alternative to queuing models for a precise estimation of expected buffer and running times.

The chapter is organized as follows. In Section 1 we will recapitulate and describe microscopic aspects of the railway system, to establish a definition of resources and capacity, see Landex et al. (2008) [145]. In the literature several approaches work directly on a microscopic level with the disadvantage that only instances of small size can be handled, see Delorme, Gandibleux & Rodriguez (2009) [74]; Fuchsberger (2007) [94]; Klabes (2010) [129]; Lusby et al. (2009) [159]; Zwaneveld et al. (1996) [220]; Zwaneveld, Kroon & van Hoesel (2001) [221].

Nevertheless, on a planning stage it is not possible to consider all these details and also not necessary. Hence, the main goal for a macroscopic model is to evaluate different timetable concepts or infrastructure decisions on a coarse granularity. Only recently approaches were developed to tackle larger corridor or even network instances. In Caimi (2009) [57] a top-down approach is presented and used to handle the complete Swiss network by a priori decomposition of the network into different zones. In contrast to that, we present a bottom-up approach to define a macroscopic railway model in Section 2. The introduced transformation from the microscopic to macroscopic view is described in detail, analyzed with respect to the discretization error, implemented as a tool called `NETCAST`, and successfully evaluated on real world scenarios, e.g. the Simplon corridor see Erol (2009) [84]. On the one hand these models are precise enough to allow for valid allocations with respect to blocking times, on the other hand they are simplified and aggregated to a coarse level, which allows for solving large scale optimization instances.

1 Microscopic Railway Modeling

Railway traffic is a high-grade complex technical system, which can be modeled in every detail. This is necessary to ensure that each microscopic infrastructure element, i.e., block segment, is occupied by at most one train at the same time. State of the art simulation systems provide accurate estimations of *running times* with respect to such a precise microscopic model. The time period, when a train is physically using a block section, is called running time. Microscopic data is for

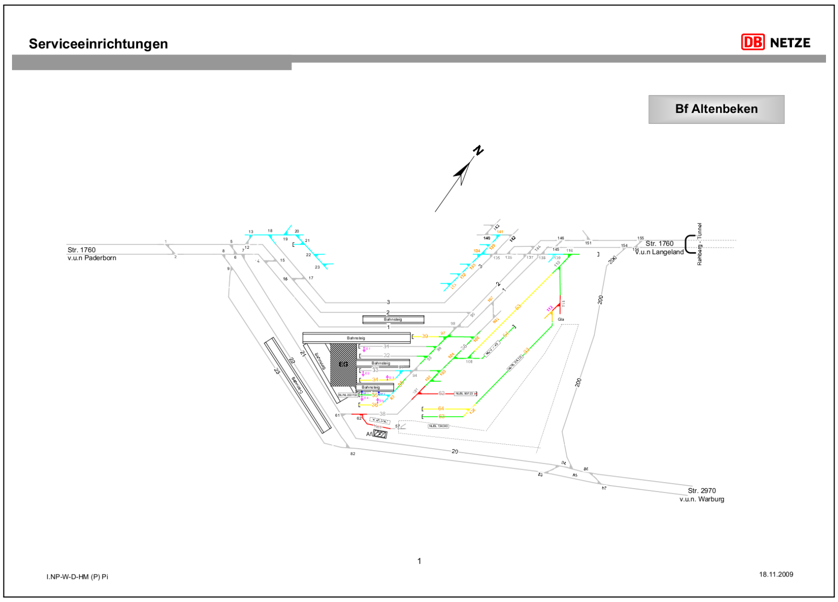


Figure 2: Detailed view of station Altenbeken provided by DB Netz AG, see Altenbeken [11]

example incline, acceleration, driving power, power transmission, speed limitations, signal positions.

In this section, we define all needed microscopic elements and data as well as all macroscopic objects. This work was done in a close collaboration with the SBB, who provided data for the scenario of the Simplon corridor, see Borndörfer et al. (2010) [42]. In Figure 5 the microscopic infrastructure of the Simplon area based on the simulation tool `OpenTrack`, see OpenTrack [172], is shown. The microscopic network consists of 1154 nodes and 1831 edges.

The input for NETCAST is the *microscopic infrastructure network* that is modeled by a graph $G = (V, E)$. `OpenTrack` uses a special graph structure where the nodes are so called *double-vertices*, that consist of a left and a right part. A convention in `OpenTrack` is that if a path in G enters a node at the left end it has to leave at the right or vice versa. This ensures that the direction of the train route is always respected and no illegal turn around at switches is done on the way. Figure 3 shows an example of a double-vertex graph from `OpenTrack`. Montigel (1994) [163] proposed this concept to describe microscopic railway networks. Figure 4 shows a straightforward transformation of a double vertex graph to general directed graph.

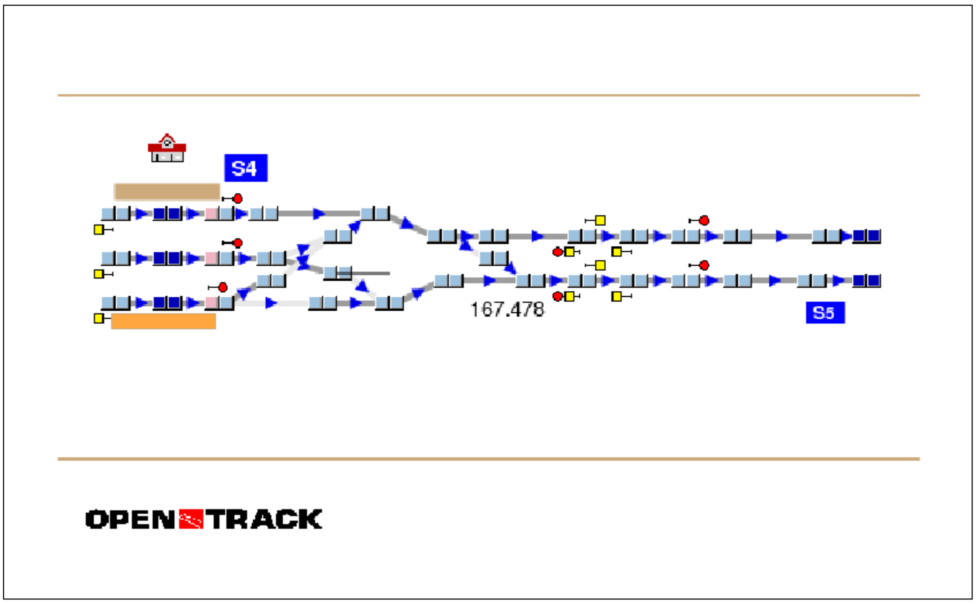


Figure 3: Screenshot of the railway topology of a microscopic network in the railway simulator `OpenTrack`. Signals can be seen at some nodes, as well as platforms or station labels.

Every railway edge $e \in E$ has some attributes like maximum speed or incline. A node $v \in V$ is always defined, if one or more attributes change or if there is a switch, a station, or a signal on this track. Every track section between two nodes is modeled as an edge.

Our transformation approach is based on a potential set of *routes* in G for standard trains, so called *train types*. The set of train types is denoted by C. Let R be the set of all given routes in G. In addition we are given a mapping $\theta : R \mapsto C$ for all routes to the rather small set of standard train types. It is for example possible to have microscopic routes to ICE trains, which differs in their weight or length due to the composition, and to aggregate them in one standard train type for ICEs.

Figure 4: Idea of the transformation of a double vertex graph to a standard digraph

A microscopic route is a valid path through the microscopic infrastructure which starts and ends at a node inside a station or at a node representing a parking track. In addition, it is possible that other nodes on the route are also labeled as stops where the train could potentially wait.

Furthermore, these train routes induce in which direction the microscopic infrastructure nodes and edges can be used. This will directly influence the definition, i.e., the headway parameter, of a macroscopic model, as we will explain later in Section 2. They ought to be reasonable and conservatively grouped with respect to their train class (heaviest cargo trains, slowest interregional or regional passenger trains). Thus only a minimal difference of the running times within a train type occurs and each associated train route can realize these times by slowing down if necessary. For these standard train routes detailed simulation data has to be evaluated carefully such that reliable running and blocking times in units of δ, i.e., times provided by the micro simulation, are given in seconds, see Figure 6. Note that several routes of R belong to the same train type. For example in case of a heavy cargo train that is allowed to stop at some intermediate station, i.e., at one microscopic node, S we simulate two routes; the first without and the second with stopping at S. Hence we have different running times and blocking times with respect to the behavior of the train at the start or end station, i.e., we will use later the term running mode for this. Obviously, trains which have to break or accelerate have larger running times and hence resource consumptions.

Example 2.2 shows the significant differences between the durations, i.e., the running and blocking times related to S. Therefore our macroscopic approach has to cope with that by considering not only train type but also event dependencies.

In Pachl (2002) [173] and Brünger & Dahlhaus (2008) [46] the laws of basic dynamics are applied to describe the *dynamics of a train movement*. Basically, three groups of forces are considered, tractive, inertia and resistance force. If all needed parameters are given, e.g., mass, acceleration and deceleration of the train, (directed) incline of the block section, running times of train movements can be estimated very accurately. In state of the art railway simulation software, e.g., `OpenTrack`, all relevant parameters are considered in order to provide plausible values, see Nash & Huerlimann (2004) [166].

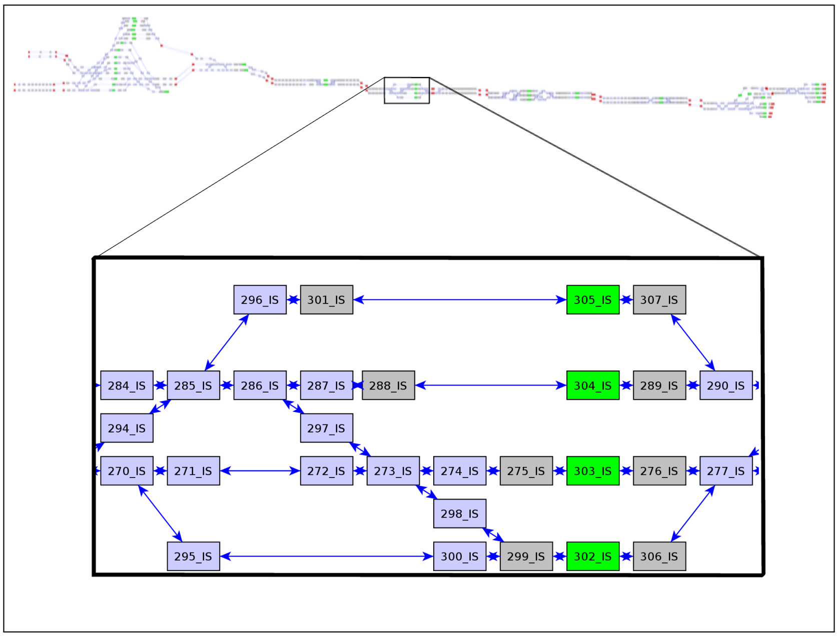


Figure 5: Microscopic network of the Simplon and detailed representation of station Iselle as given by `OpenTrack`

In Europe, *blocking times* are used to quantify the infrastructure capacity consumption of train movements. The approach is based on the early works of Happel (1950) [110] and Happel (1959) [111] and the intuitive concept to associate the use of physical infrastructure resources over certain time intervals with trains or train movements, see also Klabes (2010) [129]; Pachl (2008) [174] for a comprehensive description of blocking time theory. We will now give a brief discussion of blocking times that contributes to a better understanding of our transformation algorithm.

The origin of the *blocking time stairs*, shown in Figure 6, is the well-known train protection system, called *train separation in a fixed block distance*. Nowadays these are train control systems that indicate the moving authority to the train drivers and thus ensures safe railway operation. In this method, the railway network is divided into *block sections*, which are bordered by *main signals*. A block section must not be occupied by more than one train at a time. When a signal allows a train to enter a block section, the section is locked for all other trains. In this way, the entire route between the block starting main signal and the overlap after the subsequent main signal has to be reserved for the entering train.

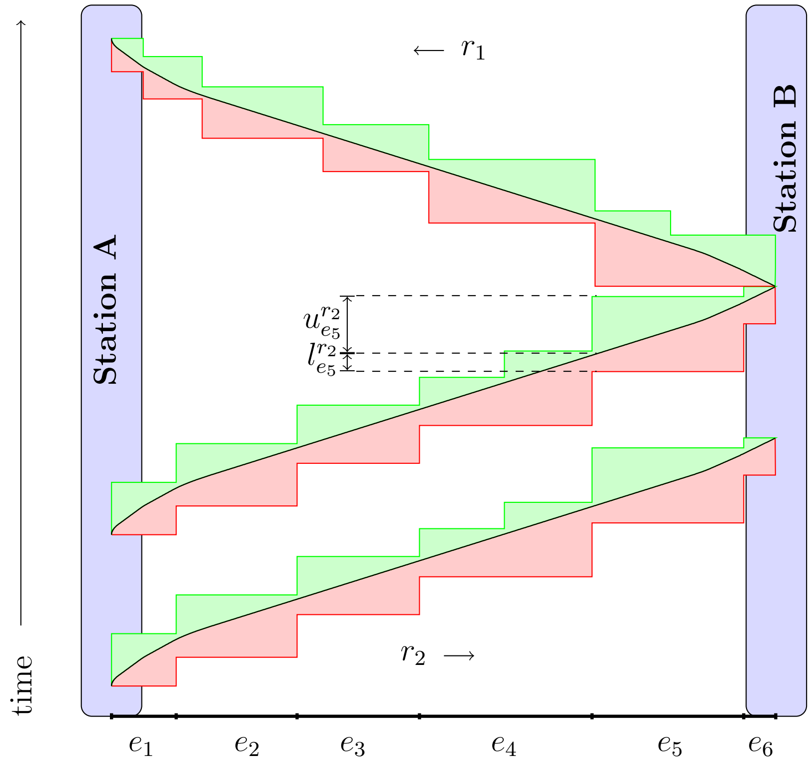


Figure 6: Blocking time diagrams for three trains on two routes using 6 blocks. In the lower part of the diagram two subsequent trains on route r_2 and at the top one train on the opposite directed route r_1 are shown.

Figure 6 shows that the time interval during which a route r occupies a track segment consists of the relative reservation duration l^r_e and the relative release duration u^r_e on edge $e \in E$. The relative reservation duration is the sum of the approach time, the signal watching time, sometimes called reacting time, and time needed to set up the route. The relative release duration is the sum of the release time, the clearing time, sometimes called switching time, and time needed by the train between the block signal at the beginning of the route and the overlap. The switching time depends significantly on the installed technology, see Klabes (2010) [129]; Schwanhäußer et al. (1992) [197]. In order to prevent trains that want to pass a block section from undesiderable stops or brakings, the block reservation should be finished before the engine driver can see the corresponding distant signal. Then the section stays locked while the train passes the track between the beginning of the visual distance to the caution signal and the main signal and thereafter the block section until it has cleared the overlap after the next main signal. Then the section is released. This regime can be improved in block sections that contain con- or diverging tracks, because in such cases it is often possible to release parts of the section before the train has passed the overlap after the next main signal.

We only want to mention that our approach can be easily adapted to other simulation tools that provide accurate running and blocking times, like `RailSys` or `RUT-K`. We remark that these tools differ in their definition of objects, interfaces and some minor interpretations and that although our exposition is based on the simulation tool `OpenTrack`, the main concepts of running and blocking times are the same and thus the methodology is generic.

We summarize the microscopic information that we use:

- ▷ an (undirected) infrastructure graph $G = (V, E)$,
- ▷ a set of directed train routes R, $r = \{e_1, e_2, \ldots, e_{n_r}\}$ with $e_i \in E$,
- ▷ a set of train types C,
- ▷ a mapping θ from routes R to train types C,
- ▷ positive running time $\tilde{d}_e^r$ on edges $e \in E$ for all routes $r \in R$ measured in δ,
- ▷ positive release duration u_e^r on edges $e \in E$ for all routes $r \in R$ measured in δ,
- ▷ positive reservation duration l_e^r on edges $e \in E$ for all routes $r \in R$ measured in δ,
- ▷ orientation of edges is induced by traversing routes (one or both directions),
- ▷ stop possibilities for some nodes $v_i \in V$ are induced by traversing routes.

Remark 1.1. *Though we develop our transformation approach for fixed block railway operation systems, the methodology and models could be easily applied to moving block systems. Future systems like ETCS Level 3 can already be modeled in simulation tools. Arbitrarily small blocks, i.e., blocks with lengths converging to zero, are considered in simulations to emulate the resulting blocking times, see also Emery (2008) [82] and Wendler (2009) [214] for an investigation of the influence of ETCS Level 3 on the headway times. Simulation tools have to respect all these technical details. From an optimization point of view, however, it is sufficient to consider abstract blocking time stairs, regardless from which safety system they result or how they were computed.*

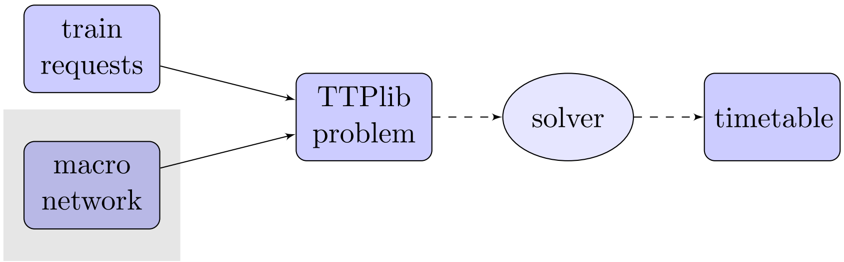


Figure 7: I/O Concept of TTPlib 2008 (focus on macroscopic railway model)

2 Macroscopic Railway Modeling

In this section we present a formal macroscopic railway model. The establishment of standard models and standard problem libraries have contributed to the success in problem solving. Such libraries exist for the famous Traveling Salesman Problem, see Reinelt (1991) [181], as well as for general Mixed Integer Programs, see Achterberg, Koch & Martin (2006) [4].

We invented a standardization of a *macroscopic railway model* and introduced the library TTPlib for the track allocation or timetabling problem, see Erol et al. (2008) [85]. Figure 7 illustrates the data handling of a train timetabling problem. Section 2.1 motivates the aggregation idea and recapitulates the standardization of the resulting macroscopic infrastructure model. Section 2.2 discusses the discretization problem when transferring microscopic models to macroscopic ones. Finally, we introduce an algorithm that performs the Micro-Macro-Transformation in Section 2.3. Furthermore, we will show that the constructed macroscopic model is reliable such that the results can be re-transformed and interpreted in a microscopic model and finally operated in "reality". The introduced algorithm constructs from a microscopic railway model a macroscopic model with the following properties:

- ▷ macroscopic running times can be realized in microscopic simulation,
- ▷ sticking to macroscopic headway-times leads to conflict-free microscopic block occupations,
- ▷ valid macroscopic allocations can be transformed into valid microscopic timetables.

2.1 Macroscopic Formalization

The desired *macroscopic network* is a directed graph $N = (S, J)$ for train types C deduced from a microscopic network $G = (V, E)$ and train routes R. On this level, our goal is to aggregate (inseparable) block sections (paths in G) to *tracks* J and station areas (subgraphs of G) to *stations* S.

The aggregation will be done in a way that depends on the given routes R and the simplification to train types C imposed by the mapping θ, such that the complexity of the macroscopic network depends only on the complexity of the interactions between the *given* train routes, and not on the complexity of the network topology, which covers all interactions between *all* potential train routes, which is much more. This is a major advantage over other approaches, because the aggregation is detailed where precision is needed and compressed where it is possible.

We will now describe the idea of the construction by means of an example. First, all potential departure and arrival nodes at some station that are used by the routes R are mapped to one macroscopic station node. Additional macroscopic nodes will be introduced in order to model interactions between routes due to shared resources. The potential interactions between train routes in a double-vertex graph are:

- ▷ complete coincidence, i.e., routes have an identical microscopic path,
- ▷ convergence, i.e., routes cross at a microscopic node (and traverse it in the same direction),
- ▷ divergence, i.e., routes separate at a microscopic node (and traverse until then in the same direction),
- ▷ or, crossing, i.e., routes cross at a microscopic node (and traverse it in the opposite direction).

Note that two routes can correlate in various and numerous ways. Let us discuss some of these interactions between train routes at the example of the infrastructure network shown in Figure 8.

Consider first a single standard train that runs from platform A (We denote any place where stopping is allowed as a platform.) to platform X. Then it is enough to consider just one single track from station A to X in the macroscopic infrastructure. Note that this macroscopic track could correspond to a long path in the microscopic representa-

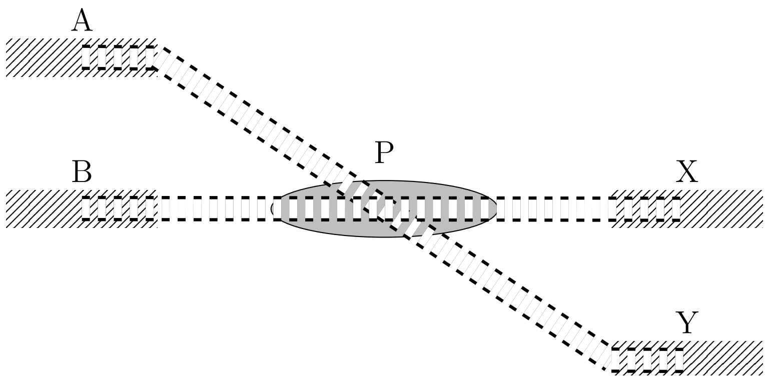


Figure 8: Example of macroscopic railway infrastructure

tion. Consider now additional standard trains from A to X. Possible interactions and conflicts between these train routes are the self correlation on the *directed track* from A to X, as well as the platform capacity for standard trains, which allows, say, exactly one train to wait in A or X. Another standard train running from B to X calls for the definition of a *pseudo-station* P at the track junction in order to model the train route convergences correctly. (Our model distinguishes between regular station nodes, where a train can stop, and pseudo-station nodes, which are not stop opportunities, i.e., in our model trains are not allowed to wait at a pseudo-station or to change their direction there.) The pseudo-station P splits the track from A and X into two tracks: from A to P and from P to X. The second of these tracks is used to model the resource conflict between converging routes of trains from A to X and trains from B to X, which is locally restricted to the track from P to X (or more precisely from the first blocks to reserve containing the switch of P). If it is possible to run trains on the same microscopic segment in the opposite direction from X to A, another directed track has to be defined in the macroscopic network. Besides the standard self correlation, the conflict for opposing routes also has to be modeled, see Figure 6. Diverging or crossing situations between opposing train routes can be handled in an analogous way. Along the lines of these examples, we can exploit aggregation potentials in the infrastructure by representing several microscopic edges on a route by only one macroscopic track. Of course, macroscopic track attributes can also be compressed. For example, if we assume that the route from A to X and the route from B to X are operated by the same train type, we can use a single value for the running time on the track from P to X.

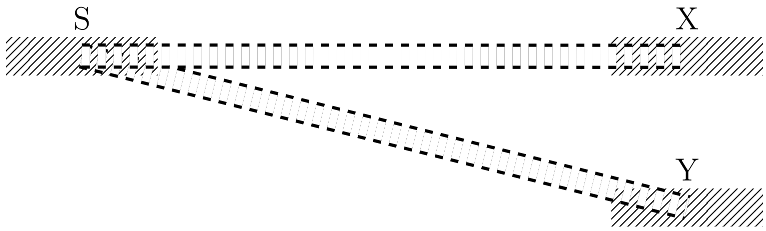


Figure 9: Example of aggregated infrastructure

After constructing the regular stations, the pseudo-stations, and the tracks between them, the network can be further reduced by a second aggregation step. Again consider the situation in Figure 8. Suppose platforms A and B belong to the same station S. If P is a close junction associated with S, then it may be viable to contract nodes A and B to one major station node S with a directed platform capacity of two as shown in Figure 9. Of course, by doing so we loose the accuracy of potentially different running times between different platforms of S and the other stations, and we also loose control over the routing through or inside S, which both can produce small infeasibilities on the operational level. However, one can often achieve significant reductions in network sizes in this way, without loosing too much accuracy.

This is exactly a decomposition of the TTP for the microscopic network to a TTP for a macroscopic network with aggregated stations and several TPP for the microscopic station areas. The next paragraphs will describe the macroscopic elements and attributes in more detail.

2.1.1 Train Types and Train Type Sets

As a first component, the macroscopic model groups trains with similar properties to a set of train types C, as mentioned above. The train sets, i.e., and so the train types, are structured hierarchically by a tree. In this tree each node corresponds to exactly one train set $f \in F \subseteq \mathcal{P}(C)$, which consists of all leaf nodes. The leaf nodes represent train sets consisting of exactly one train type $c \in C$. For each train set all properties, e.g., running or turn around times, of the parent train set are valid; analogously, restrictions, e.g., station capacities, of all parent train sets have to be fulfilled, as well as the train set specific ones.

Figure 10 shows an example tree. If a running time for train set 1 on track $j \in J$ is defined, then this time is also valid for 4. If a station capacity at station $s \in S$ is defined for all trains of set 2, then trains of

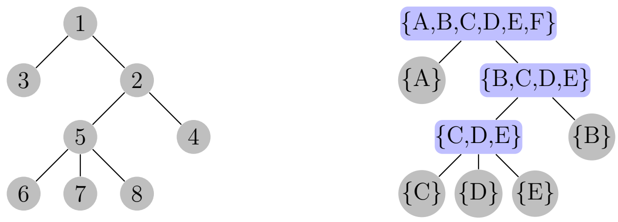


Figure 10: Train types and train sets defined as a poset

set 4 to 8 are also captured by the capacity rule. On the right side of Figure 10 the nodes of the tree are interpreted as sets of train types.

In a mathematical interpretation, these trees are Hasse diagrams visualizing a partially ordered set, see Birkhoff (1967) [26]. That is a binary relation of the finite set C, which is reflexive, antisymmetric, and transitive. In our setting the set F is ordered by inclusion and the minimal elements of this poset are the elements of the set of train types C.

2.1.2 Stations

The nodes S of the digraph $N = (S, J)$ are called stations. We distinguish three types of them:

- ▷ standard-stations (two-sided, labeled with 1 and 2), where it is possible for a train to pass through, turn around, or wait,
- ▷ dead-end stations (one-sided, labeled with 1), where no passing is possible,
- ▷ and, pseudo-stations (two-sided, labeled with 1 and 2), where no turn around or waiting is possible.

Even if in daily operation trains could stop and wait at pseudo-stations, i.e., if a red signal of the security system is shown in front of this junction, on a planning level stopping there is strictly forbidden due to the assumed green wave policy.

We restrict ourself to standard cases of station capacities, such as maximal number of trains of a certain train set at one time step at a station. More precisely, we use different running modes of trains, which will be introduced in the next section. Therefore we can further restrict the number of trains that are stopping in or passing through a station. Station capacity constraints can be many other requirements as well, such as:

- ▷ maximum capacity per side of station,
- ▷ maximum capacity of station per time interval,
- ▷ maximum capacity of station at a specific time interval,
- ▷ forbidden combinations of (running) modes per train set,
- ▷ forbidden combinations of modes per combinations of train set,
- ▷ or, forbidden meetings in stations.

The extension of the model is straightforward for these numerous imaginable special cases and can be easily achieved as we will see later. Finally, we list all attributes of station nodes:

- ▷ name and coordinates,
- ▷ type (standard, dead-end, pseudo) and number of sides,
- ▷ turnaround times $d_{s,f}$ for each $s \in S$ and $f \in F$,
- ▷ station (event) capacities $\kappa_{s,f}$ for each $s \in S$ and $f \in F$.

2.1.3 Tracks

The set of arcs J of $N = (S, J)$, denoted as tracks, correspond to several block sections of the railroad network. For a standard double-way track between station $x \in S$ and $y \in S$, more precisely between two sides of them, there exist two opposite directed arcs $(x, y) \in J$ and $(y, x) \in J$. Physical track segments, which can be used in both directions, corresponds to two opposite directed arcs of J and build a *single way track*. By definition it is not possible to overtake on a track. This is only possible inside stations by using different tracks, i.e., the station capacity must allow this. More precisely the order of entering trains on each track can not change at the arrival station. This assumption has an effect on the definition of the network segmentation as well as on the *minimal departure headway times*, see Definition 2.8.

Block section exclusivity on a microscopic stage, which we described in Section 1, transfers to minimal headway times at departure. The minimal abiding difference of the departure times between two consecutive trains is defined as the minimal departure headway time to ensure safety on each track $j \in J$.

Remark 2.1. *Note that it is possible to have more than one track between station $x \in S$ and $y \in S$. Therefore $N = (S, J)$ is a multi-graph (allowing parallel arcs) and we should use consistently the notation $a \in J$, instead of $(x, y) \in J$. However, in cases were we use $(x, y) \in J$ we indirectly assume that (x, y) is unique. Furthermore, all single way*

preceding		succeeding		minimum headway time	
train type	running mode	train type	running mode	simulation result in seconds	rounded value in minutes
GV	stop-stop	ICE	stop-stop	475	8
GV	stop-stop	ICE	stop-pass	**487**	9
GV	stop-stop	ICE	pass-stop	466	8
GV	stop-stop	ICE	pass-pass	477	8
GV	stop-pass	ICE	stop-stop	469	8
GV	stop-pass	ICE	stop-pass	474	8
GV	stop-pass	ICE	pass-stop	460	8
GV	stop-pass	ICE	pass-pass	464	8
GV	pass-stop	ICE	stop-stop	321	6
GV	pass-stop	ICE	stop-pass	333	6
GV	pass-stop	ICE	pass-stop	312	6
GV	pass-stop	ICE	pass-pass	323	6
GV	pass-pass	ICE	stop-stop	315	6
GV	pass-pass	ICE	stop-pass	320	6
GV	pass-pass	ICE	pass-stop	**306**	6
GV	pass-pass	ICE	pass-pass	310	6

Table 1: Technical minimum headway times with respect to running mode

tracks are specified as disjunctive pairs of J, so we use $\overline{j} \in J$ to denote the counterpart or complement of track $j \in J$.

As we have already motivated in Section 1, the running dynamics are relevant for the traversal time on a track and the corresponding headway times.

Example 2.2. *We want to clarify that on real numbers from the scenario* HAKAFU_SIMPLE. *The simple simulation via* ***`RailSys`*** *of the ordered pair of a cargo train (GV) and a fast intercity train (ICE) on track FOBR to HEBG produces 16 different headway times in seconds. Table 1 lists these numbers as well as the rounded values in minutes. It can be observed that depending on the running mode of the trains the headway time can differ more than 3 minutes, i.e., the worst case value reserves 50 % more capacity than the best case. Thus, a simple worst case assumption could lead to an underestimation of the potential capacity.*

By this observation it is necessary to distinguish at least between stopping and passing trains. Otherwise one could not guarantee feasibility if we would be to optimistic in chosing the headway time or contrary a too conservative value would lead to underestimation of the real track capacity. Let $M_S = \{dep(arture), arr(ival), pass\}$ be the set of possible events or modes at the stations. Furthermore, we consider the following standard running modes $M_J \subseteq M_S \times M_S$ for train runs on a track:

▹ stops at departure node and arrival node (1),
▹ stops at departure node and passes at arrival node (2),
▹ passes at departure node and stops at arrival node (3),
▹ and, passes at departure node and arrival node (4).

Minimum headway times can be defined for all modes individually, which is reasonable, see again Example 2.2. Furthermore the handling of the events inside a station can be seen in Example 2.5. Figure 13 shows the interpretation of turn around activities inside a station as dashed arcs. In pseudo stations only directed passing and in dead-end stations only arrival and departure events have to be considered. By definition, passing nodes of side 1 represent trains entering at side 1 and leaving at side 2, passing nodes of side 2 represent trains entering at side 2 and leaving at side 1.

A detailed definition and way of calculation of these times with respect to the microscopic model is topic of Section 2.3. After listing all attributes of a track $j \in J$, we will present some small examples:

▹ start station ($tail \in S$) and side ($\in \{1, 2\}$),
▹ end station ($head \in S$) and side ($\in \{1, 2\}$),
▹ type, i.e., single way track or standard,
▹ running times $d_{j,c,m} \in \mathbb{N} \setminus \{0\}$ depending on train type $c \in C$ and mode $m \in M_J$,
▹ minimum headway times $h_{j,c_1,m_1,c_2,m_2} \in \mathbb{N} \setminus \{0\}$ for departing train pairs, i.e., $c_1, c_2 \in C, m_1, m_2 \in M_J$,
▹ minimum headway times for departing train on j and a departing train on the complement track $\overline{j}$, if single way track (sets and mode).

Example 2.3. *In Figure 11 a macroscopic railway network is shown with only two standard tracks connecting standard station A via pseudo station P with dead-end station B. Running times of mode (1) are illustrated as solid lines and the corresponding minimum headway times are shown as dotted lines for two different train types. The corresponding running time values and headway matrices are:*

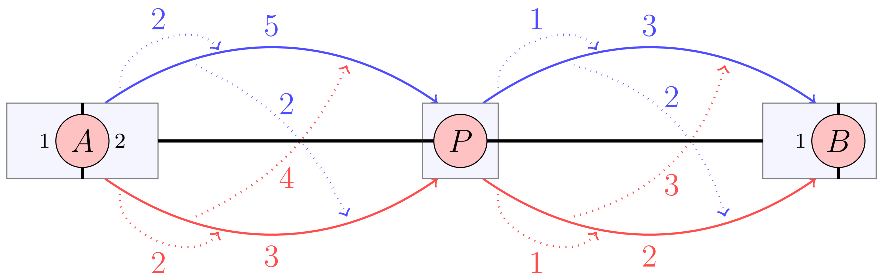


Figure 11: Macroscopic modeling of running and headways times on tracks

$$d_{A,P} = \begin{pmatrix} 5 \\ 3 \end{pmatrix}, \; H_{A,P} = \begin{pmatrix} 2 & 2 \\ 4 & 2 \end{pmatrix}, \; d_{P,B} = \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \; H_{P,B} = \begin{pmatrix} 1 & 2 \\ 3 & 1 \end{pmatrix}.$$

Example 2.4. *A more complex situation is modeled in Figure 12. We have a single way track between $P1$ and $P2$, which can be used in both directions. On the one hand, blue trains are running from A to C traversing $P1$ and $P2$. On the other hand, red trains from D run via $P2$ and $P1$ to station B. In this scenario, the two track arcs corresponding to the segment between nodes $P1$ and $P2$ are directed opposite and build a single way pair. Only one train can pass this section at a time and therefore headway times for the combination of a train from $P1$ to $P2$ and a train from $P2$ to $P1$ and vice versa are additionally needed:*

$$d_{AP,1} = \begin{pmatrix} 5 \end{pmatrix}, \; H_{A,P1} = \begin{pmatrix} 2 \end{pmatrix}, \; d_{P1,P2} = \begin{pmatrix} 3 \end{pmatrix}, \; H_{P1,P2} = \begin{pmatrix} 2 \end{pmatrix},$$

$$d_{P2,C} = \begin{pmatrix} 3 \end{pmatrix}, \; H_{P2,C} = \begin{pmatrix} 2 \end{pmatrix}, \; d_{D,P2} = \begin{pmatrix} 3 \end{pmatrix}, \; H_{D,P2} = \begin{pmatrix} 1 \end{pmatrix},$$

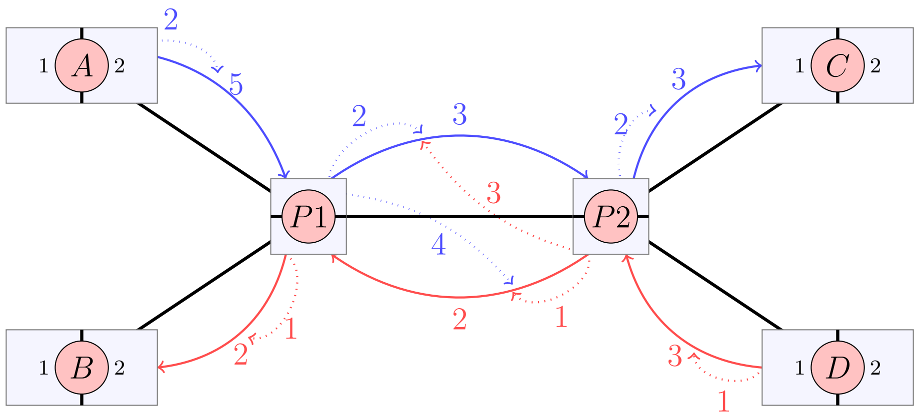


Figure 12: Macroscopic modeling of a single way track

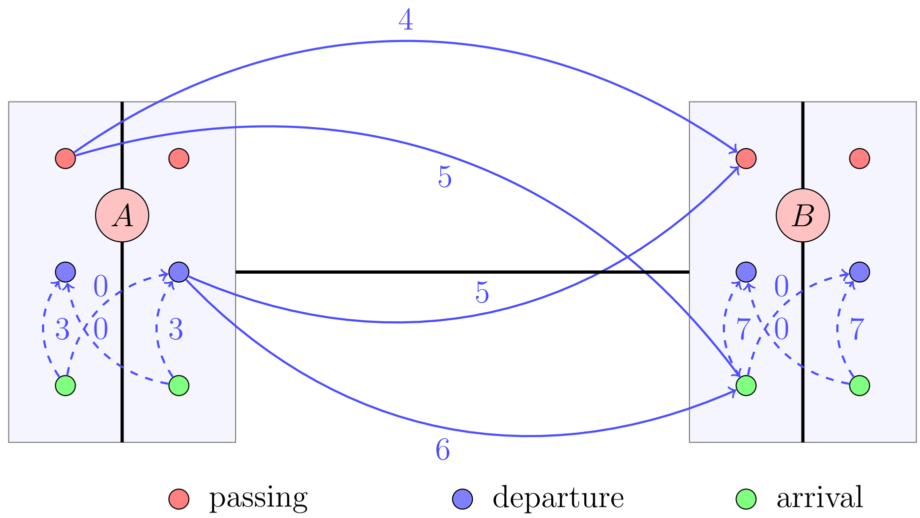


Figure 13: Representation as event-activity digraph $G = (V_N, A_N)$

$$d_{P2,P1} = \begin{pmatrix} 2 \end{pmatrix}, \; H_{P2,P1} = \begin{pmatrix} 1 \end{pmatrix}, \; d_{P1,B} = \begin{pmatrix} 2 \end{pmatrix}, \; H_{P1,B} = \begin{pmatrix} 1 \end{pmatrix},$$

$$H_{(P1,P2),(P2P1)} = \begin{pmatrix} 4 \end{pmatrix}, \; H_{(P2,P1),(P1P2)} = \begin{pmatrix} 3 \end{pmatrix}.$$

Example 2.5. *The extension of the network model to different running modes is shown in Figure 13. All potential running modes on the track from A to B can be seen in Figure 13. For simplification we do not show the complete headway relations in that figure, but of course all combinations need to be defined to ensure feasibility on that track. Furthermore, the event nodes involved in a turn around activity in station A and B are connected by dashed arcs. In a mathematical model we define a turn around as the change from arrival to departure nodes. From a railway operations point of view, a turn around is only performed if a train enters and leaves the station at the same side, e.g. a turn around has a minimum duration of 3 in station A and 7 in B. This shows that it is easy to extend the models to handle different minimum turn around times for each station side individually.*

All running time definitions on a track induce a headway definition. We can trivially bound the dimension of the headway matrix of a standard track by $|(C \times M) \times (C \times M)|$ and $2|(C \times M) \times C \times M|$ for a single way track, respectively. Due to the fact that only a relevant subset of running times and therefore also for headways times should be considered at a specific track, we suggest to use always sparse representations of these matrices H. Furthermore, we introduce useful definitions for headway matrices.

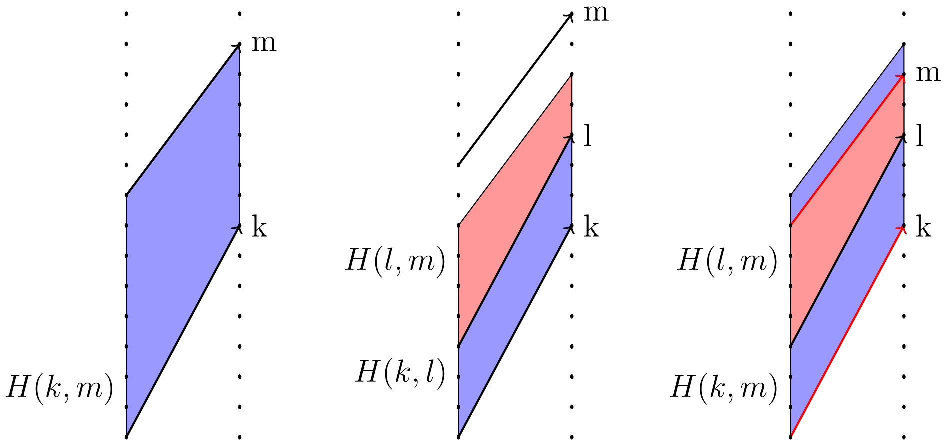


Figure 14: Implausible situation if headway matrix is not transitive

Definition 2.6. *A* headway matrix H_j *for track* $j \in J$ *is called* transitive or triangle-linear, *if all entries are strictly positive and the triangle inequality is satisfied:*

$$\forall c_1, c_2, c_3 \in C, m_1, m_2, m_3 \in M_J : \\ h_{j,c_1,m_1,c_3,m_3} \leq h_{j,c_1,m_1,c_2,m_2} + h_{j,c_2,m_2,c_3,m_3}.$$

Figure 14 motivates, why we can assume that headway matrices H to be transitive in reality. We use the simple notation $H(k,l)$ for the entry k,l that in fact corresponds to a preceding train type succeeding train type each with a certain running mode. On the left hand a train of type k is followed by a train of type m with respect to the minimum headway time $H(k,m)$. In the middle and on the right hand an intermediate train of type l is running on that track after k and before m. It can be seen that if $H(k,m) > H(k,l) + H(l,m)$ the track allocation on the left and in the middle are feasible. However, the sequence on the right is violating the headway $H(k,m)$. But it is completely implausible that running trains of type l after k on this track and trains of type m after l with respecting minimum headways, can become infeasible, due to violation of the minimum headway time of k and m. The algorithm presented in Section 2.3 produces headway matrices which are transitive simply because of the underlying block usages. In other words, if the situation on the right hand is a conflict between k and m based on timed resource usage of that track then the sequence k and l or the sequence l and m must already be in conflict.

macroscopic element	microscopic counterpart
train type c	subset of train routes R
station s	unified connected subgraph of G
track j (connecting different stations)	unified consecutive block sections, i.e., a path in G
running time on j for c (in Δ)	running times on block sections for routes (in δ)
headway times on j for pairs c_1, c_2 (in Δ)	blocking time on sections for routes (in δ)

Table 2: Relation between the microscopic and the macroscopic railway model

Definition 2.7. *A* headway matrix H_j *for track* $j \in J$ *is called* order-safe*, if all entries are strictly positive and the order is not changing (no passing on tracks):*

$$\forall c_1, c_2 \in C, m_1, m_2 \in M_J : h_{j,c_1,m_1,c_2,m_2} + d_{c_2,m_2} \leq d_{c_1,m_1} + h_{j,c_2,m_2,c_3,m_3}.$$

Definition 2.8. *A* headway matrix H *is called* valid*, if H is transitive and order-safe.*

We summarize the macroscopic infrastructure model that we have developed so far as consisting of a network $N = (S, J)$, with a set of relevant locations S, where train events occur, and the set of tracks J, where trains can run. Furthermore, we have seen how detailed macroscopic information for running, turn around, and headway times for a given set of train types C and modes M induce a digraph $G = (V_N, A_N)$ with $V_N \subset S \times \{1, 2\} \times M_S$ and $A_N \subset V_N \times V_N$. By definition all times are strictly positive integer values with respect to a fixed discretization, e.g., the times of the instances provided by the TTPlib are in minutes. The digraph $G = (V_N, A_N)$ represents all potential events and activities in $N = (S, J)$. All activities $a \in A_N$ have a positive duration $d(a) \in \mathbb{N}$. The restriction to only one train type $c \in C$ is denoted by $G|_c$. Finally, Table 2 identifies the macroscopic elements and their orginal microscopic counterparts with respect to the railway safety system and the railway infrastructure resource consumption.

2.2 Time Discretization

Discrete optimization models for timetabling and slot allocation are based on the use of *space-time graphs*, i.e., the time is discretized. Similar as for the topological aggregation, there is also a trade-off between model size and accuracy in the temporal dimension. This tradoff is controlled by the *discretization stepsize*. The discretized times in the macroscopic model will be based on microscopic simulation data, which is very precise. In fact, simulation tools provide running and blocking times with an accuracy of seconds (or even smaller). Our aim is to aggregate these values in the macroscopic model. We propose for this purpose a conservative approach, which means that running and arrival times will never be underestimated in the macroscopic model.

Simulation tools provide running and blocking times with an accuracy of seconds (or even smaller) denoted by δ. To decrease the problem size of real world instances, it is essential and a common approach to use a coarse time discretization in the macroscopic model. In addition we need a discrete model to handle decisions wether a train is running and blocking a section or not. In our approach the unit of the macroscopic time discretization is based on the microscopic simulation data. Let Δ be a fixed parameter to measure all macroscopic time information, e.g., units of 60 seconds. We propose again a conservatively approach, which means it is not valid to underestimate running, i.e., and therefore arrival times, in the macroscopic model. In the following, we denote by $\tilde{d}_j^r$ the microscopic running time of route r on track j, by d_j^r the discretized running time, and by ϵ_j^r the cumulative rounding error (in units of δ). The total rounding error at the end of each route is denoted by ϵ^r (in units of δ). A first approach would then be to simply round up all the times. The error estimation of this method is shown in Lemma 2.9.

Lemma 2.9. *Let $r \in R$ be a train route in the macroscopic network $N = (S, J)$ with length n_r, i.e., that is the number of macroscopic tracks of route r, and running times $\tilde{d}_j^r$ measured in δ for each track $j \in r$. If we simply round up the running times $\tilde{d}_j^r$ for each track to a multiple of Δ, we get a worst-case rounding error of $\Delta n_r - n_r$.*

Proof. For each track we have a maximum possible rounding error of $\Delta - 1$. In the worst-case this could occur to all n_r tracks of r. □

The error estimation shows that this rounding procedure results in rather big differences between the macroscopic and the microscopic running times. From a theoretical point of view we could assume to round up all the times so that we can always argue that the microscopic train would fit in the macroscopic planned time corridor by just slowing down. Unfortunately, this could lead to unnecessary overestimations of the running and headway times and thus to inefficient use of capacity.

Algorithm 1: Cumulative rounding method for macroscopic running time discretization

Data: track $j = (s_1, s_2) = (e_1, \ldots, e_m) \in J$ with $s_1, s_2 \in S$ and $e_i \in E, i \in 1, \ldots, m$, a train route $r \in R$ with microscopic running time $\tilde{d}_j^r > 0$ for track j, a cumulative rounding error ϵ_{j-1}^r and the time discretization $\Delta > 0$

Result: running time d_j^r and cumulative rounding error ϵ_j^r

begin

  choose $k \in \mathbf{N}$ **with** $(k-1)\Delta < \tilde{d}_j^r \leq k\Delta$;

  if $0 < (k-1)$ **and** $\tilde{d}_j^r - (k-1)\Delta \leq \epsilon_{j-1}^r$ **then**

    $d_j^r := (k-1)\Delta$; `/* round down */`

    $\epsilon_j^r := \epsilon_{j-1}^r - (\tilde{d}_j^r - (k-1)\Delta)$; `/* decrease error */`

  else

    $d_j^r := k\Delta$; `/* round up */`

    $\epsilon_j^r := \epsilon_{j-1}^r + (k\Delta - \tilde{d}_j^r)$; `/* increase error */`

  return $pair(d_j^r, \epsilon_j^r)$;

Therefore we use an alternative approach by a sophisticated rounding technique. The objective is to control the rounding error by only tolerating a small deviance between the rounded macroscopic running time and the microscopic one. The idea is pretty simple: with respect to the cumulative rounding error, it is sometimes allowed to round down, because enough buffer time was collected on the way. In that case, we know that the train can always arrive one time unit earlier at the target station of track j. Nevertheless, we have to make sure that no running time is rounded to zero, because this would imply no infrastructure usage and can lead to invalid timetables. The exact description of the procedure done at each track is given in Algorithm 1. Let denote by ϵ_{j-1}^r the absolute cumulative rounding error which cumulates all errors of r until the previous track $j-1$ on the route. At the beginning of a route r the cumulative rounding error clearly equals zero, i.e., $\epsilon_0^r = 0$. The macroscopic running times are in fact attributes of a track j. Hence we identify them by d_j^r, where d denotes that it is a running time attribute and r the related train route.

Lemma 2.10 states that this cumulative rounding technique gives a substantial better upper bound on the rounding error.

Lemma 2.10. *Let* $J^* = j_1, \ldots, j_{n_r}$ *with* $j_i = \{e_{i1}, \ldots, e_{im}\} \in J, i \in \{1, \ldots, n\}$, $e_{ik} \in E$, *be a train route* r *in the macroscopic network*

$N = (S, J)$ with microscopic running times $\tilde{d}_j^r > 0$ for each track j measured in $\delta > 0$.

If $\Delta \leq \tilde{d}_j^r \forall j \in J, r \in R$ for the time discretization Δ, the cumulative rounding error ϵ^r of the rounding procedure described in Algorithm 1 is always in the interval $[0, \Delta)$.

Proof. The proof is done by induction over the n_r tracks of route r. Consider the first track j_1 on r. The start rounding error is denoted by $\epsilon_0^r := 0$. It follows that $\tilde{d}_{j_1}^r - (k-1)\Delta > 0 = \epsilon_0^r$. Hence Algorithm 1 rounds up, and we get $\epsilon_{j_1}^r := k\Delta - d_{j_1}^r$. By definition of k, it follows that $0 \leq \epsilon_{j_1}^r < \Delta$ since $\tilde{d}_{j_1}^r > 0$.

In the induction step we analyze the rounding error of the track j_n denoted by $\epsilon_{j_n}^r$. There are two cases:

1. Let $\tilde{d}_n^r - (k-1)\Delta \leq \epsilon_{n-1}^r$. Then we round down and set

$$\epsilon_n^r := \epsilon_{n-1}^r - (\tilde{d}_n^r - (k-1)\Delta).$$

   By reason of the fact that $\Delta \leq \tilde{d}_j^r$ a rounding down to zero could not appear. By definition of k it clearly follows that

$$\epsilon_n^r < \epsilon_{n-1}^r < \Delta.$$

   And due to the “If ”condition in the algorithm it is obvious that

$$\epsilon_n^r = \epsilon_{n-1}^r - (d_n^r - (k-1)\Delta) \geq 0.$$

2. Consider the other case, that is $\epsilon_{n-1}^r < \tilde{d}_n^r - (k-1)\Delta$. Then ϵ_n^r is set to $\epsilon_{n-1}^r + (k\Delta - \tilde{d}_n^r)$. By $\tilde{d}_n^r \leq k\Delta$ it is evident that

$$0 \leq \epsilon_{n-1}^r \leq \epsilon_n^r.$$

   At last we have to consider the upper bound. It follows that

$$\begin{aligned} \epsilon_n^r &= \epsilon_{n-1}^r + (k\Delta - \tilde{d}_n^r) \\ &< \tilde{d}_n^r - (k-1)\Delta + k\Delta - \tilde{d}_n^r \\ &= \Delta. \end{aligned}$$

$\square$

With the above described rounding technique there is still one problem left. Lemma 2.10 does not apply for the case when there exists a track j where $\tilde{d}_j^r < \Delta$. Then it is not allowed to round down. This could imply a worse upper bound for our rounding procedure as shown in Lemma 2.11.

Lemma 2.11. *We consider the same rounding procedure and the same assumptions as in Lemma 2.10 except for the case that there is a set $B \subseteq \{1, \ldots, n_r\}$ where for each $b \in B$ $\tilde{d}_b^r < \Delta$ holds. Then the upper bound for the cumulative rounding error $\epsilon_{n_r}^r$ is equal to $(|B| + 1)\Delta$.*

Proof. We again use an induction technique. At the beginning we look at the first track, where $\tilde{d}_b^r < \Delta$. In this case we have $(k-1)\Delta = 0$ and therefore $k = 1$. Due to the prohibition that a macroscopic running time equals zero, we set $\epsilon_b^r := \epsilon_{b-1}^r + (k\Delta - \tilde{d}_b^r)$. It follows that

$$\begin{aligned} \epsilon_b^r &= \epsilon_{b-1}^r + (k\Delta - \tilde{d}_b^r) \\ &= \epsilon_{b-1}^r + (\Delta - \tilde{d}_b^r) \\ &< \Delta + \Delta - \tilde{d}_b^r \\ &< 2\Delta. \end{aligned}$$

Note that, as shown in Lemma 2.10, the rounding error does not grow, if the running time on the current track is greater than Δ.

Next we consider the case, that we have yet a number of i tracks with a running time less than Δ and the $i+1$ track is occurred. To simplify notations the precedent track is denoted by i. Then it follows that

$$\begin{aligned} \epsilon_{i+1}^r &= \epsilon_i^r + (k\Delta - d_{i+1}^r) \\ &= \epsilon_i^r + (\Delta - d_{i+1}^r) \\ &< i\Delta + \Delta - d_{i+1}^r \\ &< (i+1)\Delta. \end{aligned}$$

□

Figure 15 shows the difference between microscopic and macroscopic running time for a fixed value $t = 74$ at one track with respect to different macroscopic time discretizations Δ. Fine discretizations like less than 15 seconds produce only very small deviations. For larger

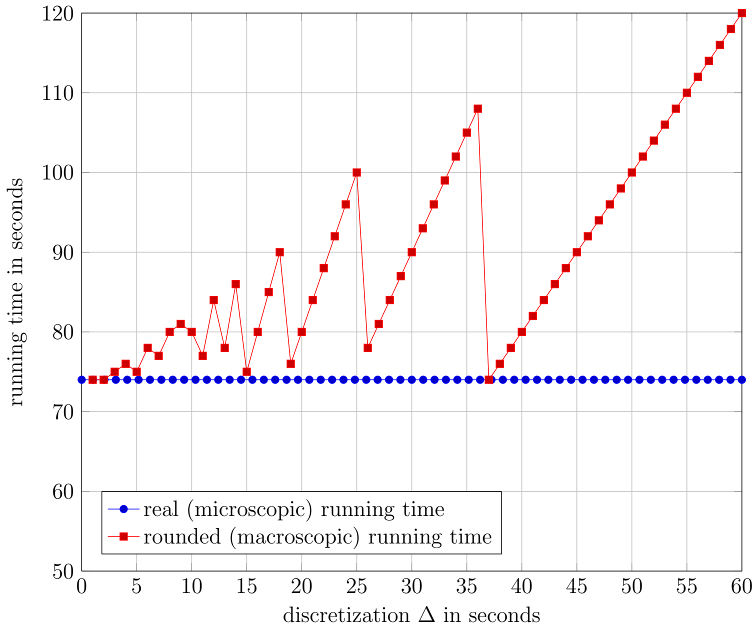


Figure 15: Transformation of running time on track $A \rightarrow B$ for time discretizations between 1 and 60 seconds

time discretization the error increases significantly, except for some pathological cases were t is a multiple of Δ.

Figure 16 compares the two rounding methods by illustrating the minimum, average, and maximum rounding errors of the macroscopic running times at the end of example routes for all considered train types through the Simplon corridor with respect to time discretizations varying from 0 to 60 seconds. The routes have a length of at most ten macroscopic tracks. It is apparent that cumulative rounding dampens the propagation of discretization errors substantially already for short routes.

We want to point explicitly that rounding up or down to the nearest integer number, i.e., in case of 1.5 to 2, would also limit the propagation of the rounding error on an individual route. However, this approach can not guarantee that the block sections can be allocated conflict-free with respect to the finer discretization δ. It is not hard to formulate a counterexample where rounding up and down come adversely together and lead to an invalid macroscopic model, e.g., a deadlock on a single way track. Hence there are feasible macroscopic allocations that can not be re-translated into feasible microscopic ones. Therefore results of such an approach are questionable and hardly transformable.

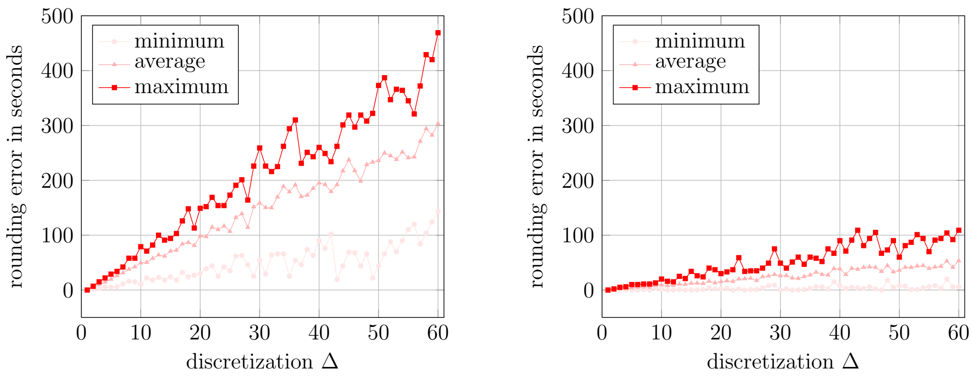


Figure 16: Rounding error for different time discretizations between 1 and 60 seconds, comparison of ceiling vs. cumulative rounding

---

Algorithm 2: Calculation of Minimal Headway Times

Data: Track $j = (s_1, s_2) = \cup_i e_i \in J$ with $s_1, s_2 \in S$, release duration $u_{e_i}^{r_1}$ and reservation duration $l_{e_i}^{r_2}$ with $r_1, r_2 \in R$, $c(r_1), c(r_1) \in C$, $e_i \in E, i \in 1, \ldots, m$, and time discretization $\Delta > 0$

Result: Minimal headway time $h(= h_{j,j,c(r_1),c(r_2)})$ for train type sequence $c(r_1), c(r_2)$ on track j

begin

  $h \leftarrow \infty$;

  for $x = \{\cup_i e_i | e_i \in r_1 \cap r_2\}$ **do**

    $h = \min\{u_x^{r_1} + l_x^{r_2}, h\}$; `/* update timing separation */`

  return $\lceil \frac{h}{\Delta} \rceil$;

Another important aspect for the macroscopic network transformation is the calculation of the headway times. Based on the occupation and release times in Figure 17 it is possible to define a minimal time difference after which a train can succeed on the same track or can pass it from the opposite direction. We want to point out explicitly, that we restrict ourself w.l.o.g. to minimal headway times for the combination of departure trains. In reality, especially railway engineers often use the term headway times for all kinds of potentially train event combinations for a reference point, e.g., the headway time between arrival of train 1 at station A and departure of train 2 at station B is 8 minutes.

Algorithm 2 describes the calculation of the *minimal headway time* for the cases of two routes r_1 and r_2 traversing the track in the same direction. We denote the corresponding train types by $c_1, c_2 \in C$.

In case of crossing routes r_1 and r_2 on track $j = (s_1, s_2)$ another headway time has to be considered. By definition each single way track j has exactly one counterpart $\bar{j} = (s_2, s_1) \in J$, which is directed in the opposite direction. In addition to the standard headway times related to each track j, this kind of track needs another headway matrix to ensure block feasibility with respect to the opposing direction. Let $j = (e_1, \ldots, e_m)$ be traversed by the directed route r_1. Obviously the minimum headway time for a departure of a train on route r_2 at station s_2 after a departure of a train on route r_1 from station s_1 is defined as

$$h_{j,\bar{j},c(r_1),c(r_2)} = \sum_{i \in \{1,2,\ldots m\}} d_{e_i}^{r_1} + u_{e_m}^{r_1} + l_{e_m}^{r_2}. \tag{1}$$

Note that in this opposing case the relevant block section is always e_m. In addition to the minimal technical headway time a standard buffer is added. Each network provider, such as DB or SBB, has a rule of thumb for this value. Nevertheless, the special knowledge and the experience of the planners can locally lead to more accurate numbers.

In Figure 17 the macroscopic output after the transformation for the situation described in Figure 6 can be seen. The infrastructure is reduced from six undirected block segments e_1 to e_6 to two directed tracks j_1 and j_2. Furthermore only two macroscopic stations are needed instead of seven microscopic nodes. On the microscopic scale the train movements are given very precisely. It is even possible to identify the acceleration, cruising, and deceleration phases. On the macroscopic scale train movements are linearized and only the state of the train at the start and at the end is controlled, i.e., we restrict ourself to two possible states, stopping and passing. In case of passing it is possible to traverse microscopic elements with different velocities and thus different durations for the same train type can occur. In order to receive a conservative macroscopic model we choose the calues for the “worst” passing.

This is a reasonable compromise between all possible passing states, which could be all allowed velocities between zero and a given maximum speed. This would unnecessarily increase the needed simulation runs, considered route data, and train type definitions. These aspects could be varied in a post-processing step after the macroscopic planning. However, a simple restriction to the “worst case” of traversing a track, that is train stops at the start and at the end, can lead to underestimation of the capacity and thus to wrong identification of

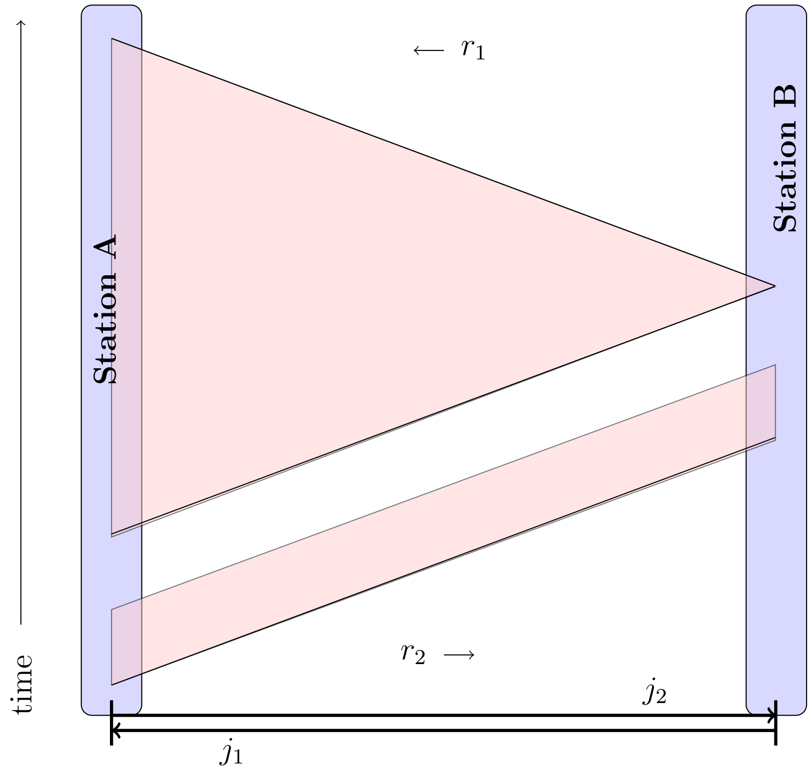


Figure 17: Headway time diagrams for three succeeding trains on one single track (j_1, j_2)

bottlenecks, as we have seen in Example 2.2. Therefore the durations of our macroscopic model depend on train types and events.

The blocking times are transfered into minimal headway times between train departures. Instead of controlling all blocking times in each block segment, we simplify the protection system to valid usages of the tracks. In Figure 17 the minimal headway times are illustrated for the given train sequence. Note that for the third and last train no headway area is plotted because no succeeding train is scheduled. Of course a forbidden area based on the blocking time stair of that train and a potentially succeeding train has to be considered.

2.3 An Algorithm for the Micro–Macro–Transformation

We developed an algorithm that carries out the transformation from the microscopic level to the macroscopic level. The whole procedure

Algorithm 3: Algorithm for the Micro-Macro-Transformation in `NETCAST`

```
Data: microscopic infrastructure graph G = (V, E), set of routes R, stations
      B(r), c(r) ∈ C, r ∈ R
Result: macroscopic network N = (S, J), with stations S, tracks J, and train types C
begin
ND    S_tmp := ∅; foreach r ∈ R do
          foreach b ∈ B^r do
              create s ;                                  /* create standard station */
              S_tmp = S_tmp ∪ {s};
      foreach (r1, r2) ∈ (R × R) do
          while divergence or convergence between r1 and r2 is found do
              create p ;                                  /* create pseudo station */
              S_tmp = S_tmp ∪ {p};
          while crossing between r1 and r2 is found do
              create p, q ;                               /* create pseudo stations */
              S_tmp = S_tmp ∪ {p, q};
AG    S := aggregateStations(S_tmp);
      J := {(s1, s2) ∈ S × S | ∃r ∈ R with s2 = nextStation(r, s1)};
TD    foreach j ∈ J do
          foreach r ∈ R do
              d_{j,c(r)} := calculateRunningTime(j, r, Δ);
          foreach (r1, r2) ∈ (R × R) do
              h_{j,j,c(r1),c(r2)} =
              max{h_{j,j,c(r1),c(r2)}, calculateHeadwayTime(j, r1, r2, Δ)};
              if j is single way then
                  h_{j,j̄,c(r1),c(r2)} =
                  max{h_{j,j̄,c(r1),c(r2)}, calculateHeadwayTime(j, j̄, r1, r2, Δ)};
      return N = (S, J);
```

is described in Algorithm 3. In the following, we will give some additional explanation to the algorithm. We skip the details on the different running modes to simplify the notation. There are three main steps, macroscopic network detection (ND), aggregation (AG), and time discretization (TD).

Macroscopic network detection means to construct the macroscopic digraph $N = (S, J)$ induced by R. Let $B(r)$ be the set of visited stations of route $r \in R$, i.e., locations (microscopic nodes) where the train stops and is allowed to wait. All visited stations are mandatory macroscopic station nodes. Note that after aggregation different microscopic nodes can belong to the same macroscopic station (area). If a conflict between two routes is detected at least one pseudo station is created. A conflict occurs not only in the case of converging or diverging routes, but especially if microscopic elements are used in both directions, e.g., if one route crosses another route. This detection is simply done by a pairwise comparison of the train routes. So in any case of using the same track in opposite directions, a conflict is detected and two pseudo stations are created to isolate the conflicting part. In the same way only one pseudo station is created if a con- or divergence occurs. The resulting set of stations S_{tmp} can be further aggregated. Note that microscopic nodes for each platform (affected by the routes) inside a station are contained in S_{tmp}. The routine $aggregateStations()$ in Algorithm 3 enforces the imaginable aggregations, as informal described in Section 2 to a station set S. Accordingly, the station capacities are defined in that function as well as the turn around times for the considered train types C.

After this step the macroscopic network detection with respect to the stations is finished. It remains to divide the routes R into sections, i.e., into tracks with respect to S. The subsequent station of node v on the train route r is denoted by $nextStation(r, v)$. For the creation of the tracks it is important to mention, that there could be more than one track between two macro stations, especially after aggregation steps, e.g, if there are two tracks between two aggregated macroscopic stations that could both be used by trains from the same direction. So a track is clearly identified by the starting and stopping microscopic (station) node and in addition to that by the set of microscopic arcs that were mapped to this track.

(TD), the calculation of the rounded running and headway times, is the last step of the algorithm. On track j we denote the running time of train route r by $d_j^r\,(= d_{j,c(r)})$, the headway time $h_{j,j,c(r_1),c(r_2)}$ for the self correlation case, i.e., when a train on route r_2 follows a train with route r_1, and the headway time for the single way case with $h_{j,\bar{j},c(r_1),c(r_2)}$. The running times are calculated by the cumulative rounding procedure $calculateRunningTime()$ is implemented by Algorithm 1. The function $calculateHeadwayTime()$ provides the headway times by Algorithm 2 and formula 1. For each route the running times, and for each pair of routes the headway times are calculated and (conservatively) aggregated according to the assignment of routes to train types $c \in C$. If there are several routes for the same train type, always the maximum time of the attribute is taken. The details on running modes have been omitted because it is only another technical question. Nevertheless, in NETCAST running and headway times with respect to running modes are implemented.

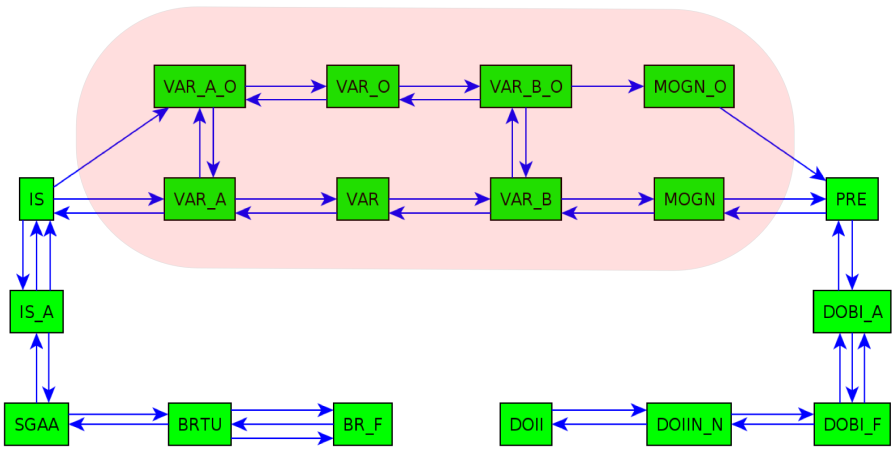


Figure 18: Constructed aggregated macroscopic network by NETCAST

In Figure 18 one of the macroscopic networks for the Simplon Tunnel generated by Algorithm 3 is shown. Finally, we summarize the resulting macroscopic data:

- ▷ (directed) network $N = (S, J)$ with stations, i.e., "station areas", S and tracks J
- ▷ mapping of subpaths of routes to tracks
- ▷ mapping of microscopic nodes to stations
- ▷ running time on tracks for all C measured in Δ
- ▷ headway time on all tracks for all pairs of C measured in Δ
- ▷ headway time on single way tracks for all pairs of C measured in Δ
- ▷ each micro element $e \in E$ corresponds to at most two (reversely directed) tracks
- ▷ each micro element $v \in V$ corresponds to at most one (pseudo) station

Remark 2.12. *The constructed (technical minimal) headway matrices H in* NETCAST *are valid, i.e., transitive and order-safe.*

Remark 2.13. *We developed our transformation tool* NETCAST *based on a given set of routes. The idea is to extract the components of these routes and map them to train types so that "new" routes can be constructed. Let routes from station A via C to D and from B via C to E for the same train type be given. Figure 19 shows the situation, i.e., both train routes stop at station C. After the transformation by* NETCAST *the macroscopic model can even handle trains from A to E and from B to D for that train type via re-combination. This allows to reduce the simulation effort to a standard set of patterns and routes.*

Remark 2.14. *Furthermore,* NETCAST *aggregates the microscopic infrastructure network as much as possible based on the set of routes, their overlappings, and their stopping pattern. In Figure 20 this is highlighted on several examples. On the left the macroscopic network is shown, which is produced by* NETCAST *if only High Speed Trains (EC) from Brig to Dommodossola and vice versa are considered. Due to the fact that no intermediate stopping for these trains is needed the macroscopic network shrinks to only two stations and two tracks (each per direction). In the middle, the same is done if you consider regional trains, which stops at some intermediate stations. On the right hand the final network for the Simplon with respect to all different types of trains can be seen. Note,: this is the same network as in Figure 18 only visualized in* TraVis *using the correct geographical coordinates.*

Remark 2.15. NETCAST *provides a re-translation of train paths from the macroscopic model to the microscopic model. That is the macroscopic path in $N = (S, J)$ will be transfered to a microscopic path in $G = (V, E)$. Note that in case of station aggregations some degree of freedom in choosing the precise routing inside a station occurs. Furthermore, the departure and arrival times of the macroscopic model which are given in Δ, are stated more precisely with respect to the original durations given in δ.*

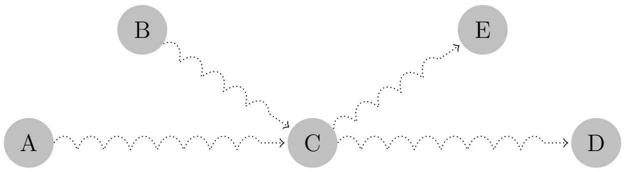
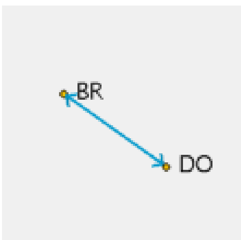
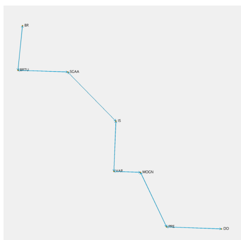


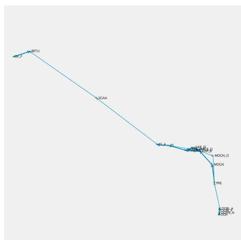
Figure 19: New routing possibilities induced by given routes



(a) only EC



(b) only R



(c) all train types

Figure 20: Macroscopic network produced by NETCAST visualize by TraVis

3 Final Remarks and Outlook

In this chapter we discussed a standard microscopic railway model and a novel macroscopic one that appropriately represents infrastructure resources and thus capacity. We introduced a convenient transformation approach which we implemented as the tool NETCAST. The big advantage is that the approach is generally applicable to any microscopic railway model, i.e., data of a standard microscopic railway simulation tool. In addition the reliability and quality of the results is obviously much higher in an integrated system than isolated applications. Our Micro-Macro Transformation algorithm detects the macroscopic network structure by analyzing interactions between standard train routes. In this way, the algorithm can ignore or compress parts of the network that are not used by the considered train routes, and still account for all route conflicts by constructing suitable pseudo stations. Time is discretized by a cumulative rounding procedure that minimizes the differences between aggregated and real running times.

Furthermore, we analyzed the error propagation of rounding procedures caused by the transformation and the more coarse discretization. Thus we can directly quantify the quality of a macroscopic railway model in comparison to the originated microscopic one. The impact of the time discretization of a railway model can be enormous. We will discuss this on several experiments in Chapter IV and Section 4.

However, with our approach a fixed discretization Δ can be determined to construct a macroscopic model with legitimated and reliable results. The question which fixed discretization one should choose arises in several optimization contexts, e.g., LPP and PESP, and is very rarely discussed. In most cases software systems in operation work with a fixed unit, i.e., minutes in most of the related literature. The work of Lusby (2008) [158] is exceptional who is using tints of 15 seconds. Hence it is an interesting field to evaluate discrete models, i.e., not only railway models, with respect to different time scales. Further developments will be to introduce a dynamic handling of discretization instead of a fixed approach to face up to the major challenge directly "inside" the solver.

Chapter III

Railway Track Allocation

In this chapter we introduce the track allocation problem, recapitulate several appropriate models from the literature and discuss them. A major contribution will be the development of an extended formulation, which yields computational advantages, especially for real world instances. We analyze the polyhedral relations of these models and present several extensions. Finally, a sophisticated algorithm for the extended formulation to solve the track allocation problem based on column generation techniques and the approximate bundle method will be presented.

The novel model approach is joint work with Ralf Borndörfer. Steffen Weider kindly provided an implementation of the *approximate bundle method* and of the *rapid branching heuristic* for set partitioning problems. This code was the basis of the adapted versions in `TS-OPT`, which has been implemented by the author of this thesis. This chapter summarizes the current state of our research, which has already been presented at conferences, i.e., Borndörfer & Schlechte (2007) [30, 31]; Borndörfer et al. (2006) [34]; Borndörfer, Erol & Schlechte (2009) [38]; Borndörfer, Schlechte & Weider (2010) [43]; Schlechte & Borndörfer (2008) [188]. It has already received considerable recognition in research on the track allocation problem, visible in recently published literature, e.g., Cacchiani (2007) [51]; Cacchiani, Caprara & Toth (2007) [52]; Cacchiani, Caprara & Toth (2010) [54]; Caimi (2009) [57]; Fischer & Helmberg (2010) [89]; Fischer et al. (2008) [90]; Klabes (2010) [129]; Kontogiannis & Zaroliagis (2008) [136]; Lusby (2008) [158]; Lusby et al. (2009) [159].

1 The Track Allocation Problem

The track allocation problem, also known as the train timetabling problem (TTP) in the literature, is the following problem: Given is an macroscopic railway model and a set of train slot requests. The (TTP) is to decide which subset of the train requests should be realized and what are the exact departure and arrival times of these trains. In this context a train slot is a path through the infrastructure network together with exact departure and arrival times for all visiting stations. Furthermore, it has to fulfill the requirements of the request specification. However, the precise definition will be evolved in this section.

Thereby, the solution schedule must be a track allocation which is feasible and optimal, i.e., the solution satisfies all operational macroscopic infrastructure constraints and maximizes a given objective, i.e., a "profit" function. This is a profit-oriented approach persecuted by network provider, governor or marketer in the near future, e.g., DB Netze AG [73],Trasse Schweiz AG [207], or ProRail [179].

One could also ask for a "cost-minimal" train schedule for given trains from an operator point of view. Online dispatching can also be seen as a track allocation problem as minimizing additional waiting times of the considered trains. Obviously, the real time dispatching problem has a different flavor, because it needs a different quality of data, and shorter solving times, but from a mathematical modeling point of view it is basically the same problem. We already discussed the related literature in Chapter I and Section 6.

One part of the input of the track allocation problem, the macroscopic railway model, was already presented in Chapter II and Section 2.1. The other one, the train demand specification, will be introduced in Section 1.1 of this chapter. Together they specify an instance of the train timetabling or track allocation problem, see Figure 1. This specification was developed as a general auction language for railway usage in Borndörfer et al. (2006) [34]. Furthermore it is used as a standardization for macroscopic train timetabling problems in the problem library TTPlib, see Erol et al. (2008) [85].

For passenger traffic which is mainly periodic and cross-linked, we refer to the work on partial periodic service intention, see Caimi (2009) [57]. In that setting the definition of connections and time dependencies between different trains, i.e., meetings of train slots, build the core

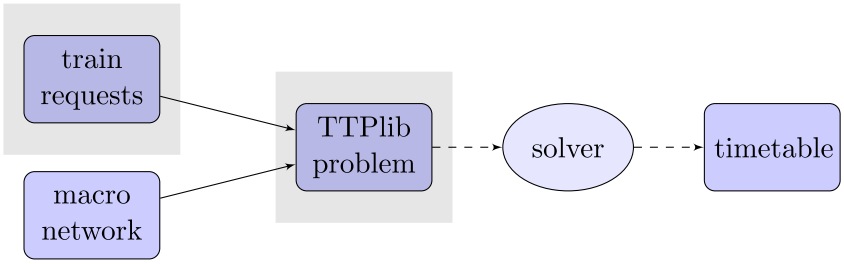


Figure 1: Concept of TTPlib 2008 (focus on train demand specification and TTP)

of the specification and models. For our purpose individual aspects are most relevant for example the requirements of cargo trains such as desired arrival times at certain stations or minimum dwelling times. Our specification is also influenced by the work of Schittenhelm (2009) [186] which provides an extensive discussion of quantifiable timetable aspects. Nevertheless, we will show how to integrate global schedule requirements like connections or periodic services in our models in Section 2.4. Section 1.2 gives a precise description and construction of an instance of the TTP by Definition 1.5.

1.1 Traffic Model – Request Set

Consider a basic setting that allows extensive valuation for individual *train slot requests* of the following general form. Denote by I the set of given train slot requests. Each slot request $i \in I$ specifies a train type $c_i \in C$, a *basic profit* $b_i \in \mathbb{Q}_+$, and a list of station *stops* with at least two elements, namely start and final destination. On the one hand for each stop mandatory definitions are required:

- ▷ station $s \in S$,
- ▷ minimum and maximum departure time, $t^{dep}_{min} \leq t^{dep}_{max} \in \mathbb{N}$.
- ▷ minimum and maximum arrival time, $t^{arr}_{min} \leq t^{arr}_{max} \in \mathbb{N}$,

On the other hand additionally optional intentions for each stop can be specified:

- ▷ optimal departure time, $t^{dep}_{opt} \in [t^{dep}_{min}, t^{dep}_{max}] \cap \mathbb{N}$,
- ▷ optimal arrival time, $t^{arr}_{opt} \in [t^{arr}_{min}, t^{arr}_{max}] \cap \mathbb{N}$,
- ▷ penalties for exceeding times, $p^{arr}_+, p^{dep}_+ \in \mathbb{Q}_+$ per time unit,
- ▷ penalties for falling below optimal times, $p^{arr}_-, p^{dep}_- \in \mathbb{Q}_+$ per time unit,
- ▷ minimum and maximum dwell time, $d_{min} \leq d_{max} \in \mathbb{N}$.

Finally, it is possible to guide certain attributes of the complete path by means of

- ▷ penalty for exceeding of minimum travel time $p_+^{travel} \in \mathbb{Q}_+$ per time unit,
- ▷ penalty for additional stops $p_+^{stops} \in \mathbb{Q}_+$.

By source of those parameters mainly the characteristics of individual cargo trains are reflected. We deliberately do not consider to specify relations between different trains, i.e., this is necessary for passenger trains, to keep the TTPlib simple. However, future challenges will be to incorporate passenger timetable optimization models like PESP in the specification of the TTPlib.

Train slots can be preferred which realize fast connections between origin and destination by choosing p_+^{travel} larger than zero. In Example 1.1 usual penalty functions are given and explained.

Analogously, it might be useful that slots on which the train has to unnecessarily brake and accelerate again are penalized by p_+^{stops}. Energy-saving, see Albrecht (2008) [10], is a hot topic in railway engineering from an operational point of view, but can also be considered in planning these slots to some extend. However, we restrict our consideration and input parameters to the list above, but of course some other aspects might also be interesting, e.g., penalties for exceeding the minimum route length to prefer direct and short routes.

Example 1.1. *Let the function on the left hand in Figure 2 specify the penalty ϵ for deviation from the optimal departure time at the first station of the train slot. It can be seen that shifting the departure time within the given time window by one time unit earlier than desired is more punished than departing by one time unit later. The function on the right hand could be useful to control an arrival event. No penalty ϵ is obtained for arriving before the optimal point, but exceeding that time at this stop is critical for the train and hence it is highly penalized. Figure 3 shows a simple profit function $w()$ with respect to a given basic profit b and both penalizations.*

Of course, the restriction of that framework to two-stepwise-linear functions is nonessential. The reason for that is to keep the definition of the objective function of any train request as simple as possible. This allows to define a huge range of different goals by just changing some parameters of each train request. Nevertheless, we want to point out

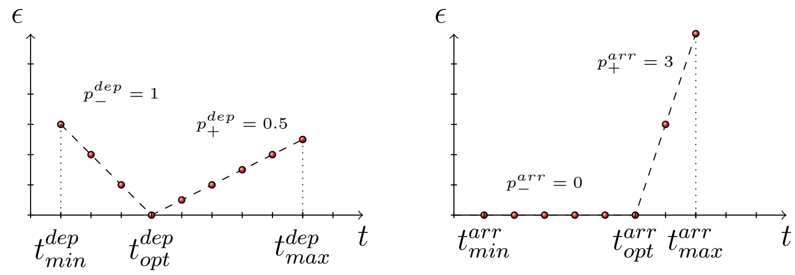


Figure 2: Penalty functions for departure(left) and arrival(right) times.

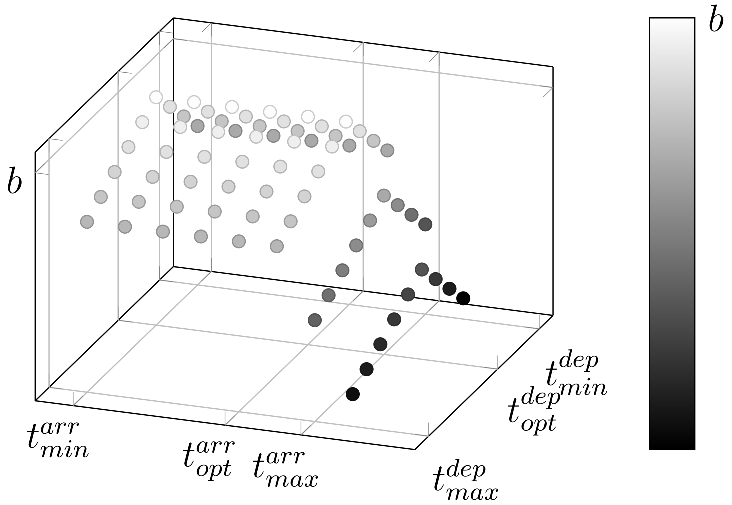


Figure 3: Profit function $w()$ depending on basic profit and departure and arrival times.

explicitly, that it would be possible to use much more complex nonlinear functions, because in the end these function evaluations only lead to different values for the objective coefficients of some arcs. However, the framework should not exceed a certain degree of complexity.

The goal for developing this framework is to give a train operator the possibility to specify easily their requirements with only a few parameters. It is an economic "bidding language" that enables train operating companies to express their train slot requests in a satisfactory, tractable, and flexible way. We present possible extensions to deal with combinatorial restrictions on the train request in a separate Section 2.4.

Finally, we want to clarify some easily mistakable terms for stopped trains. In the request specification we use the term dwell time which can either be a turn around activity or pure waiting. Due to the fact that this does not make a difference from an operator point of view we do not distinguish between them. However, for the consistency of train paths we have to handle turn around activities appropriately.

1.2 Time Expanded Train Scheduling Digraph

We expand our macroscopic railway model along a discretized time axis to model timetables in an event activity digraph $D = (V, A)$, the so called *train scheduling digraph*. All durations of $G = (V_N, A_N)$ and all times of I are given with respect to a constant discretization Δ, e.g., one minute. We construct multiple copies of the infrastructure node set V_N over a time horizon, one node set for each time and for each train request $i \in I$, i.e., we expand $G|_{c_i}$. The arcs A_N associated with train type $c_i \in C$ are also copied, connecting nodes in time layers that fit with the running or turn around times, as well as with the event definition. In that large scale digraph certain paths are realizations of requests, i.e., these graphs can easily have thousands of nodes and arcs even with a discretization of minutes. Sometimes we also use the term *path p implements request i*. By definition a request can be very flexible with respect to the route and the event times. We denote the set of implementing paths for request $i \in I$ by $\mathcal{P}_i$. The formal construction of $D = (V, A)$ is as follows:

We denote the time horizon by $T = \{t_0, \ldots, t_{\max}\} \subseteq \mathbb{N}$, i.e., t_0 is the first time of an event and $t_{\max}$ the last. The set of time-nodes associated to train request $i \in I$ is $V_i = \{(v, t) : v \in V_N, t \in T\} \subseteq V_N \times T$ with $V_N = S \times \{1, 2\} \times \{arr, dep, passing\}$, i.e., (v, t) is the copy of infrastructure event node $v \in V_N$ of side one or two and station $s \in S$ at time step t for request $i \in I$.

The next paragraphs will describe four different types of arcs $\mathtt{I}$ to $\mathtt{IV}$. Two time-nodes (u, τ) and (v, σ) are connected by a (running) time-arc $((u, \tau), (v, \sigma))$ of train type c_i if nodes u and v are connected by an arc $a \in A_N$ in the infrastructure network G. In addition the running time $d(a) = d_{j(a), c_i, m(a)}$ from u to v for a train of type c_i must be equal to $\sigma - \tau$ where $j(a)$ denotes the corresponding track of arc a and $m(a)$ the considered running mode, respectively. Note that node u can be of mode $\{dep, passing\}$ and v of mode $\{arr, passing\}$. We denote the set of running time-arcs by $A_{\mathtt{I}}$.

The second set of potential time expanded arcs are "real" turn around activities inside a station. Analogously, we connect time-nodes (u, τ) and (v, σ) by a time-arc $((u, \tau), (v, \sigma))$ of train type c_i if a turn around arc $a \in A_N$ in the infrastructure network is defined between this arrival and departure pair and $d(a) = \sigma - \tau$. Note that in this case node u must be an arrival and v a departure node on the same side of the station, i.e., $o(u) = o(v)$.

The third type of arcs is useful to model additional waiting. We distinguish between two possibilities:

- ▷ explicit waiting on a turn around arc from arrival to departure nodes,
- ▷ implicit waiting on a waiting time-line between departure nodes.

It depends on the considered degree of freedom which waiting policy is more reasonable. For train requests with a restrictive maximum waiting or dwell time at a station, i.e., most passenger trains, we suggest explicit waiting on turn around arcs between arrival nodes and departure nodes. The arrival node (v, τ) is then connected with departure node (u, σ), if a turn around arc $a \in A_N$ with duration $d(a) = d_{s,f}$ and $c_i \in f$ is defined in the infrastructure network between $v \in V_N$ and $u \in V_N$ and if $d_{min}(s, i) \leq d_{s,f} = \sigma - \tau \leq d_{max}(s, i)$. Hence, the duration of a waiting arc respects the given waiting interval for train i in station s and the minimal turn around time $d_{s,f}$. Note that in that model the total duration of a time expanded turn around arc consists of the time needed to perform the turn around[1] and a valid waiting expansion.

Remark 1.2. *Let m be the number of potential arrival points in time and n the number of departure points in time, then explicit waiting could lead to at most $m \cdot n$ turn around arcs.*

In cases where the length of the waiting interval inside a station could become arbitrary large and is a priori not bounded, we use a timeline concept. Timelines are applied to a lot of planning problems, where the number of potential arcs can become too large to handle them explicitly, see Desrosiers, Soumis & Desrochers (1982) [75]; Kliewer, Mellouli & Suhl (2006) [132]; Lamatsch (1992) [143]; Weider (2007) [213].

A turn around arc from each arrival node is created to enter the departure timeline on the other station side. Thus a minimum waiting time

[1]For the artificial case of $o(u) \neq o(v)$ the duration $d_{s,f}$ might be zero.

can be ensured. Note that these arcs are the only ones in $D = (V, A)$ with a potential duration of zero. The departure nodes $v \in V_N$ are the consecutively connected via waiting arcs time by time. In particular, waiting at node v is modeled by a time-arc $((v,t),(v,t+1))$ of type `IV` for all $t \in \{t_0, \ldots, t_{\max} - 1\}$.

Remark 1.3. *Let m be the number of potential arrival points in time and n the number of departure points in time, then implicit waiting could lead to at most $m + n - 1$ turn around and waiting arcs.*

In Figure 4 both model approaches are shown. The advantage of explicit waiting arcs is that not only minimum but also maximum duration can be handled. Furthermore, it is possible to define arbitrary objective values and attributes for each arrival and departure pair.

In a timeline this information is lost and decomposed. The arcs on the left in Figure 4 are replaced by the tree on the right. Each arc is represented by a path in the timeline and vice versa. Fortunately, in our setting the valuation and attributes of an arc are linear in the components of the representing path because of the dependence of time. Nevertheless, in an implicit waiting representation the control of the maximum waiting time is lost. This is compensated by a much smaller representation, see 1.2 and 1.3. Both representations are available in `TS-OPT`. However, default setting is to use the sparse timeline concept because a maximum waiting requirement is rather rare and can further be interpreted as a soft constraint in our instances. In the case that a hard maximum waiting is required it is possible to use the explicit model for that request. However, both arc types, i.e., `II` and `III`, are representing waiting with the difference that the first one connect arrival with departure nodes and the second one connect only departure nodes.

Finally, we define a dummy source node s_i and sink node t_i for each request $i \in I$. The source node s_i represents the start of request i and is connected via dummy arcs with all valid departure time-nodes $v = (s, o, m, \tau) \in V$. Node v must be a departure (or passing[2]) one with s equal to the start station of $i \in I$ and τ must be inside the given departure time window. Analogously, we connect a valid node v with sink t_i if v is an arrival (or passing) node of the final station of i and if τ is inside the arrival time window.

[2]Passing nodes are allowed at begin or end to handle “fly in” or “fly out” traffic.

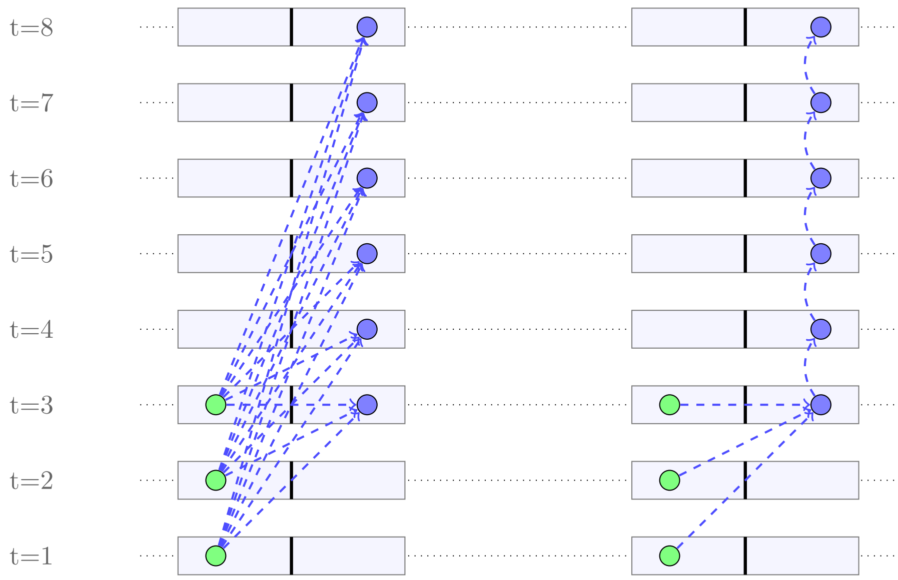


Figure 4: Explicit and implicit waiting on a timeline inside a station

To simplify the notation, we denote the time of time-node $v \in V$ by $\tau(v)$, which is the last element of this 4-tupel. Analogously, we use $m(v), s(v)$, and $o(v)$ as a mapping to access the event m, station s, and the side or orientation of $o(v)$ of node v. In the same way we denote the track, mode, and train type of a running arc $a \in A_{\mathtt{I}}$ by $j(a), m(a)$, and $c(a)$, respectively.

Due to this construction, we can partition the set of arcs A_i with respect to the four following arc types:

- `I` running arcs on tracks $j \in J$,
- `II` turn around arcs inside stations $s \in S$,
- `III` waiting arcs inside station $s \in S$,
- `IV` artificial arcs for begin and end of a train request $i \in I$.

Note that by definition $s(u) = s(v)$ for all $(u, v) \in A_{\mathtt{II}} \cup A_{\mathtt{III}}$ and $s(u) \neq s(v)$ for all $(u, v) \in A_{\mathtt{I}} \cup A_{\mathtt{IV}}$ with $s(s_i) = s(t_i) = \emptyset$, respectively. To make the notation clear, we use sometimes the set A_i, which is the subset of all time-arcs related to request $i \in I$. By A_I the set of all running arcs $a \in A$ are denoted. Thus the set of arcs A is a disjunctive union $\dot\cup_{i \in I} A_i$, as well as $A = A_{\mathtt{I}} \dot\cup A_{\mathtt{II}} \dot\cup A_{\mathtt{III}} \dot\cup A_{\mathtt{IV}}$.

Furthermore, we associate with each arc $a \in A$ an utility or profit value w_a, which reflects the objective parameters of the request definition. The idea is that the profit or utility value w_p of a path $p \in \mathcal{P}_i$, which implements request $i \in I$, can be expressed as the sum of all incident components, i.e., this value is linear with respect to incident arcs:

$$w_p = \sum_{a \in p} w_a.$$

To avoid unnecessary notational overhead, we restrict ourselves to the basic case of two mandatory stops, that is, departure at origin and arrival at destination station. The special case where a train request asks for more than two stops can be appropriately reduced to the basic case stop by stop. However, to ensure that each intermediate (station) stop is visited in an $s_i - t_i$-path, several copies of time-nodes have to be considered.

Let v_i^{travel} be the optimal values[3] for the duration of the requests $i \in I$, that is the difference between arrival time at final station and departure time at first station of request $i \in I$. Then, the objective values w_a of $a = (u, v) \in A_i$ are defined, as follows:

$$w_a = \begin{cases} -p_+^{travel}(\tau(v) - \tau(u)) & \text{, if } a \in A_{\mathrm{I}} \cup A_{\mathrm{II}} \cup A_{\mathrm{III}}, \\ b_i + v_i^{travel} p_+^{travel} - p_-^{dep}(t_{opt}^{dep} - \tau(v)) & \text{, if } a \in A_{\mathrm{IV}}, u = s_i, \tau(v) \leq t_{opt}^{dep}, \\ b_i + v_i^{travel} p_+^{travel} - p_+^{dep}(\tau(v) - t_{opt}^{dep}) & \text{, if } a \in A_{\mathrm{IV}}, u = s_i, \tau(v) \geq t_{opt}^{dep}, \\ -p_-^{arr}(t_{opt}^{arr} - \tau(v)) & \text{, if } a \in A_{\mathrm{IV}}, v = t_i, \tau(u) \leq t_{opt}^{arr}, \\ -p_+^{arr}(\tau(v) - t_{opt}^{arr}) & \text{, if } a \in A_{\mathrm{IV}}, v = t_i, \tau(u) \geq t_{opt}^{arr}. \end{cases}$$

The result is a space-time network $D = (V, A) = \bigcup_{i \in I}(V_i, A_i)$ in which train slots correspond to directed paths, proceeding in time. In particular directed paths from s_i to t_i are slot realizations of train request $i \in I$.

Observation 1.4. *The train scheduling graph $D = (V, A)$ is acyclic and therefore there exists a topological order of the nodes.*[4]

Obviously, we have to perform the time expansion in an efficient manner because of the enormous number of potential nodes and arcs. The idea is to identify non-redundant station nodes and track arcs for each request individually in a first step. A priori shortest path computations,

[3] These can easily be determined by appropriate shortest path computations with respect to the duration in $G = (V_N, A_N)$.

[4] Even if we allow (artificial) turn around inside a station, which could have a duration of zero, the strong monotony of time on all other arcs, especially all outgoing arcs of departure nodes, prevent cycles.

Algorithm 4: Construction of D.

Data: network $N = (S, J)$ and requests I (discretized in Δ)
Result: train scheduling graph $D = (V, A)$
init $V \leftarrow \emptyset$, $A \leftarrow \emptyset$;
foreach i *in* I **do**
  compute time expansion of $D_i = (V_i, A_i)$;
  compute irreducible digraph $D_i = (V_i, A_i)$;
  compute profit maximizing path in $D_i = (V_i, A_i)$;
set $D = \bigcup_{i \in I} D_i$;

i.e., for each train type, help to avoid time expansion in unnecessary directions of the network (V_N, A_N).

After this trivial route preprocessing we only perform the time expansion of the remaining network part to reduce the number of considered time-arcs and time-nodes. Finally invalid sources, which are not connected to at least one valid sink, or invalid sinks, which cannot be reached by at least one source, are eliminated.

Figure 5 shows an example, i.e., in network HAKAFU_SIMPLE, for a train routing graph, before preprocessing, with 123 potential event nodes and 169 activity arcs. The corresponding train wants to depart from FSON in time interval $[0, 5]$ and arrive at station FCG in time interval $[0, 15]$. Depicted are all potential event nodes (station, event, side, time) which are reachable from the dummy source s in the given time window. After preprocessing, the graph shrinks to 12 nodes and 13 arcs, see Figure 6.

Algorithm 4 spans the graph for each individual train request $i \in I$ stop by stop, i.e, from the first station to next specified stop of the request, and produces an irreducible graph representation $D_i = (V_i, A_i)$ for request $i \in I$. In particular no redundant time nodes or arcs are present. Furthermore, we compute a profit-maximizing path for each request $i \in I$, that is, a longest path with respect to weights w in each acyclic digraph D_i. The sum of these values is a trivial a priori upper bound of the TTP. In Example 1.6 and in Figure 7 a preprocessed network $D = (V, A)$ is shown in detail.

The space-time network $D = (V, A)$ can also be used to make all potential *conflicts* between two or more train slots explicit. In fact, each

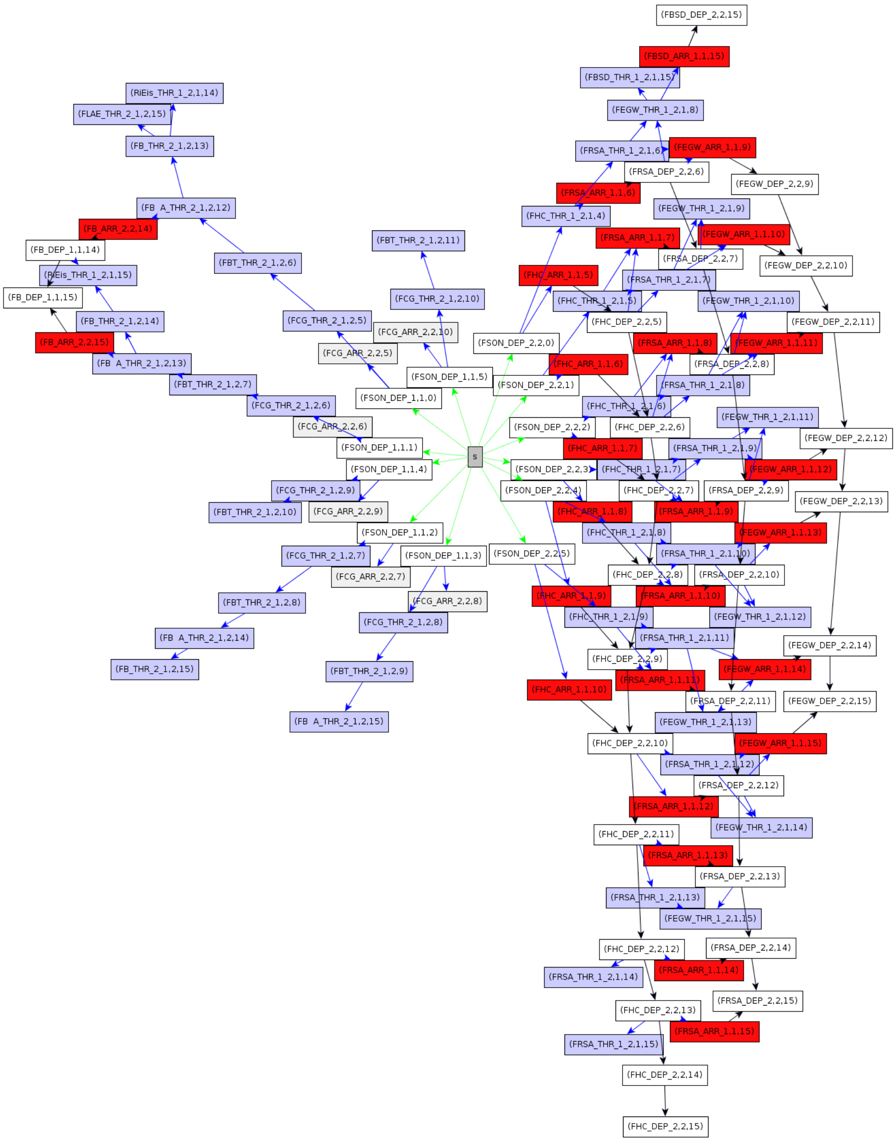


Figure 5: Complete time expanded network for train request

conflict corresponds to timed resource consumption on tracks or inside stations and can be defined by an appropriate subset of time-arcs A.

For a potential headway conflict on a track, consider two train slots of type c_1 and mode m_1 and type c_2 and mode m_2 departing from the track $j \in J$ via arcs $a_1 \in A$ and $a_2 \in A$, arriving at times t_1 and t_2, respectively; w.l.o.g. let $t_1 \leq t_2$. There is a headway conflict between these slots if $t_2 < t_1 + h_{j,c_1,m_1,c_2,m_2}$. This conflict can be ruled out by stipulating the constraint that a conflict free set of slots can use only

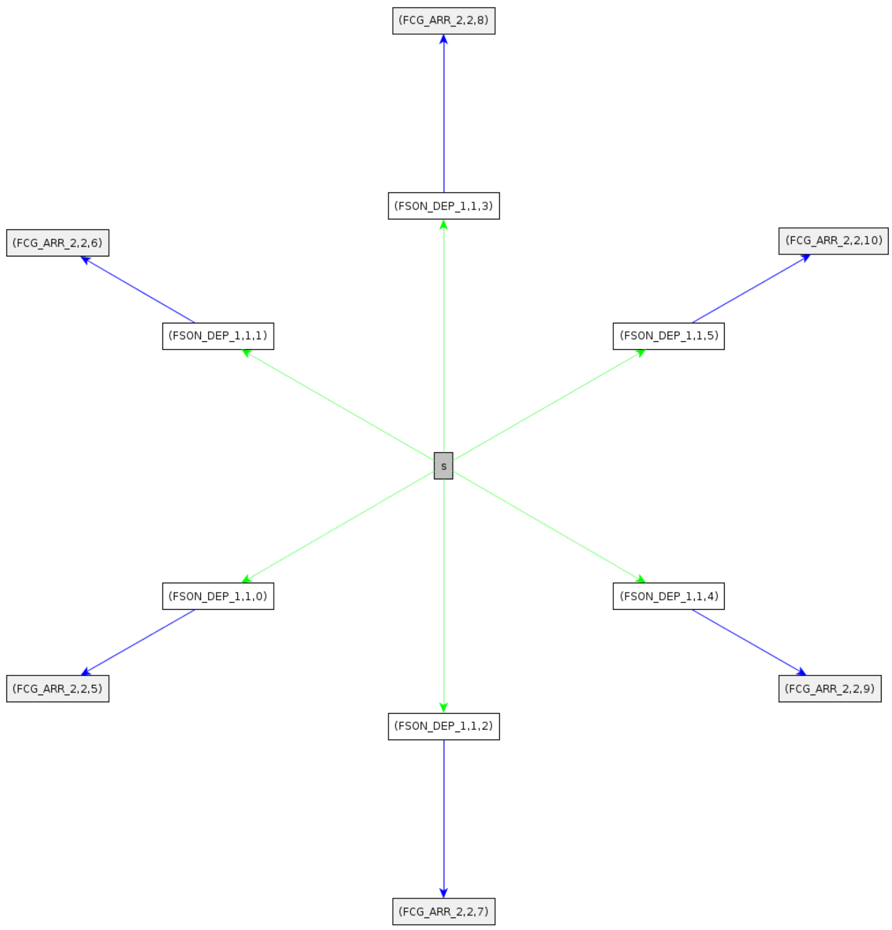


Figure 6: Irreducible graph for train request

one of the arcs a_1 and a_2. Doing this for all pairs of conflicting arcs enforces correct minimum headways.

For a station capacity conflict, consider train slots p_i of train type $c_i \in C$, $i = 1, \dots, k$, entering station $s \in S$ with capacity $\kappa_{s,f}, c_i \in f$ at time t. The capacity at time t is exceeded if more than $\kappa_{s,f}$ trains belonging to that train set are present at this station at time t. Note that we assume that departing trains at time t do not count at time t because they are leaving the station at this moment.

This conflict can be ruled out in a similar way as before by stipulating the constraint that a conflict free set of slots can use at most $\kappa_{s,f}$ of the following arcs:

▷ $uv \in A_{\mathtt{I}} \cup A_{\mathtt{IV}}$, which enters an arrival or a passing node v of station s at time t, i.e., $s(v) = s$ and $\tau(v) = t$,

▷ $uv \in A_{\text{II}} \cup A_{\text{III}}$, which starts before time t, i.e., $\tau(u) < t$, and ends after time t, i.e.,$\tau(v) > t$.

This definition for a general station capacity $\kappa_{s,f}$ illustrates the flexibility of the model and the possibility to handle more specific station restrictions, which can easily be modeled by appropriate definitions of the restricted subset of A.

Cacchiani (2007) [51] construct conflicts sets for consecutive arrivals, consecutive departures, and overtaking situations at certain intermediate stations.

Fischer et al. (2008) [90] consider, for instance, station capacities depending on the side of the station to control the incoming trains per direction.

This flexibility of the conflict sets is not needed if the network corresponds exactly to the microscopic infrastructure as in the work of Brännlund et al. (1998) [44], Lusby (2008) [158], and Fuchsberger (2007) [94]. However, on this scale only small scenarios can be handled and further requirements which are arising non-naturally, e.g., forbidden meetings of trains, are very hard to incorporate.

We denote an arbitrary conflict by γ, the set of all conflicts by Γ, the set of conflict arcs associated with conflict γ by A_γ, and the maximum number of arcs from A_γ that a conflict-free set of slots can use by κ_γ. If a chosen set of $s_i - t_i$ paths is conflict-free with respect to Γ, we sometimes use the term *simultaneously feasible*. The train timetabling or track allocation problem can then be defined as follows:

Definition 1.5. *Given train slot requests I, a corresponding digraph $D = (V, A)$, a profit value w_a for each time-arc $a \in A$ and an explicit definition of conflicts Γ on the time-arcs A, the problem to find a conflict-free maximum routing from s_i to t_i is called* optimal track allocation problem. *In other words, we seek for a profit-maximizing set of simultaneously feasible $s_i - t_i$ paths in $D = (V, A)$.*

This is a natural and straightforward generalization of the train timetabling problem described in Brännlund et al. (1998) [44], Caprara, Fischetti & Toth (2002) [62], and Caprara et al. (2007) [64] to the case of networks. There, only the case of a single, one-way track corridor is considered. For convenience, we will use the acronym TTP to denote the optimal track allocation problem. It was shown in Caprara, Fischetti & Toth (2002) [62] that the TTP is $\mathcal{NP}$-hard, being a gen-

request	basic value	train type	stop at station	time window preferences $(t_{min}, t_{opt}, t_{max}, p_-, p_+)$
blue	10	PT	X	$(1, 3, 4, 1, 2)$
			Z	$(3, 5, 6, 0, 1)$
red	10	CT	X	$(1, 3, 3, 2, 0)$
			Z	$(5, 6, 7, 2, 0)$

Table 1: Definition of train request set

eralization of the well-known maximum stable set problem, see Garey & Johnson (1979) [97].

Example 1.6. *Consider again a tiny network graph consisting of three stations and only two tracks. Assume that the infrastructure can be used by two different train types, called blue and red, who need one respectively two time units to pass the given tracks, and each has to respect a minimal headway of one minute on each track. For simplification, these trains can only perform a running mode of type 1. With the introduced notation, we have given:*

$$\begin{array}{lr}
\textit{stations} & S = \{X, Y, Z\}, \\
\textit{tracks} & J = \{(X(2), Y(1)), (Y(2), Z(1))\}, \\
\textit{train types} & C = \{PT,\ CT\}, \\
\textit{running times} & d_{j,PT,1} = 1,\ d_{j,CT,1} = 2, \forall j \in J \textit{ and} \\
\textit{minimal headway times} & h_{j,c_1,1,j,c_2,1} = 1, \forall j \in J,\ c_1, c_2 \in C.
\end{array}$$

We consider two train requests. Both should start in station X and target station Z, and are allowed to stop in Y for an arbitrary time. The first train should start in the time interval $[1, 4]$ and arrive in the window $[3, 6]$, while the second train should depart in $[1, 3]$ and arrive in $[5, 7]$. As we see, we obtain a time horizon of $T = \{1, 7\}$ for the total train routing graph. In Table 1 the preferences and valuations of the requests are listed, which consists only of a basic value and penalties for scheduled arrival and departure times. The graph $D = (V, A)$ produced by Algorithm 4 is shown in Figure 7.

The given request valuations of Table 1 were transferred to objective weights w_a of the time-arcs, see labels in Figure 7. In this example only the artificial begin and end arcs of the "train routing" flow have values w_a different from zero.

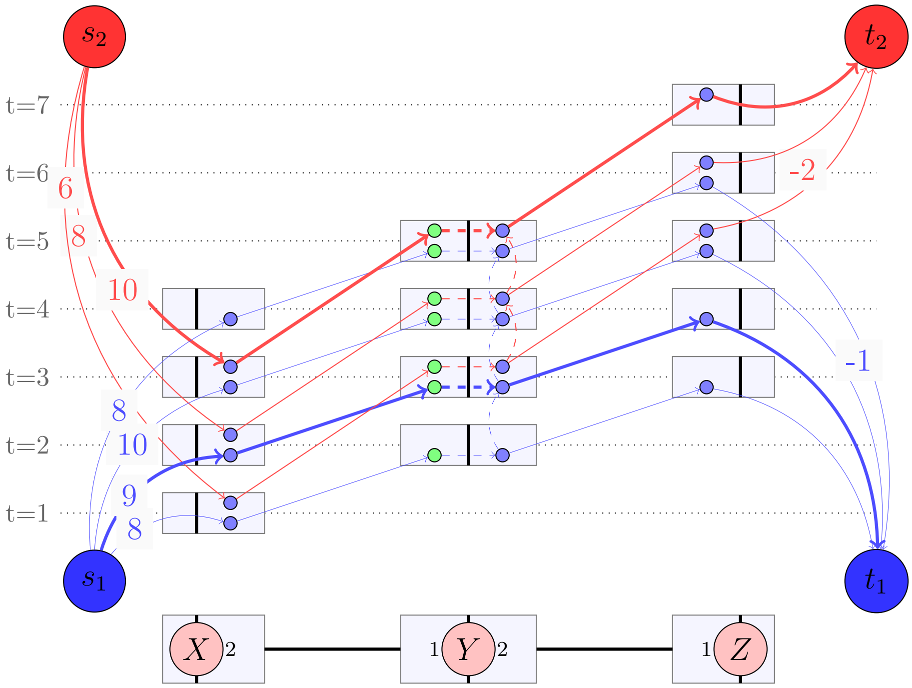


Figure 7: Preprocessed time-expanded digraph $D = (V, A)$ of example 1.6

The optimal track allocation problem is then to find a utility maximizing set of conflict-free $s_i - t_i$ -flows. Here is a complete list of the conflict set Γ:

$$\begin{aligned}
\gamma_1 &= \{((X,2,dep,1),(Y,1,arr,2)),((X,2,dep,1),(Y,1,arr,3))\},\\
\gamma_2 &= \{((X,2,dep,2),(Y,1,arr,3)),((X,2,dep,2),(Y,1,arr,4))\},\\
\gamma_3 &= \{((X,2,dep,3),(Y,1,arr,4)),((X,2,dep,3),(Y,1,arr,5))\},\\
\gamma_4 &= \{((Y,2,dep,3),(Z,1,arr,4)),((Y,2,dep,3),(Z,1,arr,5))\},\\
\gamma_5 &= \{((Y,2,dep,4),(Z,1,arr,5)),((Y,2,dep,4),(Z,1,arr,6))\},\\
\gamma_6 &= \{((Y,2,dep,5),(Z,1,arr,6)),((Y,2,dep,5),(Z,1,arr,7))\}.
\end{aligned}$$

The best path for the red and blue request has value 10 each, but unfortunately the simultaneous routing on track $(X(2), Y(1))$ is invalid with respect to the headway conditions, i.e., the red and the blue train want to departing at node $X(2)$ at time 3. To finish the example, an optimal solution, realizing a profit value 19, is to schedule the blue train on path

$$p_1 = (s_1, (X, 2, dep, 3), (Y, 1, arr, 5), (Y, 2, dep, 5), (Z, 1, arr, 7), t_1)$$

with utility value $w_{p_1} = 10$*, and the red one on path*

$$p_2 = (s_2, (X, 2, dep, 2), (Y, 1, arr, 3), (Y, 2, dep, 3), (Z, 1, arr, 4), t_2)$$

with $w_{p_2} = 9$*, respectively.*

2 Integer Programming Models for Track Allocation

Section 2.1 discusses standard integer programming formulations to the track allocation problem based on the train scheduling graph $D = (V, A)$. Furthermore, we develop an alternative formulation to take advantage of the structure of the headway conflicts in Section 2.2. Due to the very large size of real world problem instances, static arc formulations are limited. To overcome this limitation, path versions are often formulated. These are suitable to be solved by sophisticated column generation approaches or approximate bundle methods as we will present in Section 3.

In Section 2.3 the models are theoretically compared and analyzed. We will also show that our coupling formulations are extended formulations of the original packing formulations. Finally, we present several practical extensions to the problem and models in Section 2.4.

2.1 Packing Models

As mentioned before operational railway safety restrictions can be handled by conflict sets in $D = (V, A) = \bigcup_{i \in I} D_i$. This modeling approach was introduced by the pioneer works of Brännlund et al. (1998) [44] and Caprara et al. (2006) [63] on railway track allocation. Each conflict $\gamma \in \Gamma$ consists of a subset of arcs $A_\gamma \subseteq A$ and an upper bound $\kappa_\gamma \in \mathbb{Z}$. To formulate the track allocation problem as an integer program, we introduce a zero-one variable x_a (i.e., a variable that is allowed to take values 0 and 1 only) for each arc $a \in A_i$. If x_a takes a value of 1 in an $(\mathrm{APP})'$ solution, this means that a slot request i associated with arc a passes through arc a; clearly, this implies that slot request i has been assigned. On the other hand $x_a = 0$ means that arc a is not used by a slot associated with slot request i, independently of whether slot request i is assigned or not. Furthermore, we are given w_a for each arc a of slot request i in order to account for the overall proceedings or utility of a track allocation. Let us finally denote by $\delta_{\text{in}}(v) := \{(u,v) \in A_i\}$ the set of all arcs entering a time-node $v \in V_i$. Similarly, let $\delta_{\text{out}}(v) := \{(v,w) \in A_i\}$ be the set of arcs leaving time-node v. With these definitions and the notation of Section 1 the track allocation problem can be formulated as the following integer program:

$$
\begin{array}{llllr}
(\mathrm{APP})' & & & & \\
\max & \displaystyle\sum_{i\in I}\sum_{a\in A_i} w_a x_a & & & \text{(i)} \\
s.t. & \displaystyle\sum_{a\in\delta_{\text{out}}(s_i)} x_a & \leq 1, & \forall i \in I & \text{(ii)} \\
 & \displaystyle\sum_{a\in\delta_{\text{in}}(t_i)} x_a & \leq 1, & \forall i \in I & \text{(iii)} \\
 & \displaystyle\sum_{a\in\delta_{\text{out}}(v)} x_a - \sum_{a\in\delta_{\text{in}}(v)} x_a & = 0, & \forall v \in V_i \setminus \{s_i, t_i\}, i \in I & \text{(iv)} \\
 & \displaystyle\sum_{a\in A_\gamma} x_a & \leq \kappa_\gamma, & \forall \gamma \in \Gamma & \text{(v)} \\
 & x_a & \in \{0,1\}, & \forall a \in A_i, i \in I & \text{(vi)}
\end{array}
$$

In this model, the integrality constraints (vi) state that the arc variables take only values of 0 and 1. Constraints (ii)–(iv) are flow constraints for each slot request i; they guarantee that, in any solution of the problem, the arc variables associated with slot request i are set to 1 if and only if they lie on a path from the source s_i to the sink node t_i in $D = (V, A)$, i.e., they describe a feasible slot associated with slot request i. They are all set to 0 if no slot is assigned to slot request i. Note that constraints (iii) are redundant because $(\mathrm{APP})'$ (ii) and $(\mathrm{APP})'$ (iv) already define the flow, see Ahuja, Magnanti & Orlin (1993) [5]. Constraints (v) rule out conflict constraints, as described before.

The objective function (i) maximizes total network utility by summing all arc utility values w_a. This integer program can be seen as a "degenerate" or "generalized" multi-commodity-flow problem with additional arc packing constraints. In the sense that even though all train flows are individual longest path problems in acyclic digraphs D_i, they are connected by conflict set Γ and constraints (v), respectively.

As we already mentioned, Caprara et al. (2001) [61] and Caprara, Fischetti & Toth (2002) [62] defined conflict sets for departures, arrivals, and overtakings to ensure operational feasibility. Although this formulation allows for a very flexible definition of conflicts, a disadvantage of model (APP)$'$ is the "hidden structure", the detection, and the potentially large size of Γ. We will examine this issue for the case of headway conflicts, for which constraints (v) are *packing constraints*, i.e., $\kappa_\gamma = 1$. This can be done as follows. We create a conflict graph $\Lambda = (A_{\mathtt{I}}, E)$ with node set $A_{\mathtt{I}}$ of all running time-arcs. As already described in Section 1, for a potential headway conflict on a track $j \in J$, we can consider two train slots of type c_1 and mode m_1 and type c_2 and mode m_2 departing from the track j via arcs $a_1 \in A$ and $a_2 \in A$, arriving at times t_1 and t_2, respectively. W.l.o.g. let $t_1 \leq t_2$, then there is a headway conflict between these slots if $t_2 < t_1 + h_{j,c_1,m_1,c_2,m_2}$.

Each pair of conflicting arcs a_1 and a_2 defines an edge $(a_1, a_2) \in E$ and a corresponding conflict set γ containing both time-arcs and an upper bound $\kappa_\gamma = 1$. Doing this for all pairs of conflicting arcs enforces correct minimum headways. We denote this preliminary model by (APP)$'$, because further observations will lead to much stronger formulations.

It is clear that these pairwise conflict sets can be enlarged to inclusion-maximal ones which correspond to cliques in Λ. In the following we will collect some basic facts about detection and occurrence of maximum cliques in special graph classes. The statements translate directly into our setting. The case of "full block occupation" can be seen as the simplest one, that is, the headway time is set to the corresponding running time of the train. Keep in mind that in this setting headways are completely independent from the type of the successor train, they depend only on the departure time. The graph Λ becomes an interval graph. Figure 8 illustrates the construction of Λ and the maximal cliques in that case.

Lemma 2.1. *In a block occupation model all maximal conflict sets can be found in polynomial time since Λ is an interval graph.*

Proof. The cliques in the conflict graph are collections of compact real intervals. By Helly's Theorem, see Helly (1923) [113], the intervals of each such clique $\gamma \in \Gamma$ contains a common point $p(\gamma)$, and it is easy to see that we can assume $p(\gamma) \in \tau(A_{\mathtt{I}}) = \{\tau(v) : v \in A_{\mathtt{I}}\}$. It follows that the conflict graph Λ has $O(A_{\mathtt{I}})$ inclusion maximal cliques, which can be enumerated in polynomial time. In Booth & Lueker (1976) [27]

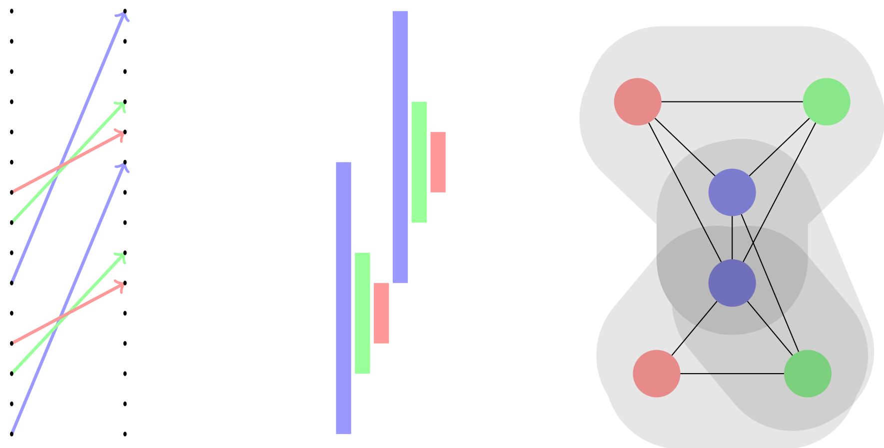


Figure 8: Example for maximum cliques for block occupation conflicts.

and Habib et al. (2000) [107] linear time recognition algorithms can be found. □

Example 2.2. *In Figure 8 the relation between headway conflict sets on a track $j \in J$ and the corresponding conflict graph Λ is shown. On the left hand, six trains are shown with the corresponding departure and arrival times. In the middle, the blocked intervals are projected. On the right hand, the induced conflict graph Λ can be seen. Furthermore, we highlighted all maximal cliques in that small example by shaded areas.*

Observation 2.3. *The train timetabling problem with full block occupation conflicts on a single track is equivalent to finding a maximum independent set in interval graphs.*

In general the separation of the maximal clique constraints is not trivial. This is because the entries[5] of the headway matrix H are in general different for each train type and for each stopping behavior combination.

Furthermore, realistic minimal headway matrices as presented in Section 2.1 are transitive, see Definition 2.6, and in the majority of cases asymmetric. Lukac (2004) [157] gives an extensive analysis of the structure of clique constraints arising from triangle-linear and quadrangle-linear matrices and proves that the time window of interest is bounded by twice the maximum headway time. However, in realistic cases this can be quite large. Since the number of constraints $(\text{APP})'(v)$ can be exponential in the number of arcs Fischer et al. (2008) [90] pro-

[5] In case of full block occupation all entries are equal to the corresponding running time.

pose to use a greedy heuristic to find large violated cliques. Note that constraints $(\mathrm{APP})'(v)$ induced by station capacities can be separated by complete enumeration. We denote the arc sets corresponding to all maximal cliques in Λ by Γ_{max} and receive:

$$
\begin{array}{llrll}
(\mathrm{APP}) & & & & \\
\max & \sum_{i\in I}\sum_{a\in A_i} w_a x_a & & & \text{(i)}\\
s.t. & \sum_{a\in\delta_{\text{out}}(s_i)} x_a & \le 1, & \forall i\in I & \text{(ii)}\\
 & \sum_{a\in\delta_{\text{in}}(t_i)} x_a & \le 1, & \forall i\in I & \text{(iii)}\\
 & \sum_{a\in\delta_{\text{out}}(v)} x_a - \sum_{a\in\delta_{\text{in}}(v)} x_a & = 0, & \forall v\in V_i\setminus\{s_i,t_i\}, i\in I & \text{(iv)}\\
 & \sum_{a\in A_\gamma} x_a & \le \kappa_\gamma, & \forall \gamma\in\Gamma_{max} & \text{(v)}\\
 & x_a & \in\{0,1\}, & \forall a\in A_i, i\in I & \text{(vi)}
\end{array}
$$

Note that constraints (APP) (iii) are again redundant. The packing model can also be formulated with binary decision variables x_p for each path instead of arc variables x_a. Consequently, we define the proceedings of a path p as the sum of its incident arcs:

$$
w_p = \sum_{a\in p} w_a.
$$

The resulting version (PPP) reads as follows:

$$
\begin{array}{llrll}
(\mathrm{PPP}) & & & & \\
\max & \sum_{i\in I}\sum_{p\in P_i} w_p x_p & & & \text{(i)}\\
s.t. & \sum_{p\in P_i} x_p & \le 1, & \forall i\in I & \text{(ii)}\\
 & \sum_{p\cap A_\gamma\neq\emptyset} x_p & \le \kappa_\gamma, & \forall \gamma\in\Gamma_{max} & \text{(iii)}\\
 & x_p & \in\{0,1\}, & \forall p\in\mathcal{P}_i, i\in I & \text{(iv)}
\end{array}
$$

Constraints (PPP) (ii) ensure that each request is implemented by at most one path. Conflict constraints (PPP) (iii) make sure that no headway or station conflict is violated. (PPP) (iv) state that all path variables x_p are zero or one. Finally, objective (PPP) (i) is to maximize the profit of the schedule.

formulation	variables	non-trivial constraints
(APP)	$O(A)$	$O(A)$
(PPP)	$O(P)$	$O(V)$
(APP)$'$	$O(A)$	$O(A^2)$
(PPP)$'$	$O(P)$	$O(A^2)$

Table 2: Sizes of packing formulation for the track allocation problem with block occupation

The packing formulations of the optimal track allocation problem with block occupation conflicts only have the sizes listed in Table 2. For a set S we write $O(S) = O(|S|)$. Model (PPP)$'$ is thereby a path formulation based on pairwise headway conflict sets.

We have seen for the block occupation case that the number of maximal conflicting sets can be bounded by the number of nodes and can be efficiently constructed. Unfortunately, in the general case, which are models (APP)$'$ and (PPP)$'$, it might lead to conflicts sets quadratically in the number of running arcs.

2.2 Coupling Models

We propose in this section an alternative formulation for the optimal track allocation problem that guarantees a conflict free routing by allowing only feasible route combinations, and not by excluding conflicting ones as described in Section 2.1. The formulation is based on the concept of feasible arc *configurations*, i.e., sets of arcs on a track without headway conflicts. Formally, we define a configuration for some track $j = (x, y) \in J$ as a set of arcs $q \subseteq A_j := \{(u, v) \in A_{\mathrm{I}} : s(u)s(v) = (x, y) \text{or } j((x, y)) = j\}$ such that

$$|q \cap A_\gamma| \leq 1 \quad \forall \gamma \in \Gamma.$$

Denote by $\mathcal{Q}_j$ the set of all such configurations for track $j \in J$, and by $\mathcal{Q}$ the set of all configurations over all tracks. The idea of the extended model is to introduce 0/1 variables y_q for choosing a configuration on each track and to force a conflict free routing of train paths $p \in \mathcal{P}$ through these configurations by means of inequalities

$$\sum_{p \in \mathcal{P}, a \in p} x_p \leq \sum_{q \in \mathcal{Q}, a \in q} y_q \quad \forall a \in A_{\mathtt{I}}.$$

In Section 2.3 we will prove that this is equivalent to the packing constraints (APP) (v) and (PPP) (iii) in case of headway conflicts. In the following, we will show that these feasible time-arc configurations or sequences for each track $j \in J$ can be constructed very efficiently under several reasonable assumptions.

In a first step, we introduce a headway conflict equivalence class for each running arc $a \in A_{\mathtt{I}}$, if their resource consumption on a track is equal. The reason is that many time-arcs share the same headway restrictions, i.e., the next potential departure times are equal, even if other attributes might be different (objective, train type, request, mode, etc.).

Definition 2.4. *Two arcs $a = (x, y)$ and $b = (w, z)$ with $a, b \in A_I$ are resource equivalent, i.e., $a \sim b$, if*

- ▷ $j(a) = j(b)$, *(same track)*
- ▷ $\tau(x) = \tau(w)$, *(same departure time)*
- ▷ $\tau(y) = \tau(z)$, *(same arrival time)*
- ▷ *and,* $h_{j(a),c(a),m(a),k,l} = h_{j(b),c(b),m(b),k,l}, \forall k \in C, l \in M$ *(same headway time for any succeeding train type and mode).*

Obviously, the relation defined by 2.4 is reflexive, symmetric and transitive, and thus a equivalence relation. In Figure 9 running arcs of two requests on track (X, Y) can be seen. Assume that they fulfill additionally the Definition 2.4, then a hyperarc represents the corresponding equivalence class.

Denote by A_j^{Ψ} the set of all equivalence classes on track $j \in J$ and $A^{\Psi} = \bigcup_{j \in J} A_j^{\Psi}$ of all running arcs $A_{\mathtt{I}}$, respectively. Due to the headway definition, i.e., all minimal headway times are strictly positive, only one arc of each class can be chosen. However, it does not matter which one. The idea is to define local feasible flows, which ensure headway feasibility on each track and couple them appropriately with the train or route flows. Even if this trivial observation might complicate the notation, it is a crucial and necessary point to aggregate and strengthen the models. Otherwise, this would lead to too many and foremost weaker constraints. Instead of directly writing down a corresponding model, however, we propose a version that will model configurations

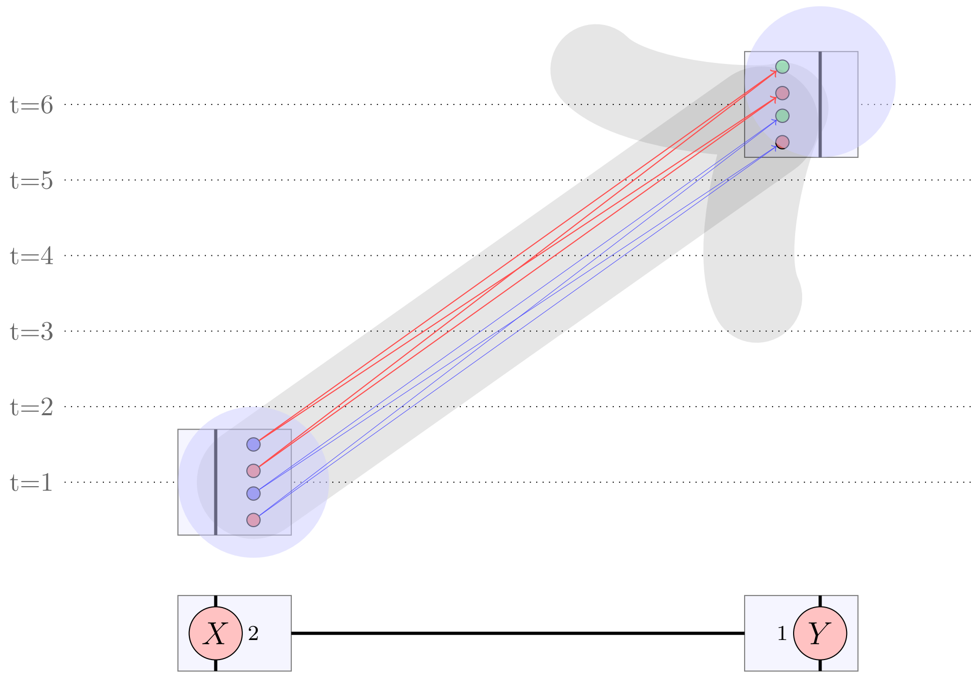


Figure 9: Example for an equivalence class and a hyperarc.

as paths in a certain acyclic routing digraph, if the headway matrix is valid. The advantages of such a formulation will become clear in the following. The construction extends the already described routing digraph $D = (V, A)$ to a larger digraph as illustrated in Figure 10. We will denote the extended digraph by $\overline{D} = (V \cup \overline{V}, A \cup A^{\Psi} \cup \overline{A})$.

The construction is as follows: Let s_j be an artificial source and t_j an artificial sink node to define a flow on track $j = (x, y)$. Consider the running arc classes A_j^{Ψ} on track j. Denote by $L_j := \{u : (u, v) \in A_j^{\Psi}\}$ and $R_j := \{v : (u, v) \in A_j^{\Psi}\}$ the associated sets of event nodes at the start and end station of track j. Note that all arcs in A_j^{Ψ} go from L_j to R_j. We denote by $n(\tau_1, c_1, m_1, c_2, m_2) \in \mathbb{Z}$ for $v = (-, c_1, m_1, \tau_1) \in R_j$ the next possible departure time of a train of type $c_2 \in C$ and $m_2 \in M$ after a train $c_1 \in C$ has departed with mode $m_1 \in M$ at τ_1. Now let $\overline{A}_j := \{(v, u) : v \in R_j, u \in L_j\}$ be a set of "return" arcs that go back in the opposite direction and represent the next potential departure on that track; they connect the end of a running arc on j (or node s_j) with all possible follow-on arcs (or node t_j) on that:

$$n(\tau_1, c_1, m_1, c_2, m_2) = \tau_1 - d_{j,c_1,m_1} + h_{j,c_1,m_1,c_2,m_2}, \tag{1}$$

$$(v, u) \in \overline{A}_j \Leftrightarrow \tau(u) \geq n(\tau_1, c_1, m_1, c_2, m_2). \tag{2}$$

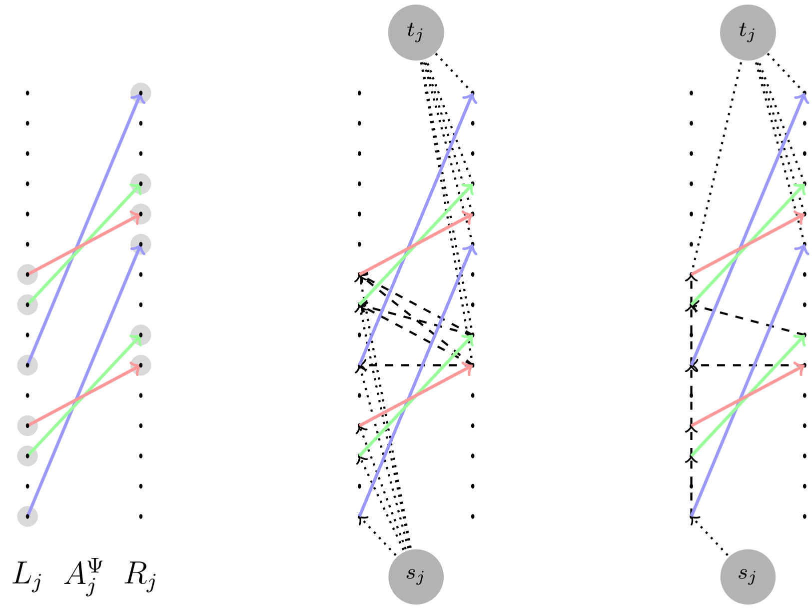


Figure 10: Example for the construction of a track digraph.

It is easy to see that the *configuration routing digraph* $D_j := (L_j \cup R_j \cup \{s_j, t_j\}, A_j^\Psi \cup \overline{A}_j)$ is bipartite and acyclic, if all minimal headway times are strictly positive.

In Figure 10, the construction is shown on a small set A_j^Ψ. On the left, the set of arcs (one per equivalence class) of track $j \in J$ and the node sets L_j and R_j can be seen. In the middle, the constructed graph D_j is shown with dashed and dotted auxiliary arcs for the easy case of full block occupation.

The graph size can be significantly reduced by merging structural nodes and introducing a time-line. In the trivial case of full block occupation the next possible train departure on track j is independent of the preceding and succeeding train type or running mode, i.e., the formula 2 simplifies to:

$$n(\tau_1, c_1, m_1, c_2, m_2) = \tau_1 - d_{c_1,m_1} + h_{j,c_1,m_1,c_2,m_2} = \tau_1.$$

Since $n(\tau_1, c_1, m_1, c_2, m_2)$ is exactly the arrival time of the considered running train on track j, we can merge nodes of set L_j and R_j, if their times match. Therefore, we connect consecutive departure nodes of L_j, i.e., s_j with the first one and the last one with t_j, respectively. Instead of constructing all possible return arcs each arrival node in R_j is only connected once with the time-line, i.e., with the next potential departure node L_j (or t_j). On the right side of Figure 10 this reduced graph based on a time-line concept can be seen. The precise time-line construction and corresponding mathematical formulas can be found in Borndörfer & Schlechte (2007) [30].

Hence $s_j t_j$-paths $\overline{a}_1, a_1, \ldots, \overline{a}_k, a_k, \overline{a}_{k+1}$ in D_j (without time-lines) and configurations $\{a_1, \ldots, a_k\}$ in $\mathcal{Q}_j$ are in one-to-one correspondence for the case of block occupation. Let us formally denote this isomorphism by a mapping

$$\bar{\cdot} : \mathcal{Q}_j \to \mathcal{P}_j, q \mapsto p, j \in J,$$

where $\mathcal{P}_j$ denotes the set of all $s_j t_j$-paths in $D_j = (V_j, A_j)$; however, we will henceforth identify paths $p \in \mathcal{P}_j$ and configurations $q \in \mathcal{Q}_j$.

In the following, we will discuss the construction for the general headway case. It is easy to see that the construction rule (2) can again be applied to ensure consecutive valid headway times. However, Figure 11 gives an example, what can happen if H^j is not transitive. On the left, three running arcs on track j and in the middle the constructed track digraph $D = (V, A)_j$ with respect to H^j are shown. Note that arc k and l as well as l and m are feasible successor, but k and m are not connected due to non-transitivity of H^j. On the right a $s_j t_j$-path in $\overline{D}_j$ is highlighted, which violates a minimum headway time of trains which are not direct successors.

Therefore, transitivity of H is a necessary condition to allow for an exact construction via D_j. Otherwise $\overline{D}_{(x,y)}$ defines only a relaxation of the configuration $\mathcal{Q}_j$, because there are $s_{(x,y)} t_{(x,y)}$-paths which could violate non-consecutively headway times.

Lemma 2.5. *There is a bijection from all $s_j t_j$-paths in D_j to the set of valid configurations $\mathcal{Q}_j$ on track $j \in J$ if the headway matrix H is transitive.*

Proof. We provide two variants of the proof to facilitate the understanding. Let D_j be the track digraph induced by headway matrix H.

1. assume H is transitive, then the following map $\bar{\cdot}$ is a bijection:

$$\bar{\cdot} : \mathcal{P}_j \to \mathcal{Q}_j, p = \overline{a}_1, a_1, \ldots, \overline{a}_k, a_k, \overline{a}_{k+1} \mapsto q = \{a_1, \ldots, a_k\}, j \in J,$$

2. or assume H is not transitive, then we can construct a path $p \in D_j$, which is not a valid configuration, see Figure 11. In that case

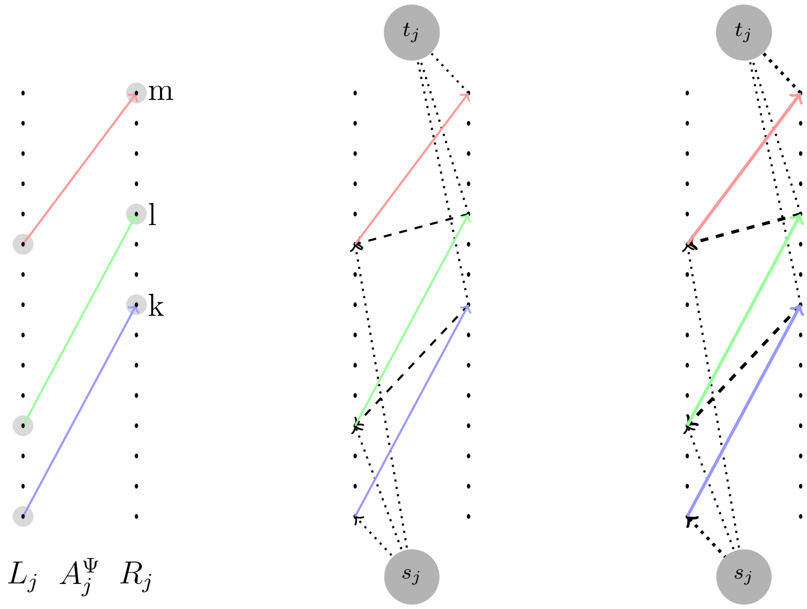


Figure 11: Example for a path which does not correspond to a valid configuration if the headway times violate the transitivity.

no bijection can exist between these spaces of different dimension, which is a contradiction.

□

Remark 2.6. *The idea of reducing the huge number of potential return arcs by a time-line in D_j can be transfered. We only have to distinguish between the basic equivalence classes induced by Definition 2.4, i.e., independent of the times τ. In the worst case, these are $C \times M$ departure time-lines, one for each train type c and running mode m. We do not give a precise formulation for this construction. However, in our software module* `TS-OPT` *a timeline concept, that is based on the equivalence classes, is implemented.*

Remark 2.7. *In Section 2 we have introduced an algorithm that provides a macroscopic network with transitive headway matrices on all tracks. Technical minimal headway times are naturally transitive for real world data.*

Henceforth, we have defined all objects to introduce an extended formulation of the TTP. Variables x_a, $a \in A_i$, $i \in I$ control again the use of arc a in D_i and y_b, $b \in A_j^\Psi \cup \overline{A}_j$, $j \in J$ in D_j, respectively.

$$
\begin{array}{llrll}
\text{(ACP)} & & & & \\
\max & \sum\limits_{a\in A} w_a x_a & & & \text{(i)} \\
s.t. & \sum\limits_{a\in\delta^i_{\text{out}}(v)} x_a - \sum\limits_{a\in\delta^i_{\text{in}}(v)} x_a & = 0, & \forall\, i \in I, v \in V_i \backslash \{s_i, t_i\} & \text{(ii)} \\
& \sum\limits_{a\in\delta^i_{\text{out}}(s_i)} x_a & \leq 1, & \forall\, i \in I & \text{(iii)} \\
& \sum\limits_{a\in\delta^i_{\text{out}}(v)} y_a - \sum\limits_{a\in\delta^i_{\text{in}}(v)} y_a & = 0, & \forall\, j \in J, v \in V_j \backslash \{s_j, t_j\} & \text{(iv)} \\
& \sum\limits_{a\in\delta^i_{\text{out}}(s_j)} y_a & \leq 1, & \forall\, j \in J & \text{(v)} \\
& \sum\limits_{a\in b} x_a - y_b & = 0, & \forall\, b \in A^\Psi & \text{(vi)} \\
& x_a, y_b & \in \{0,1\}, & \forall\, a \in A, b \in A^\Psi \cup \overline{A}_j. & \text{(vii)}
\end{array}
$$

The objective, denoted in (ACP) (i), is to maximize the weight of the track allocation. Equalities (ii) and (iv) are well-known *flow conservation constraints* at intermediate nodes for all trains flows $i \in I$ and for all flows on tracks $j \in J$, (iii) and (v) state that at most one flow, i.e., train and track, unit is realized. Equalities (vi) link arcs used by train routes and track configurations to ensure a conflict-free allocation on each track individually, i.e., the hyperarcs $b \in A^\Psi$ are coupled with the arc set A_I. Finally, (vii) states that all variables are binary.

Remark 2.8. *Note that conflict constraints induced by station capacities are not considered in that construction. In the work of Erol (2009) [84] the configuration idea was also applied to these kind of constraints. Actually, we prefer a "lazy" approach to add them only if needed. Even though they do not arise naturally. In fact, only the aggregation of tracks inside and in the area around a station leads to them.*

Remark 2.9. *Conflict constraints induced by single way usage of two opposing tracks can be easily considered in that construction, as well. The main difference is the definition of the return arcs, which decide what a valid successor after each running arc is. In that case they can be adjacent to both stations of the track because the next departure can either be in the same or in the opposing direction on track j. Consequently, we have departure time-lines on both sides of the track. Due to the properties of headway times for single way tracks the resulting graph D_j remains acyclic. Note that a minimal technical headway time for the opposing direction must be larger than the running time of the preceding train, see formula 1 in Section 2.3.*

Pure static approaches and models are handicapped due to memory limitations. The presented digraphs and thus the model formulation can easily become very large and exceed 8GB of main memory even for instances with some hundred trains. Explicit numbers are given in Chapter IV and Section 1. To overcome these restrictions dynamic approaches to create and solve these models are very efficient and successful. We already presented the idea of column generation and branch and price in Section 8.5. To apply these techniques we developed a path based formulation of the (ACP), called (PCP), which will be the topic of Section 3. The path coupling model (PCP) is formulated with binary decision variables x_p for each path instead of arc variables x_a and y_q for each configuration ("path") instead of arc variables y_b as follows:

$$
\begin{array}{lllllll}
\text{(PCP)} & & & & & & \\
\max & \sum\limits_{p\in\mathcal{P}} w_p x_p & & & & & \text{(i)} \\
s.t. & \sum\limits_{p\in\mathcal{P}_i} x_p & & & \leq 1, & \forall i \in I & \text{(ii)} \\
& & & \sum\limits_{q\in\mathcal{Q}_j} y_q & \leq 1, & \forall j \in J & \text{(iii)} \\
& \sum\limits_{p\in\mathcal{P}, b\in p} x_p & - & \sum\limits_{q\in\mathcal{Q}, b\in q} y_q & \leq 0, & \forall b \in A^{\Psi} & \text{(iv)} \\
& & & y_q & \in \{0,1\}, & \forall q \in \mathcal{Q} & \text{(v)} \\
& x_p & & & \in \{0,1\}, & \forall p \in \mathcal{P}. & \text{(vi)}
\end{array}
$$

The objective, denoted in (PCP) (i), is to maximize the weight of the track allocation. Inequalities (ii) and (iii) are set packing constraints to ensure that for each request $i \in I$ and each track $j \in J$ at most one path or configuration is chosen. Inequalities (iv) link arcs used by train routes and track configurations to ensure a conflict-free allocation on each track individually. We say that $b \in A^{\Psi}$ is an element of path p, $b \in p$, if there is an arc $a \in p$ with $a \in b$. Finally, (v) and (vi) state that all variables are binary.

Let $\gamma \in \mathbb{R}^{|I|}, \pi \in \mathbb{R}^{|J|}$ and $\lambda \in \mathbb{R}^{|A^{\Psi}|}$ be dual vectors. Consider the linear program arising from (PCP) (i) to (iv) with $y_q \geq 0, q \in \mathcal{Q}$ and $x_p \geq 0, p \in \mathcal{P}$. Because of (PCP) (ii) and (iii) the upper bound constraints $y_q \leq 1$ and $x_p \leq 1$ are redundant, and therefore we can ignore them for the dualization. We get the following dual problem:

$$
\begin{array}{llllr}
\text{(DLP)} & & & & \\
\min & \sum_{j\in J}\pi_j + \sum_{i\in I}\gamma_i & & & \text{(i)}\\
s.t. & \gamma_i + \sum_{a\in p, b\ni a}\lambda_b & \geq w_p & \forall p\in\mathcal{P}_i, \forall i\in I & \text{(ii)}\\
 & \pi_j - \sum_{b\in q}\lambda_b & \geq 0 & \forall q\in\mathcal{Q}_j, \forall j\in J & \text{(iii)}\\
 & \gamma_i & \geq 0 & \forall i\in I & \text{(iv)}\\
 & \pi_j & \geq 0 & \forall j\in J & \text{(v)}\\
 & \lambda_b & \geq 0 & \forall b\in A^{\Psi} & \text{(vi)}
\end{array}
$$

Furthermore we receive the corresponding pricing problem for the x-variables:

$$
\text{(PRICE}(x)) \quad \exists\, i\in I, p\in\mathcal{P}_i: \quad \sum_{a\in p} w_a - \sum_{a\in p, b\ni a}\lambda_b - \gamma_i > 0.
$$

Remember that each arc $a\in A_{\mathrm{I}}$ is exactly coupled with one resource-equivalent hyperarc $b\in A^{\Psi}$, denoted by $b(a)$. Solving this pricing problem is equivalent to answer the question, whether there exists a request $i\in I$ and a path $p\in\mathcal{P}_i$ with positive reduced cost. Due to the fact that all D_i are acyclic, this problem decomposes into $|I|$- longest path problems with arc lengths $l_a = w_a - \lambda_{b(a)}$, if $a\in A_I$ and $l_a = w_a$ otherwise. For the y-variables we get:

$$
\text{(PRICE}(y)) \quad \exists\, j\in J, q\in\mathcal{Q}_j: \quad \sum_{b\in q}\lambda_b - \pi_j > 0
$$

Analogously, the pricing problem for the y- variables decomposes into $|J|$- easy longest path problems, one for each acyclic digraph D_j. The pricing of configurations $\mathcal{Q}_j$ is equivalent to find a shortest s_jt_j-path in D_j using arc lengths $l_b = \lambda_b$, $b\in A^{\Psi}$ and 0 otherwise. Since D_j is acyclic, this is polynomial. By the polynomial equivalence of separation and optimization, see Grötschel, Lovász & Schrijver (1988) [104], here applied to the (DLP), we obtain:

Lemma 2.10. *The linear relaxation of* (PCP) *can be solved in polynomial time.*

Let us state in this pricing context a simple bound on the LP-value of the path configuration formulation (PCP). We set $b(a) = \emptyset$ for

$a \notin A_{\mathrm{I}}$ to simplify notation. In practical implementations, this bound might be utilize to detect tailing-off effects in a column generation procedure, i.e, one can stop the column generation with a certain optimality gap at the root node and start so-called "early branching". Namely, computing the path lengths $\max_{p\in\mathcal{P}_i} \sum_{a\in p} w_a - \sum_{a\in p, b\ni a} \lambda_b$ for all $i \in I$ and $\max_{q\in\mathcal{Q}_j} \sum_{b\in q} \lambda_b$ for all $j \in J$ yields the following LP-bound $\beta = \beta(\gamma, \pi, \lambda)$.

Lemma 2.11. *Let $\gamma, \pi, \lambda \geq 0$ be dual variables*[6] *for* (PCP) *and* $v_{\mathrm{LP}}(\mathrm{PCP})$ *the optimum objective value of the LP-relaxation of* (PCP)*. Define*

$$\begin{aligned}
\eta_i &:= \max_{p\in\mathcal{P}_i} \sum_{a\in p}(w_a - \sum_{a\in p, b\ni a} \lambda_b) - \gamma_i, && \forall i \in I,\\
\theta_j &:= \max_{q\in\mathcal{Q}_j} \sum_{b\in q} \lambda_b - \pi_q, && \forall j \in J,\\
\beta(\gamma,\pi,\lambda) &:= \sum_{i\in I} \max\{\gamma_i + \eta_i, 0\} + \sum_{j\in J} \max\{\pi_j + \theta_j, 0\}.
\end{aligned}$$

Then

$$v_{\mathrm{LP}}(\mathrm{PCP}) \leq \beta(\gamma, \pi, \lambda).$$

Proof. Assuming the pricing problems are solved to optimality, we have:

- ▷ $\gamma_i + \eta_i \geq \sum_{a\in p}(w_a - \sum_{a\in p, b\ni a} \lambda_b) \Rightarrow \gamma_i + \eta_i + \sum_{a\in p, b\ni a} \lambda_b) \geq w_p \quad \forall i \in I, p \in \mathcal{P}_I$.
- ▷ $\pi_j + \theta_j \geq \sum_{b\in q} \lambda_b \Rightarrow \pi_j + \theta_j - \sum_{b\in q} \lambda_b \geq 0 \quad \forall j \in J, q \in \mathcal{Q}_j$.
- ▷ $(\max\{\gamma+\eta, 0\}, \max\{\pi+\theta, 0\}, \lambda)$ (the maximum taken componentwise) is dual feasible for the LP-relaxation of (PCP).

□

Remark 2.12. *Note that this is true in general for all column generation approaches where the pricing is solved exactly. If the pricing problem could not be solved to optimality then solving a relaxation of the pricing problem can also provide a global bound. We analysed this approach for the multiple resource constraint shortest path problem by using enhanced linear relaxations, see Schlechte (2003) [187] and Weider (2007) [213].*

[6] Note that these will be global infeasible during a column generation.

2.3 Polyhedral Analysis

In this section, we show that (PCP) and (ACP) are extended formulations of (PPP) and (APP), respectively. Furthermore some basic polyhedral observations are presented using the standard notation and definitions that can be found in Ziegler (1995) [219]. Starting points are the LP-relaxations of the configuration formulations and those of the packing formulations. As the LP-relaxations of (APP) and (PPP), and of (ACP) and (PCP) are obviously equivalent via flow decomposition into paths, it suffices to compare, say, (APP) and (ACP). Furthermore, we consider models (APP) based on the simple case of block occupation conflicts only. The case of general headway conflicts would only unnecessary complicate the notation. However, in case of station capacity conflicts a more general definition of "configurations" and hence different models are needed, i.e., see Erol (2009) [84]. Let us shortly list the needed sets:

- ▷ A set of all "standard" time-arcs representing train operations,
- ▷ A_{I} set of time-arcs representing track usage,
- ▷ A^{Ψ} set of resource equivalence classes representing track usage,
- ▷ V_j set of time-nodes of track digraph induced by track j,
- ▷ Γ_j subset of conflict set induced by track j,
- ▷ and, $\overline{A} = \cup_{j \in J} \overline{A}_j$ set of all "auxiliary" time-arcs representing the consecutive succession of arcs on track j.

Lemma 2.13. *Let*

$$P_{\mathrm{LP}}(\mathrm{APP}) := \{x \in \mathbb{R}^A : (\mathrm{APP})\ (\mathrm{ii})\text{–}(\mathrm{v})\}$$
$$P_{\mathrm{LP}}(\mathrm{ACP}) := \{(x,y) \in \mathbb{R}^{A \times A^{\Psi} \times \overline{A}} : (\mathrm{ACP})\ (\mathrm{ii})\text{–}(\mathrm{vi})\}$$
$$\pi_x : \mathbb{R}^{A \times A^{\Psi} \times \overline{A}} \to \mathbb{R}^A, \quad (x,y) \mapsto x$$

be the polyhedron associated with the LP-relaxations of (APP) *and* (ACP)*, respectively, and a mapping that produces a projection onto the coordinates of the train routing variables. Then*

$$\pi(P_{\mathrm{LP}}(\mathrm{ACP})) = P_{\mathrm{LP}}(\mathrm{APP}).$$

Proof. Let $\Gamma_j := \{\gamma \in \Gamma : \gamma \subseteq A_j\}$, $j \in J$, be the set of block conflict cliques associated with track j. Consider the polyhedron

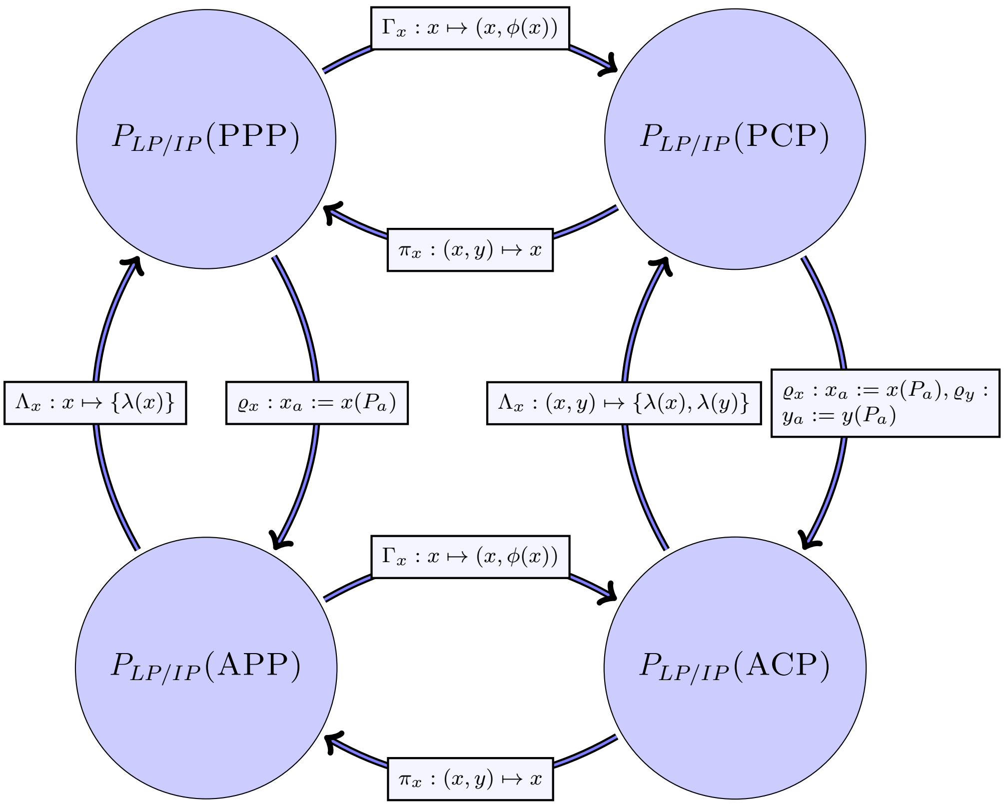


Figure 12: Relations between the polyhedra of the different models.

$$
\begin{aligned}
P &:= \{x \in \mathbb{R}^A && : \text{(APP) (ii), (iii), (v)}\},\\
P^j &:= \{x \in \mathbb{R}_+^{A_j} && : \sum_{a \in \gamma} x_a \leq 1 \quad \forall \gamma \in \Gamma_j\}, \quad j \in J,\\
Q^j &:= \{y \in \mathbb{R}_+^{A_j^\Psi \times \overline{A}_j} && : \sum_{a \in \delta_j^+(v)} y_a = \sum_{a \in \delta_j^-(v)} y_a, \forall v \in V_j \backslash \{s_j, t_j\},\\
& && \quad \sum_{a \in \delta_j^+(s_j)} y_a \leq 1\}, j \in J,\\
R^j &:= \{x \in \mathbb{R}_+^{A_j} && : \exists y \in Q^j : x \leq y\}, \quad j \in J.
\end{aligned}
$$

P^j is integer, because Γ_j is the family of all maximal cliques of an interval graph, which is perfect; Q_j is integer, because it is the path polytope associated with an acyclic digraph; finally, R^j is integer, because it is the anti-dominant of an integer polytope. Consider integer points, it is easy to see that P^j and R^j coincide, i.e., $P^j = R^j$, $j \in J$. It follows

$$P_{\mathrm{LP}}(\mathrm{APP}) = P \cap \bigcap_{j\in J} P^j = P \cap \bigcap_{j\in J} R^j = \pi(P_{\mathrm{LP}}(\mathrm{ACP})).$$

□

This immediately implies our main Theorem.

Theorem 2.14. *Denote by $v(P)$ and $v_{\mathrm{LP}}(P)$ the optimal value of problem P and its LP-relaxation, respectively, with $P \in \{(\mathrm{APP})', (\mathrm{APP}), (\mathrm{PPP}), (\mathrm{ACP}), (\mathrm{PCP})\}$. Then:*

- ▷ $v_{\mathrm{LP}}(\mathrm{APP})' \geq v_{\mathrm{LP}}(\mathrm{APP})$.
- ▷ $v_{\mathrm{LP}}(\mathrm{APP}) = v_{\mathrm{LP}}(\mathrm{PPP}) = v_{\mathrm{LP}}(\mathrm{ACP}) = v_{\mathrm{LP}}(\mathrm{PCP})$.
- ▷ $v(\mathrm{APP})' = v(\mathrm{APP}) = v(\mathrm{PPP}) = v(\mathrm{ACP}) = v(\mathrm{PCP})$.

Figure 12 illustrates the transformation between the different models. The given projections show that coupling models are extended formulations of the original packing ones. More details on extended formulations and projections of integer programming formulations can be found in Balas (2005) [16]. The idea of extended formulations is shown in Figure 13. On the left hand side, the rough structure of the packing formulation (PPP) can be seen, i.e., with appropriate binary matrices A and R. On the right hand side the structure of model (PCP) after the transformation of the packing constraints associated with matrix R is shown. Matrix B denotes the auxiliary configuration partitioning part and C and D the necessary coupling part.

Lemma 2.15. $P_{LP}(\mathrm{PCP}) := \{x \in \mathbb{R}^{\mathcal{P}\cup\mathcal{Q}} : (\mathrm{PCP})\ (\mathrm{ii})–(\mathrm{iv})\}$ *is full-dimensional.*

Proof. To show that $P_{\mathrm{LP}}(\mathrm{PCP})$ is full-dimensional, we have to construct $|\mathcal{P}|+|\mathcal{Q}|+1$ affinely independent and feasible points in $P_{LP}(\mathrm{PCP})$. For each path $p \in \mathcal{P}$ ($q \in \mathcal{Q}$), we denote the set of arcs incident to p (q) and contained in A^{Ψ} by A_p (A_q). The set of all coupling hyper-arcs is again denoted by A^{Ψ}.

First, consider for each $p \in \mathcal{P}$ the associated path-configuration incidence vector $\phi(p) \in \{0,1\}^{\mathcal{P}}, \nu(p) \in \{0,1\}^{\mathcal{Q}}$ with $k \in \mathcal{P}$ and $l \in \mathcal{Q}$, constructed as follows:

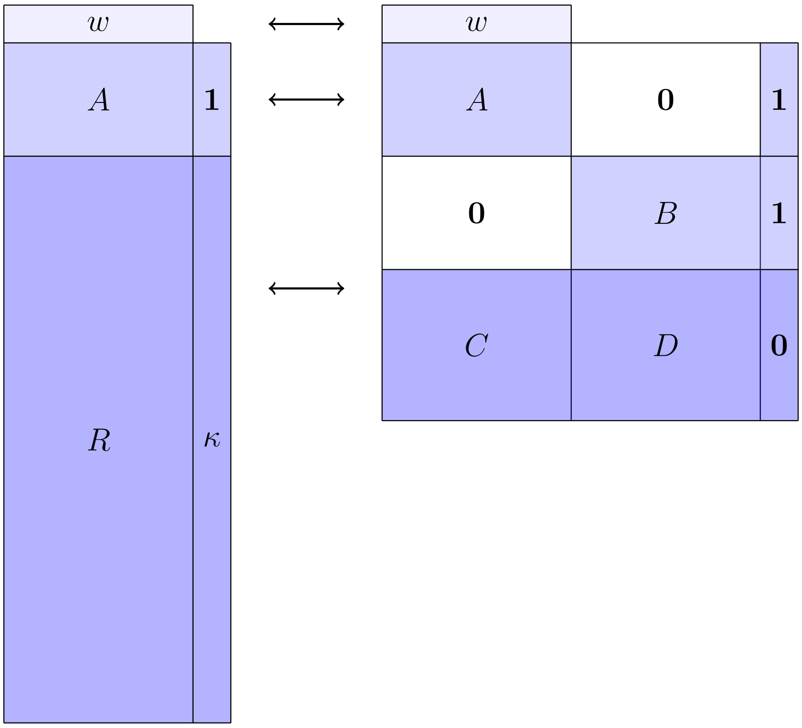


Figure 13: Idea of the extended formulation (PCP) for (PPP).

$$\phi_k(p) = \begin{cases} 1 & \text{if } k = p, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

$$\nu_l(p) = \begin{cases} 1 & \text{if } A_l = \{b \subseteq A^\Psi \,:\, b = b(a), \forall a \in A_p, j(a) = j(l)\} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The entries $\nu_l(p)$ "activate" exactly the minimum configuration on track $j(l)$ "consumed" by path p, i.e., only the arcs $b \in A_\Psi, b(a) \in p \cap A_{\mathrm{I}}$ are used in configuration l. Request and track packing constraints are trivially fulfilled, because we only chose one path to be at one and because at most one configuration is used by path p for each track j. The coupling constraints are fulfilled for all $b \in A^\Psi$ by the definition of $\nu(p)$, since p is a feasible path. Thus, $(\phi(p), \nu(p))$ is obviously contained in $P_{IP}(\text{PCP})$ for all $p \in \mathcal{P}$. Next, consider for each "configuration" $q \in \mathcal{Q}$ the qth unit vector $(\phi(q), \nu(q))$.

We have constructed $|\mathcal{P} + \mathcal{Q}|$ many vectors which form the matrix

$$\begin{pmatrix} \phi(p) & \phi(q) \\ \nu(p) & \nu(q) \end{pmatrix} = \begin{pmatrix} E_{|\mathcal{P}|} & \mathbf{0} \\ \nu(p) & E_{|\mathcal{Q}|} \end{pmatrix},$$

where E_n denotes the n-dimensional identity matrix.

These vectors are linearly independent due to the fact that the determinant of this lower triangular matrix is obviously 1. Together with the feasible vector $\mathbf{0} \in \mathbb{R}^{|\mathcal{P}+\mathcal{Q}|}$, we have constructed $|\mathcal{P}|+|\mathcal{Q}|+1$ affinely independent points of $P_{LP}(\text{PCP})$, proving our proposition. $\square$

Lemma 2.16. *Constraint* (PCP) (iii) *associated with track* $j \in J$ *defines a facets of* $P_{IP}(\text{PCP})$, *if* $\mathcal{Q}_j \neq \emptyset$.

Proof. We have to show that the hyperplane $H_j = \{(\phi, \nu) \in [0,1]^{|\mathcal{P}|+|\mathcal{Q}|} : \sum_{q \in \mathcal{Q}_j} y_q = 1\}$ contains $|\mathcal{P}| + |\mathcal{Q}|$ affinely independent points of the polyhedron $P_{LP}(\text{PCP})$.

First, for each $p \in \mathcal{P}$ we construct a vector $(\tilde{\phi}(p), \tilde{\nu}(p))$ based on the vector $(\phi(p), \nu(p))$ as follows. If path p contains at least one coupling arc of track j, then define vector $(\tilde{\phi}(p), \tilde{\nu}(p)) = (\phi(p), \nu(p))$, and otherwise let $(\tilde{\phi}(p), \tilde{\nu}(p)) = (\phi(p), \nu(p)) + (\mathbf{0}, e_{\overline{q}_j})$, where $(\phi(p), \nu(p))$ is the vector from formula 3 and 4 and $e_{\overline{q}_j}$ is the $\overline{q}_j$th unit vector for some configuration $\overline{q}_j \in \mathcal{Q}_j$.

Obviously, $(\tilde{\phi}(p), \tilde{\nu}(p))$ is feasible and satisfies packing constraints (PCP) (iii) associated with track j with equality.

Next, for each "configuration path" $q \in \mathcal{Q}_j$ we define $(\tilde{\phi}(q), \tilde{\nu}(q)) = (\mathbf{0}, e_q)$ with e_q as the qth unit vector, and otherwise (if $q \in \mathcal{Q} \setminus \mathcal{Q}_j$) let $(\tilde{\phi}(q), \tilde{\nu}(q))$ be the sum of the $(\mathbf{0}, e_q)$ and $(\mathbf{0}, e_{\overline{q}_j})$. Hence, $(\tilde{\phi}(q), \tilde{\nu}(q))$ is a feasible point of $P_{IP}(\text{PCP})$ and H_j.

Finally, we have constructed $|\mathcal{P}|+|\mathcal{Q}|$ many vectors which are contained in H_j and $P_{IP}(\text{PCP})$. Re-sorting the vectors in an appropriate way we obtain a lower-triangular matrix such that the last row and column corresponds to configuration $\overline{q}_j$, then we get:

$$\begin{pmatrix} \tilde{\phi}(p) & \tilde{\phi}(q) \\ \tilde{\nu}(p) & \tilde{\nu}(q) \end{pmatrix} = \begin{pmatrix} E_{|\mathcal{P}|} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \cdots & E_{|\mathcal{Q} \setminus \mathcal{Q}_j|} & \mathbf{0} & \mathbf{0} \\ \cdots & \mathbf{0} & E_{|\mathcal{Q}_j \setminus \overline{q}_j|} & \mathbf{0} \\ \cdots & \mathbf{1} & \mathbf{0} & \mathbf{1} \end{pmatrix}.$$

Since the determinant of this matrix is one, the vectors are linearly independent proving that $H_j \cap P_{IP}(\text{PCP})$ is a facet.

□

Remark 2.17. *The analysis of the packing constraints* (PCP) (ii) *and the coupling constraints* (PCP) (iv) *remains as an open problem. It is not trivially clear in which cases these constraints are facet defining or not. Even if this is more a theoretical research question, we believe that deep polyhedral insights can support the algorithmic solution approach. Hence, we hope that in the future these questions might be answered.*

We want to point out that this is not only a basic theoretical analysis of the model. The main motivation was to find out whether there is a structural reason why the coupling models perform better than there counterparts. Even if we can only provide some theoretical answer for that we believe that this an interesting topic for future research. To answer the question in which cases coupling constraints are facets might be useful in designing and further development of solution algorithms.

2.4 Extensions of the Models

In the last section we analyzed in detail the track allocation problem with respect to "hard" combinatorial constraints. In this part we want to discuss how to handle global combinatorial requirements on the set of train request and rather "soft" constraints on the implicit buffer times.

Manifold reasons cause combinatorial interaction between train slots. Our definitions are based on the bidding language of an auction design introduced in Borndörfer et al. (2006) [34], therefore we use synonymously bid and train slot request. Three potential sources for combinatorial bids are mentioned, tours to support rolling stock planning, regular service intentions to allow for attractive offers for the passengers, and operator neutral connections to establish reliable and fast interlining connections.

Another extension is based on the potential of the extended formulation to control the implicit buffer times on each track. We exploit this structural advantage by introducing a robustness measure on the "return" arcs, and developed a straight-forward bi-criteria model in Schlechte & Borndörfer (2008) [188]. This allows for evaluating the trade-off between efficiency, i.e., the utilization of the macroscopic network, and the stability or robustness, i.e., in terms of the implicit buffer times of consecutive trains.

2.4.1 Combinatorial Aspects

A main point in the discussion on railway models is whether it is possible to deal with complex combinatorial technical and economical constraints in a real-world setting or not. We do, of course, not claim that we can give a real answer to this question, but we want to give an example of a more realistic scenario to indicate that our approach has potential in this direction. To this purpose, we discuss a setting that extends the previous one, i.e., see Section 1.1, by allowing for combinatorial AND and XOR requirements.

With these extensions, it is possible to model most features of the bidding language, i.e., the specification of train requests in an auction environment, described in Borndörfer et al. (2006) [34]. Bids for complete tours can be expressed as AND connected bids, and an optional stop can be expressed as a XOR connection of requests for slots with and without this stop. An AND relation could further be useful to indent slots for a frequent service. Railway undertakings which can only operate a limited number of train slot could further be interested in formulating XOR bids. A way how to incorporate general connections for passengers is described in Mura (2006) [164], i.e., an auxiliary flow is defined that is induced by and coupled with the connective train slots.

Let a combinatorial bid k refer to some subset $I_k \subseteq I$ of bids for single train request; it may either be an AND or an XOR bid. An AND-bid stipulates that either all single slot bids in $I_k = \{i_1, i_2, \ldots, i_m\}, m \geq 2$ must be assigned or none of them. A XOR-bid states that at most one of the bids in the set I_k can be chosen. Let I_{AND} denote the set of AND bids, and I_{XOR} the set of XOR bids.

The arc based formulations (APP) and (ACP) can be easily extended by introducing a zero-one variable z_i for each train request i that is 1 if bid i is assigned and 0 else. These variables are useful in dealing with combinatorial bids by the following constraints:

$$\sum_{a \in \delta^i_{\text{out}}(s_i)} x_a - z_i = 1, \forall i \in I, \tag{5}$$

$$z_{i_n} - z_{i_{n+1}} = 0, \forall n \in \{1, 2, \ldots, |I_k| - 1\}, k \in I_{AND}, \tag{6}$$

$$\sum_{i \in I_k} z_i \leq 1, \forall k \in I_{XOR}. \tag{7}$$

Constraints 5 make sure that z_i is only one if train i is scheduled. Constraints 6 and 7 enforce combinatorial AND and XOR bids, i.e., an additional one for each XOR set and $|I_k| - 1$-many for each AND set k.

2.4.2 Robustness Aspects

We exploit the possibility to use the additional variables of the extended formulations (ACP) and (PCP) to measure robustness in terms of implicit available buffer times of a timetable. We refrain from supporting this by recent statistics to punctuality and reliability of any railway company. But obviously, decision makers are more and more sensitive to the importance of finding a good compromise between profitable and reliable timetables.

Robust optimization, that means the incorporation of data uncertainties through mathematical models in its original definition as proposed by Soyster (1973) [202], is not applicable to large scale optimization problems. Moreover these models produce too conservative solutions, which are resistant against all considered eventualities, but far away from implementable in real world. Robust optimization, however, has become a fruitful field recently because more and more optimization problems can be solved in adequate time. This opens the door to additionally deal with stochastic assumptions instead of only nominal given data. In Ben-Tal & Nemirovski (1998) [23] and El-Ghaoui, Oustry & Lebret (1998) [81], less conservative models were introduced, which adjust the robustness of the solution by some protection level parameters. Bertsimas & Sim (2003) [25] survey robust optimization theory and its network flow applications. Fischetti, Salvagnin & Zanette (2009) [91]; Kroon et al. (2006) [139]; Liebchen et al. (2007) [151]; Liebchen et al. (2009) [152] apply these robust considerations to the world of railways, i.e., to the periodic railway timetabling. They investigate a cyclic version of the timetabling problem, modeled as a Periodic Event Scheduling Problem and introduce a stochastic methodology of Light Robustness and Recoverable Robustness. For the detailed routing through stations or junctions, Caimi, Burkolter & Herrmann (2004) [58] and Delorme, Gandibleux & Rodriguez (2009) [74] proposed approaches to find delay resistant and stable routings. The aim of these considerations is to gain more insights into the trade-off between efficiency and robustness of solutions and find a practical "price of robustness".

We focus on a pure combinatorial optimization approach, which is somehow related to Ehrgott & Ryan (2002) [79] and Weide, Ryan & Ehrgott (2010) [212], broaching the issue of robustness in airline crew scheduling. We consider robustness (available buffer times, quality of day-to-day operations) and efficiency (used track kilometers, planned capacity utilization) to be incomparable entities and consequently favor a bi-criteria optimization approach. Later, Schöbel & Kratz (2009) [191] applied the same methodology to the problem of periodic railway timetabling.

We extend models (ACP) and (PCP) to measure robustness, which leads directly to a bi-criteria optimization approach of the problem. To determine efficient solutions, i.e., the Pareto-frontier, of the bi-criteria models we used the trivial so-called scalarization and ϵ-constraint method. More details on the general theory and solution of multi-criteria optimization problems can be found in Ehrgott (2005) [78].

In Schlechte & Borndörfer (2008) [188] details on a straight-forward column generation approach to solve the scalarized optimization problem can be found, i.e., we proved that the LP-relaxation of the (PCP) including an additional ϵ-constraint remains solvable in polynomial time.

However, let us explain the incorporation of some "robustness" on a simple example. By r_q we denote a robustness value for each configuration $q \in \mathcal{Q}$. We assume that a high robustness value r_q means configuration q is robust and a smaller the contrary. As a simplification, we expect $r_q = \sum_{a \in q} r_a$, i.e., the robustness of a configuration can be expressed as the sum of the robustness of its incident arcs.

Figure 14 illustrates the idea on a single track. Considering a track digraph D_j induced by three train requests. Straight forwardly maximizing the number of scheduled trains in our setting will always lead

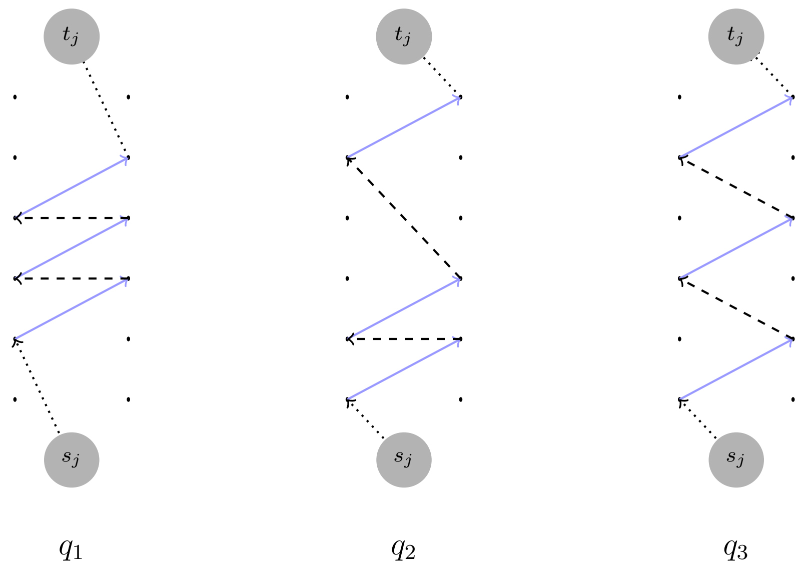


Figure 14: From fragile q_1 and q_2 to robust configuration q_3.

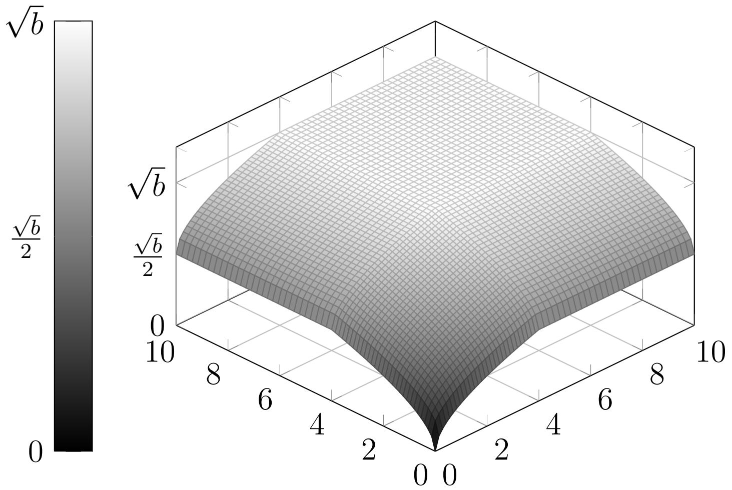


Figure 15: Robustness function r of two buffer arcs.

to a schedule with profit value 3, but, as you can see, this can result in a lot of varying schedules. In fact all $s_j t_j$-paths are solutions, e.g., the three shown in Figure 14. We are given a desired implicit buffer $b \in \mathbb{N}$, i.e., 5 minutes, which we maximally want to hedge against. Note that these are soft buffer times between train succession. Standard buffer time which must be strictly adhered to are already incorporated in the headway times.

Then the following robustness function $r : \mathbb{R}^{|A|} \to \mathbb{R}$ with

$$r((u,v)) := \begin{cases} \sqrt{b} & (u,v) \in \overline{A_j} \text{ and } t(v) - t(u) > b, \\ \sqrt{t(v) - t(u)} & (u,v) \in \overline{A_j} \text{ and } t(v) - t(u) \leq b, \\ 0 & \text{otherwise} \end{cases}$$

will measure the available buffers appropriately. Note that only "return arcs" contribute to the robustness measure. The function r benefits arcs with duration values close to or above b. Moreover this function balances the partition of the available implicit buffer times by its concaveness, see Figure 15. Assume $b = 2$ in our example in Figure 14. Then the first configuration q_1 has value $r_{q_1} = 0$, for the second configuration r_{q_2} is $\sqrt{2}$, and the third one has $r_{q_3} = 2$. For the sake of completeness we set r_q to a sufficiently big M for an empty configuration q, i.e., we use the b times half the length of the longest path in D_j. To find all efficient solutions, we propose a straight-forward combined weighted sum and ϵ-constraint hybrid method, see Ehrgott (2005) [78]. Considering model (PCP), this leads to the following objective function with a scalar $\alpha \in [0,1]$:

$$\max \ \alpha(\sum_{p \in P} w_p x_p) + (1 - \alpha)(\sum_{q \in Q} r_q y_q).$$

As a result, we can compile an analysis of the crucial parameters to support track allocation decisions as shown in Figure 16. In addition such a computational experiment produces a broad spectrum of solutions. Thus, new problem insights are provided and planners have the possibility to try complete new track allocation concepts.

We only present and discuss results for the linear relaxation of model (ACP). In Schlechte & Borndörfer (2008) [188] the settings and focus of these experiments are explained more precisely. On the right both objectives depending on α are shown. The extreme cases are as expected: For $\alpha = 1$, only the robustness measure contributes to the objective and is therefore maximized as much as possible at the cost of scheduling only some or even no trains. For $\alpha = 0$, the robustness measure does not contribute to the objective and is therefore low, while the total profit is maximal. With decreasing α, the total robustness monotonically decreases, while the total profit increases. On the left part of Figure 16 the Pareto frontier can be seen. Note that each computed pair of total robustness and profit constitutes a Pareto optimal point, i.e., is not dominated by any other attainable combination. Conversely,

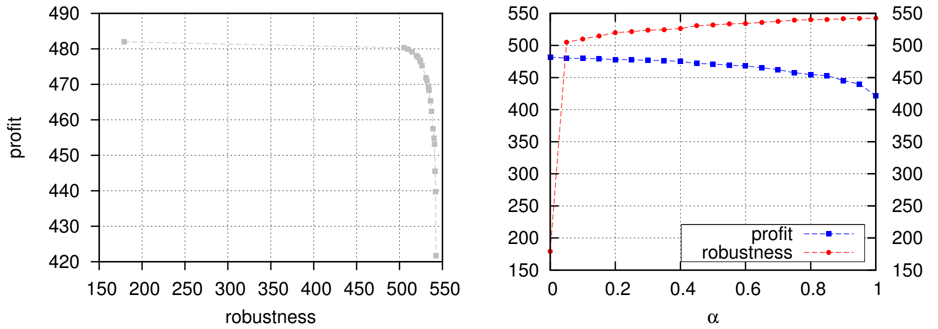


Figure 16: Pareto front on the left hand and total profit objective (blue, left axis) and total robustness objective (green, right axis) in dependence on α on the right hand.

any Pareto optimal solution of the LP relaxation can be obtained as the solution for some $\alpha \in [0,1]$, see, e.g., Ehrgott (2005) [78].

3 Branch and Price for Track Allocation

This Section discusses sophisticated algorithmic approaches to solve very large scale instances of the track allocation problem. Standard integer programming solver, such as `CPLEX`, `SCIP` or `GuRoBi`, can solve static model formulations like (APP) and (ACP) up to a certain problem size. However, to tackle large-scale instances we developed the optimization module `TS-OPT`. It solves the dynamic model formulation (PCP) by taking advantage of the approximate bundle method and a rapid branching heuristic to produce high quality solutions with a moderate running time even for very large scale instances. The aim of this chapter is to provide a comprehensive understanding of the less than conventional branch and price approach, i.e., the tailor made methods in `TS-OPT`.

3.1 Concept of `TS-OPT`

Schrijver (1998) [193] and Nemhauser & Wolsey (1988) [167] provide a comprehensive discussion on the general theory of integer programming. State of the art techniques to solve mixed integer programs, i.e., even the more general class of constraint integer programs, can be found in the prizewinning thesis Achterberg (2007) [3]. The basic methodology of branch and price was introduced in Barnhart et al. (1998)

[18]. Details can also be found in Villeneuve et al. (2005) [210]. In the following sections, we apply these technique to the model (PCP).

In Figure 17 the concept of `TS-OPT` is shown. In a first step the problem is constructed. This entails reading in all data, i.e., the macroscopic railway network and the train request set subject to the specification of the TTPlib, constructing the train scheduling graph $D = (V, A)$ as proposed in Algorithm 4, and constructing the track digraphs as discussed in Section 2.2.

Besides that the main algorithm can be divided in two parts. On the one hand the linear programming or Lagrangean relaxation is solved by a dynamic column generation approach, i.e., using an approximate bundle method or a LP solver to produce dual values. The pricing of variables are shortest path computations in large acyclic digraphs with respect to these duals. Fischer & Helmberg (2010) [89] propose a dynamic graph generation to solve these pricing problems for very large graphs, i.e., the original objective function has to fulfill the requirement that an earlier arrival is always beneficial. Unfortunately for our instances this is not always the case. However, this seems to be a fruitful approach to shrink the problem size of the pricing problems that could be extended to arbitrary objective functions. The idea is simple to use only a subset of the nodes and arcs and to define a border-set that will we adapted with respect to the duals and the solution of the "restricted" pricing problem.

On the other hand a branch and price heuristic, i.e., rapid branching, is used to produce high quality integer solutions. Instead of an exact branch and price approach, we only evaluate promising branch and bound nodes and perform some partial pricing. Furthermore, we only explore the branch of variables to 1 because there will be almost no effect when setting path and configuration variables to 0. The decision which subset is chosen is highly motivated by the solution of the relaxation, i.e., the best candidate set with respect to a score function depending on the bound and the size of the candidate set for a reasonable perturbation of the objective function. Section 3.2 and Section 3.3 will describe the components in more detail.

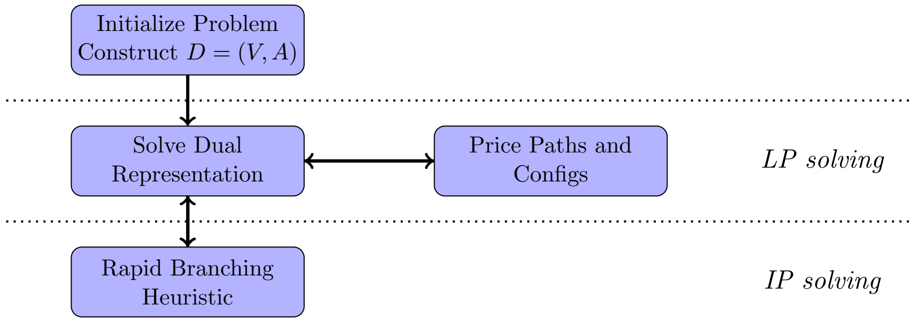


Figure 17: Flow chart of algorithmic approach in `TS-OPT`.

3.2 Solving the Linear Relaxation

In this section we use a slightly different notation with the following appropriate binary matrices A,B,C and D:

- ▷ $A \in \{0,1\}^{|I| \times |\mathcal{P}|}$ is the path-request incidence matrix,
- ▷ $B \in \{0,1\}^{|J| \times |\mathcal{Q}|}$ is the configuration-track incidence matrix,
- ▷ $C \in \{0,1\}^{|A^{\Psi}| \times |\mathcal{P}|}$ is the hyperarc-path incidence matrix,
- ▷ $D \in \{0,1\}^{|A^{\Psi}| \times |\mathcal{Q}|}$ is the hyperarc-configuration incidence matrix.

Without loss of generality we can change packing inequalities (PCP) (ii) and (iii) to partitioning equalities by introducing slack variables corresponding to empty paths $p \in \mathcal{P}$ with profit $w_p = 0$ or empty configuration, respectively. Observe that the upper bounds on x and y in model (PCP) are redundant because A and B are binary and we can assume that the profit coefficients w are positive, i.e., paths with negative profit value are redundant.

$$
\begin{array}{llcclll}
\text{(PCP)} & \max & w^T x & & & & \text{(i)} \\
 & s.t. & Ax & & & = \mathbf{1}, & \text{(ii)} \\
 & & & & By & = \mathbf{1}, & \text{(iii)} \\
 & & Cx & - & Dy & \le \mathbf{0}, & \text{(iv)} \\
 & & & & y & \in \{0,1\}^{|P|}, & \text{(v)} \\
 & & x & & & \in \{0,1\}^{|Q|}. & \text{(vi)}
\end{array}
$$

A standard technique to solve large scale linear relaxation as those of (PCP) is column generation, see Chapter I in Section 8.5 and Figure 14. We have already seen that the pricing problems are shortest path problems in acyclic digraphs, see Section 2.2 and Lemma 2.10.

However, in TS-OPT we implemented a slightly different approach based on a Lagrangean relaxation.

3.2.1 Lagrangean Relaxation

Lagrangean relaxation is a technique to find bounds for an optimization problem, e.g., upper bounds in case of maximization problems. In Hiriart-Urruty & Lemaréchal (1993) [116, 117]; Lemaréchal (2001) [147] the basics as well as further details can be found. Under certain circumstances also optimal solutions of the "convexified relaxation" are provided, see Frangioni (2005) [93]; Helmberg (2000) [114]; Weider (2007) [213].

Two time consuming problems have to be solved repeatedly in any column generation approach: First of all an optimal dual solution of the restricted problem has to be found, i.e., LPs have to be solved. Secondly, we have to find new columns or prove that none exists depending on the solutions of the LPs, i.e., dual values, by solving the pricing problems.

However, using Lagrangean relaxation and subgradient methods is often faster and less memory-consuming than LP-methods, see Weider (2007) [213]. Even if in general, this approach only gives bounds and approximated solutions of the relaxed problem. We transfer the large set of coupling constraints into the objective, i.e., therefore they can be violated by the solution of the Lagrangean relaxation. A Lagrangean relaxation with respect to the coupling constraints (iv) and a relaxation of the integrality constraints (v) and (vi) results in the Lagrangean dual:

$$\text{(LD)} \quad \min_{\lambda \geq \mathbf{0}} \left[\max_{\substack{Ax=\mathbf{1},\\ x\in[0,1]^{|P|}}} (w^\intercal - \lambda^\intercal C)x + \max_{\substack{By=\mathbf{1},\\ y\in[0,1]^{|Q|}}} (\lambda^\intercal D)y \right].$$

Each solution of (LD) gives a valid upper bound of (PCP). Let us define functions and associated arguments by

$$\begin{aligned}
f_P &: \mathbb{R}^{|A^\Psi|} \to \mathbb{R}, \quad \lambda \mapsto \max(w^\intercal - \lambda^\intercal C)x; \quad Ax = \mathbf{1}; \quad x \in [0,1]^{|P|}, \\
f_Q &: \mathbb{R}^{|A^\Psi|} \to \mathbb{R}, \quad \lambda \mapsto \max(\lambda^\intercal D)y; \qquad\quad By = \mathbf{1}; \quad y \in [0,1]^{|Q|}, \\
f_{P,Q} &:= f_P + f_Q.
\end{aligned}$$

That are longest path problems in acyclic digraphs with respect to λ and

$$\begin{aligned}
x_P(\lambda) &:= \mathrm{argmax}_{x \in [0,1]^{|P|}} f_P(\lambda), \\
y_Q(\lambda) &:= \mathrm{argmax}_{y \in [0,1]^{|Q|}} f_Q(\lambda),
\end{aligned}$$

breaking ties arbitrarily. With this notation, (LD) becomes

$$\text{(LD)} \quad \min_{\lambda \geq \mathbf{0}} f_{P,Q}(\lambda) = \min_{\lambda \geq \mathbf{0}} \left[f_P(\lambda) + f_Q(\lambda) \right].$$

It is well known that the Lagrangean dual of an integer linear program provides the same bound as a continuous relaxation involving the convex hull of all the optimal solutions of the Lagrangean relaxation. The functions f_P and f_Q are convex and piecewise linear. Their sum $f_{P,Q}$ is therefore a decomposable, convex, and piecewise linear function; $f_{P,Q}$ is, in particular, nonsmooth. This is precisely the setting for the proximal bundle method.

3.2.2 Bundle Method

The proximal bundle method (PBM) is a method to minimize an unbounded, continuous, convex, and possibly non-smooth function $f : \mathbb{R}^m \mapsto \mathbb{R}$. The PBM can be used in combination with Lagrangean relaxation to approximate primal and dual solutions of linear programs. A detailed description of the bundle method itself can be found in Kiwiel (1990) [127] and of its quadratic subproblem solver in Kiwiel (1995) [128].

In the following we will discuss our straight-forward adaption of the general bundle method. We use the PBM to approximate LP-relaxations of model (PCP) via the Lagrangean problem (LD), defined in Section 3.2.1. The corresponding computational results can be found in

Chapter IV. The LP-relaxation of (PCP) is in general too large to be solved by standard solvers such as the barrier algorithm or the dual simplex because theses LPs consist in general of millions of columns for the paths and configurations and several thousands of rows for the coupling constraints, i.e., even if we already reduce theses constraints by the definition of A^{Ψ}.

When applied to (LD), the PBM produces two sequences of iterates $\lambda^k, \mu^k \in \mathbb{R}^{|A^{\Psi}|}$, $k = 0, 1, \ldots$. The points μ^k are called *stability centers*; they converge to a solution of (LD). The points λ^k are trial points; function evaluations (line 5 of Algorithm 5) at the trial points result either in a shift of the stability center, or in some improved approximation of $f_{P,Q}$.

More precisely, the PBM computes at each iteration for λ^k linear approximations

$$
\begin{aligned}
f_P(\lambda;\lambda^k) &:= f_P(\lambda^k) + g_P(\lambda^k)^{\mathsf{T}}(\lambda - \lambda^k),\\
f_Q(\lambda;\lambda^k) &:= f_Q(\lambda^k) + g_Q(\lambda^k)^{\mathsf{T}}(\lambda - \lambda^k),\\
f_{P,Q}(\lambda;\lambda^k) &:= f_P(\lambda;\lambda^k) + f_Q(\lambda;\lambda^k),
\end{aligned}
$$

of the functions f_P, f_Q, and $f_{P,Q}$ by determining the function values $f_P(\lambda^k)$, $f_Q(\lambda^k)$ and the subgradients $g_P(\lambda^k)$ and $g_Q(\lambda^k)$; by definition, these linear approximations underestimate the functions f_P and f_Q, i.e., $f_P(\lambda;\lambda^k) \leq f_P(\lambda)$ and $f_Q(\lambda;\lambda^k) \leq f_Q(\lambda)$ for all λ. Note that f_P and f_Q are polyhedral, such that the subgradients can be derived from the arguments $y(\lambda^k)$ and $x(\lambda^k)$ associated with the multiplier λ^k as

$$
\begin{aligned}
g_P(\lambda^k) &:= - \quad Cx_P(\lambda^k) &&= - \sum_{a \ni p \in P, b \in A^{\Psi}, a \in [b]} x_P(\lambda_b^k),\\
g_Q(\lambda^k) &:= \quad\;\; Dy_Q(\lambda^k) &&= \sum_{b \ni q \in Q, b \in A^{\Psi},} y_Q(\lambda_b^k),\\
g_{P,Q}(\lambda^k) &:= - \quad Cx_P(\lambda^k) + Dy_Q(\lambda^k).
\end{aligned}
$$

This linearization information is collected in so-called *bundles*

$$
\begin{aligned}
J_P^k &:= \{(\lambda_l, f_P(\lambda_l), g_P(\lambda_l) : l = 0, \ldots, k\},\\
J_Q^k &:= \{(\lambda_l, f_Q(\lambda_l), g_Q(\lambda_l) : l = 0, \ldots, k\}.
\end{aligned}
$$

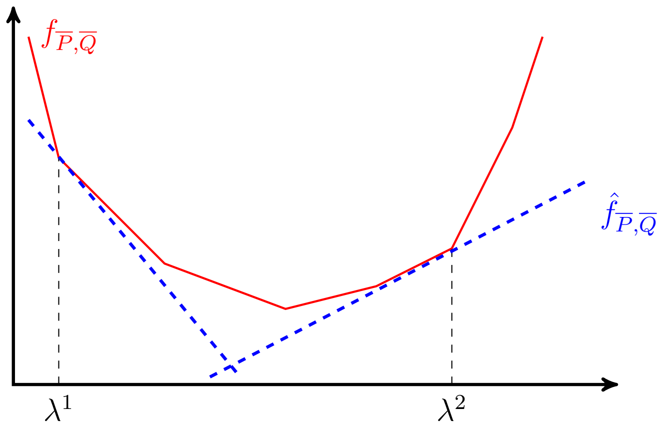


Figure 18: Cutting plane model $\hat{f}_{P,Q}$ of Lagrangean dual $f_{P,Q}$.

We will use notations such as $\lambda_l \in J_P^k$, $g_P(\lambda_l) \in J_P^k$, etc. to express that the referenced item is contained in some appropriate tuple in the bundle associated to the path variables of iteration k. The PBM uses the bundles to build piecewise linear approximations

$$\begin{aligned}
\hat{f}_P^k(\lambda) &:= \max_{\lambda_l \in J_P^k} f_P(\lambda; \lambda_l), \\
\hat{f}_P^k(\lambda) &:= \max_{\lambda_l \in J_Q^k} f_Q(\lambda; \lambda_l), \\
\hat{f}_{P,Q}^k &:= \hat{f}_P^k + \hat{f}_Q^k.
\end{aligned}$$

of $f_{P,Q}$, see Figure 18. Furthermore, a quadratic term is added to this model that penalizes large deviations from the current stability center μ^k. The direction (line 3) to the next trial point λ_{k+1} is calculated by solving the quadratic programming problem

$$(QP_{P,Q}^k) \qquad \lambda^{k+1} := \underset{\lambda}{\operatorname{argmin}}\, \hat{f}_{P,Q}(\lambda) - \tfrac{u}{2} \left\| \mu^k - \lambda \right\|^2.$$

Denote by u a positive weight (step size) that can be adjusted to increase accuracy or convergence speed. If the approximated function value $\hat{f}_{P,Q}^k(\lambda^{k+1})$ at the new iterate λ^{k+1} is sufficiently close to the function value $f_{P,Q}(\mu^k)$, the PBM stops; μ^k is the approximate solution. Otherwise a descent test (line 8) is performed whether the predicted decrease $f_{P,Q}(\mu^k) - \hat{f}_{P,Q}^k(\lambda^{k+1})$ leads to sufficient real decrease $f_{P,Q}(\mu^k) - f_{P,Q}(\lambda^{k+1})$. In this case, the model is judged accurate and a serious step is done, i.e., the stability center is moved to $\mu^{k+1} := \lambda^{k+1}$.

Algorithm 5: Proximal Bundle Method (PBM) for (LD) of (PCP).

Data: (LD) of (PCP) instance, starting point $\lambda^0 \in \mathbb{R}^n$, weights $u^0, m > 0$, optimality tolerance $\epsilon \geq 0$.
Result: primal $x_P, y_Q \in \mathbb{R}^{|P| \times |Q|}$ and dual approximation $\mu^i \in \mathbb{R}^n$ of optimal solutions of the (LD)

1 **init** $k \leftarrow 0$, $J_P^k \leftarrow \{\lambda^k\}$, $J_Q^k \leftarrow \{\lambda^k\}$, and $\mu^k = \lambda^k$;
2 **repeat** `/* until tolerance is reached */`
3 **solve** problem $(QP_{P,Q}^k)$;
`/* find direction */`
4 **compute** trial point λ^{k+1}, $\tilde{g}_P^k$, $\tilde{g}_Q^k$;
5 **compute** $f_P(\lambda^{k+1})$, $g_P(\lambda^{k+1})$, $f_Q(\lambda^{k+1})$, $g_Q(\lambda^{k+1})$;
6 **select** $J_P^{k+1} \subseteq J_P^k \cup \{(\lambda^{k+1}, f_P(\lambda_{k+1}), g_P(\lambda_{k+1})), (\lambda^{k+1}, \hat{f}P^k(\lambda^{k+1}), \tilde{g}_P^k)\}$;
7 **select** $J_Q^{k+1} \subseteq J_Q^k \cup \{(\lambda^{k+1}, f_Q(\lambda^{k+1}), g_Q(\lambda^{k+1})), (\lambda^{k+1}, \hat{f}_Q(\lambda^{k+1}), \tilde{g}_Q)\}$;
`/* update bundle set */`
8 **if** $f_{P,Q}(\mu^i) - f_{P,Q}(\lambda^{k+1}) \leq m(f_{P,Q}(\mu^k) - \hat{f}_{P,Q}^k(\lambda^{k+1}))$ **then**
9 $\mu^{k+1} \leftarrow \mu^k$;
10 **else** `/* update stability center */`
11 $\mu^{k+1} \leftarrow \lambda^{k+1}$;
12 **compute** u_{k+1}, $k \leftarrow k + 1$;
`/* update stepsize */`
13 **until** $\hat{f}_{P,Q}^k(\lambda^{k+1}) - f_{P,Q}(\mu_k) < \epsilon(1 + |f_{P,Q}(\mu^k)|)$;

In the other case, we call this iteration a null step, i.e., in which only the approximation of the function by the bundles was improved.

The bundles are updated (line 6 and 7) by adding the information computed in the current iteration, and, possibly, by dropping some old information. More precisely, vectors $\tilde{g}_P^k$ and $\tilde{g}_Q^k$ are aggregated subgradients, which will be explained in the next paragraph. Finally, we adopt the stepsize. Then the next iteration starts, see Algorithm 5 for a complete pseudo code of the PBM.

Besides function and subgradient calculations, the main work in the PBM is the solution of the quadratic problem $(QP^k_{P,Q})$. This problem can also be stated as

$$
\begin{array}{llllll}
(\mathrm{QP}^{\mathrm{k}}_{\mathrm{P,Q}}) & \max & v_P + & v_Q & -\frac{u}{2}\left\|\mu^k - \lambda\right\|^2 & \\
\text{(i)} & & v_P & & -f_P(\lambda;\lambda_l) \leq 0 & \forall \lambda_l \in J^k_P \\
\text{(ii)} & & & v_Q & -f_Q(\lambda;\lambda_l) \leq 0 & \forall \lambda_l \in J^k_Q.
\end{array}
$$

A dualization is in the equivalent formulation

$$
\begin{array}{lll}
(\mathrm{DQP}^{\mathrm{k}}_{\mathrm{P,Q}}) & \operatorname{argmax} & \sum\limits_{\lambda_l \in J^k_P} \alpha_{P,l} f_P(\mu^k;\lambda) + \sum\limits_{\lambda_l \in J^k_Q} \alpha_{Q,l} f_Q(\mu^k;\lambda) \\
& & -\frac{1}{2u}\left\| \sum\limits_{\lambda_l \in J^k_P} \alpha_{P,l} g_P(\lambda) + \sum\limits_{\lambda_l \in J^k_Q} \alpha_{Q,l} g_Q(\lambda) \right\|^2 \\
& & \sum\limits_{\lambda_l \in J^k_P} \alpha_{P,l} \quad = 1 \\
& & \sum\limits_{\lambda_l \in J^k_Q} \alpha_{Q,l} \quad = 1 \\
& & \alpha_P, \alpha_Q \quad \geq 0.
\end{array}
$$

Here, $\alpha_P \in [0,1]^{J^k_P}$ and $\alpha_Q \in [0,1]^{J^k_Q}$ are the dual variables associated with the constraints $(QP^k_{P,Q})$ (i) and (ii), respectively. Given a solution $(\alpha_P, \alpha_Q$ of $((DQP^k_{P,Q})$, the vectors

$$
\begin{aligned}
\tilde{g}^k_P &:= \sum_{\lambda_l \in J^k_P} \alpha_P g_P(\lambda_l), \\
\tilde{g}^k_Q &:= \sum_{\lambda_l \in J^k_Q} \alpha_Q g_Q(\lambda_l), \\
\tilde{g}^k_{P,Q} &:= \tilde{g}^k_P + \tilde{g}^k_Q,
\end{aligned}
$$

are convex combinations of subgradients; they are called *aggregated subgradients* of the functions f_P, f_Q, and $f_{P,Q}$, respectively. It can be shown that they are, actually, subgradients of the respective functions at the point λ^{k+1} and, moreover, that this point can be calculated by means of the formula

$$\lambda^{k+1} = \mu + \frac{1}{u}\left[\sum_{\lambda_l \in J_P^k} \alpha_P g_P(\lambda_l) + \sum_{\lambda_l \in J_Q^k} \alpha_Q g_Q(\lambda_l)\right].$$

Note that $(DQP_{P,Q}^k)$ is again a quadratic program, the dimension is equal to the size of the bundles, while its codimension is only two. For solving this problem we use a specialized version of the *spectral bundle method*, see Kiwiel (1990) [127], Kiwiel (1995) [128] and Borndörfer, Löbel & Weider (2008) [37]. Finally, the PBM (without stopping) is known to have the following properties:

- ▷ The series (μ^k) converges to an optimal solution of (LD), i.e. an optimal dual solution of the LP-relaxation of (PCP).
- ▷ The series $(x_P^k(\lambda^k), y_Q^k(\lambda^k))$ defined as

  $$(x_P^k(\lambda^k), y_Q^k(\lambda^k)) = \left(\sum_{\lambda_l \in J_P^k} \alpha_P x(\lambda_l), \sum_{\lambda_l \in J_Q^k} \alpha_Q y(\lambda_l)\right)$$

  converges to an optimal primal solution of the LP-relaxation of (PCP).

Furthermore, the primal approximation is useful to guide branching decision of the primal heuristic as we will describe in Section 3.3. The bundle size controls the convergence speed of the PBM. If large bundles are used, less iterations might be needed because of the better approximation model, however, problem $(QP_{P,Q}^k)$ becomes more difficult. We use a simple control schema for the stepsize u similar to Weider (2007) [213]. The idea is to increase the stepsize if serious steps are performed, if the distance of new trial point and the last one is small. In case of null steps, we gradually decrease the stepsize u.

In Chapter IV, Section 2 we present results of various experiments with different strategies and parameter settings of our bundle implementation.

3.3 Solving the Primal Problem by Rapid Branching

In this section, we describe a heuristic approach based on the branch and price principle to tackle very large scale instances. In fact it is a branch-and-generate (BANG) heuristic, i.e., a branch-and-price algorithm with partial branching, see Subramanian et al. (1994) [204]. The heuristic can be classified as a special plunging heuristic with a objective perturbation branching rule.

Wedelin (1995) [211] a similar successful heuristic which perturbs the objective function of large set-partitioning problems in a dual ascent method to find integral solution. In Weider (2007) [213] this heuristic was invented as rapid branching. Therein, impressive results for large-scale instances of integrated vehicle and duty scheduling problems arising in public transport are presented. We will adopt main ideas and transfer them to the (PCP) formulation of the track allocation problem.

A simple rounding heuristic is used in Fischer et al. (2008) [90] to produce feasible integral solution of the (PPP), but sometimes fails to produce high quality solutions. In Cacchiani, Caprara & Toth (2007) [52] a greedy heuristic based on near-optimal Lagrangian multiplier was used to produce solutions of the (PPP). In Section 1 we will see that simple greedy approaches or rounding heuristics also fails very often for the (PCP).

Instead of branching on variables, Foster & Ryan (1991) [92] proposed another branching rule, which can be generalized as branching on arcs. One branching decision is to fix an arc to one, the other branch to ignore the arc completely. Lusby (2008) [158] discussed this solution approach to a generalization of (PPP). This branching rule results normally in more balanced branch and bound trees. Koch, Martin & Achterberg (2004) [134] give a general survey on branching rules for solving MIPs.

The motivation of rapid branching given in Weider (2007) [213] applies also in our setting to a large extent.

- ▷ The fixing of single variables (path or configuration) to zero changes the problem only slightly.
- ▷ The fixing of single arcs to zero changes the problem only slightly, i.e., in general the set of arcs is too large.
- ▷ The fixing of single arcs to one is equivalent to fixing a large set of arcs to zero.
- ▷ The fixing of single variables (path or configuration) to one is equivalent to fix all arcs of the corresponding columns to one.

Same observations for large scale LPs that are solved by column generation are mentioned in Lübbecke & Desrosiers (2005) [156]. Thus, rapid branching fixes a set of variables at once to one. Which somehow reflects our goal to explore only a *main branch* and to reach fast high quality solutions. The idea of the perturbation branching rule is to find one branch, called the main branch, that fixes as many variables as possibles to one to quickly find a solution of (PCP). This is done by solving a series of LP-relaxations of (PCP) with varying profit functions w. We perturb the profit function from one iteration to the next to "make the LP more integer": The profit of variables with large primal values are increased to move them towards an even higher value or to keep the value at one.

The other branches are unimportant unless the main branch turns out to either not include a feasible solution or to include only feasible solutions with too low profit. Borndörfer, Löbel & Weider (2008) [37], see also the thesis of Weider (2007) [213], proposed also an associated backtracking mechanism to correct wrong decisions. Our setting is of obvious similarity, and it will turn out that rapid branching can indeed be successfully applied to solve large-scale track allocation problem. Even more we are confident that a generalized variant of rapid branching can be a very effective plunging heuristic in standard MIP solvers.

Let $l, u \in \{0,1\}^{P\times Q}$, $l \leq u$, be vectors of bounds that model fixings of variables to 0 and 1. Denote by $L := \{j \in P \times Q : u_j = 0\}$ and $U := \{j \in P \times Q : l_j = 1\}$ the set of variables fixed to 0 and 1, respectively, and by

$$\begin{array}{llcccrl}
(\mathrm{PCP})(l,u) & \max & w^T x & & & & \text{(i)} \\
 & s.t. & Ax & & & = \mathbf{1}, & \text{(ii)} \\
 & & & & By & = \mathbf{1}, & \text{(iii)} \\
 & & Cx & - & Dy & \leq \mathbf{0}, & \text{(iv)} \\
 & & l & \leq & \begin{pmatrix} x \\ y \end{pmatrix} & \leq u, & \text{(v)}
\end{array}$$

the IP derived from (PCP) by such fixings. Denote further by $N \subseteq P \times Q = S$ some set of variables which have, at some point in time, already been generated by a column generation algorithm for the solution of (PCP). Let (RPCP) and (RPCP)(l,u) be the restrictions of the respective IPs to the variables in N (we assume that $L, U \subseteq N$ holds at any time when such a program is considered, i.e., variables that have not yet been generated are not fixed). Finally, denote by (MLP), $(MLP)(w,l,u)$, (RMLP), and $(RMLP)(w,l,u)$ the LP relaxations of the integer programs under consideration; (MLP) and $(MLP)(w,l,u)$ are called *master LPs*, $(RMLP)$ and $(RMLP)(w,l,u)$ *restricted master LPs* (the objective w is included in the notation for $(MLP)(w,l,u)$ and $(RMLP)(w,l,u)$ for reasons that will become clear in the next paragraphs.

Rapid branching tries to compute a solution of (PCP) by means of a search tree with nodes $(\mathrm{PCP})(l,u)$. Starting from the root $(\mathrm{PCP}) = (\mathrm{PCP})(0,\mathbf{1})$, nodes are spawned by additional variable fixes using a strategy that we call *perturbation branching*. The tree is depth-first searched, i.e., rapid branching is a plunging heuristic. The nodes are analyzed heuristically using restricted master LPs $(RMLP)(w,l,u)$. The generation of additional columns and node pruning are guided by so-called *target values* as in the branch-and-generate method. To escape unfavorable branches, a special *backtracking mechanism* is used that performs a kind of partial binary search on variable fixings. The idea of the method is to try to make rapid progress towards a feasible integer solution by fixing large numbers of variables in each iteration, repairing infeasibilities or deteriorations of the objective by regeneration of columns if possible and by controlled backtracking otherwise.

The idea of *perturbation branching* is to solve a series of (MLP)s with objectives $w^k, k = 0,1,2,\ldots$ that are perturbed in such a way that the associated LP solutions x^k are likely to become more and more integral. In this way, we hope to construct an almost integer solution at little computational cost. The perturbation is done by increasing the utility of variables with LP values close to one according to the formula:

$$\begin{array}{lll} w_j^0 & := w_j, & j \in N \\ w_j^{k+1} & := w_j^k + w_j \alpha x_j^2, & j \in N, \quad k = 0,1,2,\ldots. \end{array}$$

The progress of this procedure is measured in terms of the potential or score function

$$v(x^k) := w^\mathsf{T} x + \delta |B(x^k)|,$$

where ϵ and δ are parameters for measuring near-integrality and the relative importance of near-integrality (we use $\epsilon = 0.1$ and $\delta = 1$), and $B(x^k) := \{j \in N : x_j^k > 1-\epsilon\}$ is the set of variables that are set or almost set to one, i.e., also called candidate set. The perturbation is continued as long as the potential function increases; if the potential does not increase for some time a spacer step is taken in an attempt to continue. Another reasonable criteria could be that the candidate set does not change. On termination, the variables in the set $B(x^k)$ associated with the highest potential are fixed to one. If no variables at all are fixed, we choose a single candidate by *strong branching*, see Applegate et al. (1995) [13]. Objective perturbation has also been used by Wedelin (1995) [211] for the solution of large-scale set partitioning problems, and, e.g., by Eckstein & Nediak (2007) [77] in the context of general mixed integer programming.

Algorithm 6: Perturbation Branching.

Data: $(RMLP)(w, l, u)$, integrality tolerance $\epsilon \in [0, 0.5)$, integrality weight $\delta > 0$, perturbation factor $\alpha > 0$, bonus weight $M > 0$, spacer step interval k_s, iteration limit $k_{\max}$

Result: set of variables B^* that can be fixed to one

init $k \leftarrow 0$; $w^0 \leftarrow w$; $B^* \leftarrow \emptyset$; $v^* \leftarrow \infty$;
while $k < k_{\max}$ **do** `/* maximum number of iterations not reached */`
    compute $x^k \leftarrow \operatorname{argmax}(RMLP)(w^k, l, u)$;
    set $B^k \leftarrow \{j : x_j^k \geq 1 - \epsilon,\ l_j = 0\}$;
    set $v(x^k) \leftarrow w^\top x^k + \delta |B^k|$;
    if x^k is integer **then**
        set $B^* \leftarrow B^k$; `/* candidates found */`
        break;
    else
        if $k \equiv 0 \mod k_s$ and $k > 0$ **then**
            set $j^* \leftarrow \operatorname{argmax}_{l_j=0} x_j^k$;
            set $w_j^k \leftarrow M$;
            set $B^* \leftarrow B^k \cup \{j^*\}$; `/* spacer step */`
        else
            if $v(x^k) > v^*$ **then**
                set $B^* \leftarrow B^k$; $v^* \leftarrow v(x^k)$; $k \leftarrow -1$; `/* progress */`
            set $w_j^{k+1} \leftarrow w_j^k + \alpha w_j (x_j^k)^2 \quad \forall j$; `/* perturb */`
    set $k \leftarrow k + 1$;
if $B^* = \emptyset$ **then**
    set $B^* \leftarrow \{j^*\} \leftarrow$ strongBranching() ; `/* strong branching */`
return B^*;

Algorithm 6 gives a pseudocode listing of the complete perturbation branching procedure. The main work is in solving the perturbed re-

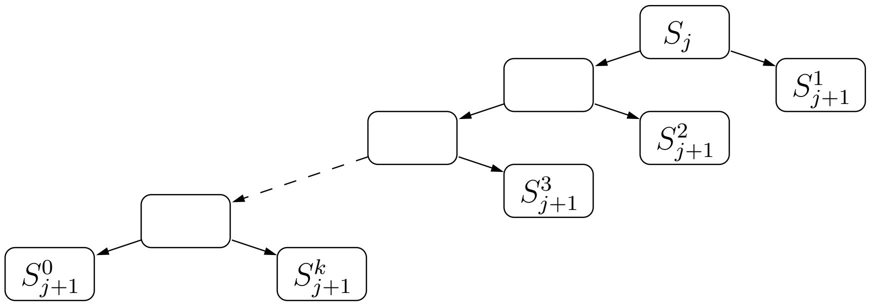


Figure 19: The new solution sets at iteration k, source: Weider (2007) [213]

duced master LP (line 3) and generating new variables if necessary. Fixing candidates are determined (line 4) and the potential is evaluated (line 5). If the potential increases (lines 15–16), the perturbation is continued (line 17). If no progress was made for k_s steps (line 10), the objective is heavily perturbed by a spacer step in an attempt to continue (lines 10–13). However, even this perturbation does not guarantee that any variable will get a value above $1 - \epsilon$, if $\epsilon < 1/2$. If this happens and the iteration limit is reached, a single variable is fixed by strong branching (line 20).

The fixing candidate sets B^* produced by the perturbation branching algorithm are used to set up nodes in the branch-and-generate search tree by imposing bounds $x_j = 1$ for all $j \in B^*$. This typically fixes many variables to one, which is what we wanted to achieve. However, sometimes too much is fixed and some of the fixings turn out to be disadvantageous. In such a case we must backtrack. We propose to do this in a binary search manner by successively undoing half of the fixes until either the fixings work well or only a single fix is left as shown in Figure 19 This procedure is called *binary search branching*.

Here are the details. Let B^* be a set of potential variable fixes and $K = |B^*|$. Order the variables in B^* by some criterion as $i_1, i_2, \ldots, i_K$ and define sets

$$B^*_k := \{i_1, \ldots, i_k\}, \quad k = 1, \ldots, K.$$

Consider search tree nodes defined by fixing

$$x_j = l_j = 1, \quad j \in B^*_k, \quad k = K, \lceil K/2 \rceil, \lceil K/4 \rceil, \ldots, 2, 1.$$

These nodes are examined in the above order. Namely, we first try to fix all variables in B^*_K to one, since this raises hopes for maximal progress. If this branch comes out worse than expected, it is pruned, and we backtrack to examine $B^*_{\rceil K/2 \lceil}$ and so on until possibly B^*_1 is reached. The resulting search tree is a path with some pruned branches, i.e., binary search branching is a plunging heuristic. In our implementation, we order the variables by increasing reduced cost of the restricted root LP, i.e., we unfix half of the variables of smallest reduced cost. This sorting is inspired by the scoring technique of Caprara, Fischetti & Toth (1998) [60]. The decision whether a branch is pruned or not is done by means of a *target value* as introduced by Subramanian et al. (1994) [204]. Such a target value is a guess about the development of the LP bound if a set of fixes is applied; we use a linear function of the integer infeasibility. If the LP bound stays below the target value, the branch develops according to our expectations, if not, the branch "looks worse than expected" and we backtrack.

Chapter IV

Case Studies

In the last chapter we report on several computational experiments. Section 1 compares standard models and our novel extended formulation. In Section 2 we present results of several computational experiments to analyze the benefit of the algorithmic ingredients of our novel solution approach, i.e., the proximal bundle method 2.2 and the rapid branching heuristic 2.3.

Section 3 discusses results of an auction based track allocation. These results and evaluation have a theoretical and visionary character due to various questionable assumptions. Thus, we will also discuss pure theoretical and rather philosophical auction design questions.

Finally, we present computational results for solving track allocation problems on real-world scenarios for the Simplon corridor in Section 4. The basis for the presented results are the contributions of Chapter II and Chapter III. Furthermore, it demonstrates the practical applicability of optimization for railway track allocation. To the best knowledge of the author and confirmed by several railway practitioners this was the first time that on a macroscopic scale automatically produced track allocations fulfill the requirements of the original microscopic model.

1 Model Comparison

`TS-OPT` is implemented in the programming language C++. It is able to generate the static formulations $(\mathrm{APP})'$ and (ACP) as well as to solve model (PCP) by the proposed branch and price algorithm in Chapter III, Section 3. All computations in the following were performed on computers with an Intel Core 2 Extreme CPU X9650 with 3 GHz, 6 MB cache, and 8 GB of RAM, or an Intel Core i7 870 with 3 GHz, 8 MB cache, and 16 GB of RAM.

This choice is motivated as follows. (APP)′ is the dominant model in the literature, which we want to benchmark. (PCP) and (ACP) are equivalent models that improve (APP)′. (APP)′ and (ACP) are both arc-based, rather easy to implement and very flexible.

We did not implement the strong packing model (APP), and also not (PPP), because these models are not robust with respect to changes in the problem structure, namely, their simplicity depends on the particular clique structure of interval graphs. If more complex constraints are considered, these models can become hard to adapt. In fact, the instances that we are going to consider involve real world headway matrices that give rise to more numerous and more complex clique structures as mentioned by Fischer et al. (2008) [90]. Thus, an implementation of suitably strong versions of models (APP) and (PPP) would have been much more difficult than an implementation of the basic versions discussed in Chapter III, Section 2.1.

In marked contrast to these models is our configuration model in which headway constraints are easy to implement. The reason is simple that they specify possible follow-on trips on a track, which is precisely what a configuration does. Formulation (PCP) is in this sense very robust to handle headway conflicts, if the corresponding headway matrices are transitive. It is also well suited for column generation to deal with very large instances as we will discuss in Section 2.

We performed computational experiments with both static models. Our aim was to gather from these test runs information that would allow us to choose a “winner”, i.e., a model that, for the range of the problem instances we address, displays the best computational performance in practice.

The instances for the comparison were solved as follows. The root LP-relaxations of the static models (APP)′ and (ACP) were solved with the barrier method of IBM ILOG CPLEX 11.2 (64 Bit, 4 threads, barrier), see CPLEX 12.2.0.2 [119]. Then, the MIP solver of CPLEX was called for a maximum of at most 1h of running time.

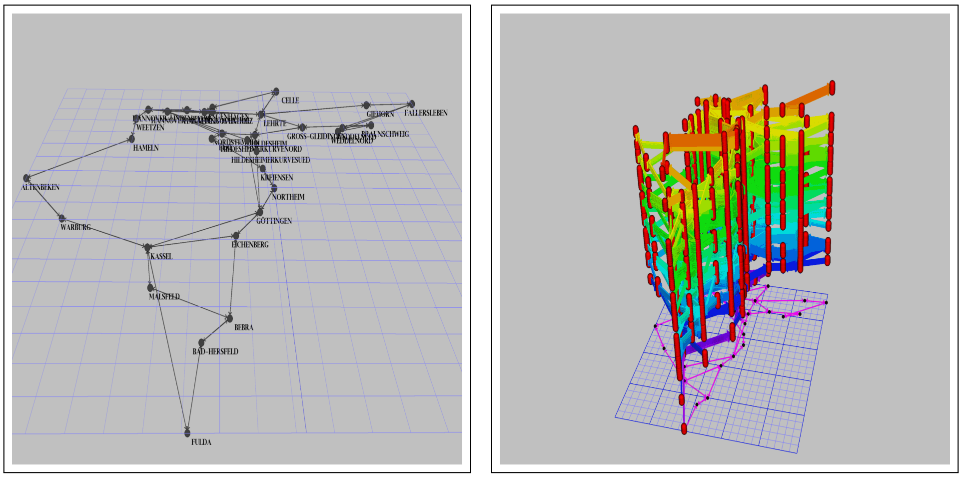


Figure 1: Infrastructure network (left), and train routing digraph (right); individual train routing digraphs bear different colors.

1.1 Effect of Flexibility

In our experiments, we consider the Hanover-Kassel-Fulda area of the German long-distance railway network. All our instances are based on the macroscopic infrastructure network that is illustrated in Figure 1. It includes data for 37 stations, 120 tracks and 6 different train types (ICE, IC, RE, RB, S, ICG). Our project partner from IVE and SFWBB provided this macroscopic data. Because of various possible turn around and running times for each train type, this produces an macroscopic railway model with 146 nodes, 1480 arcs, and 4320 headway constraints – infrastructure scenario HAKAFU_SIMPLE.

Based on the 2002 timetable of Deutsche Bahn AG, we constructed several scenarios. We considered all trains inside that area in a time interval of about 480 minutes at a normal weekday from 9:00 to 17:00 (or smaller). We varied several objective parameters, selected subsets of the request, and generated artificial additional freight traffic, see Mura (2006) [164].

All instances related to HAKAFU_SIMPLE are freely available at our benchmark library TTPlib, see Erol et al. (2008) [85]. From the test runs we have made, we have chosen to discuss the results of instance HAKAFU_SIMPLE and REQ_36 – a scenario with 285 train requests.

Table 1 demonstrates that reasonable track allocation problems can become very large even if the consider time windows are limited. The main objective is to maximize the total number of trains in the sched-

	before preprocessing		after preprocessing	
τ	*#nodes*	*#arcs*	*#nodes*	*#arcs*
0	123239	267080	282	316
2	140605	300411	863	1005
4	155607	331631	2611	3589
6	169989	361927	4228	6372
8	186049	395688	6563	10515
10	204423	434499	9310	15726
12	224069	476431	12380	21730
14	245111	522119	15779	28569
16	267989	572185	19838	36673
18	291473	625083	24374	45882
20	316631	681668	29738	56951

Table 1: Size of the test scenarios REQ_36.

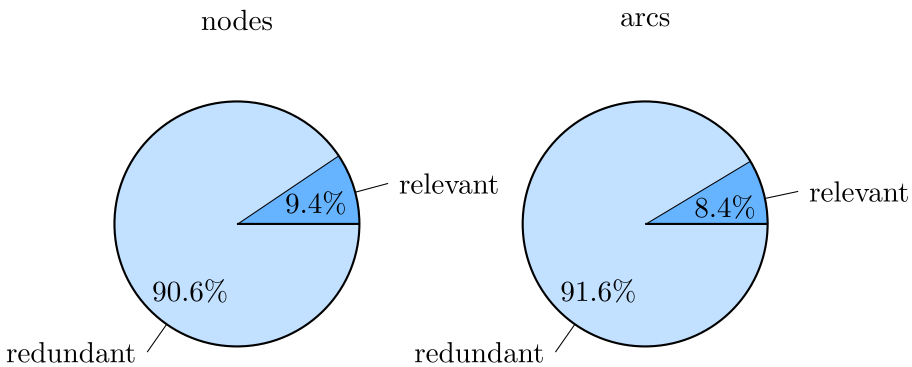


Figure 2: Reduction of graph size by trivial preprocessing for scenarios REQ_36 and $\tau = 20$.

ule; on a secondary level, we slightly penalize deviations from certain desired departure and arrival times. "Flexibility" to reroute trains is controlled by departure and arrival time windows of length at most τ, where τ is a parameter. To be precise, let t_{opt} be the optimal arrival (or departure) time then we set the minimum arrival (or departure) time t_{min} to $t_{opt} - \frac{\tau}{2}$ and the maximum arrival (or departure) time t_{max} to $t_{opt} + \frac{\tau}{2}$, respectively. Hence, increasing τ from 0 to 20 minutes in steps of 2 minutes increases flexibility, but also produces larger train routing digraphs and IPs. We used a maximum of 20 minutes because in the allocation process for the annual timetable desired times (in minutes) were varied of at most 5 minutes.

After graph preprocessing by algorithm 4 (eliminating arcs and nodes which cannot be part of a feasible train route), the resulting 11 instances have the sizes listed in Table 1. Figure 2 shows the concrete benefit of the graph preprocessing for the largest instance of that set.

τ	*#rows*	*#cols*	*#trains*	ub^*	v(LP)	v^*	gap in %	$t_\sum$ in s	*#bbn*
0	288	316	29	37.10	**37.10**	**37.10**	–	8.87	1
2	962	1005	67	99.92	**99.92**	**99.92**	–	8.20	1
4	3134	3589	121	219.05	222.92	219.05	–	8.90	68
6	5552	6372	143	238.67	246.25	238.67	–	9.49	570
8	9584	10515	161	260.77	279.99	260.77	–	11.14	569
10	15481	15726	185	309.54	322.47	309.54	–	12.93	518
12	23135	21730	198	336.63	348.29	334.93	0.51	3609.85	1521298
14	33004	28569	220	375.97	387.26	373.94	0.54	3612.16	1209431
16	47245	36673	239	401.50	408.92	399.81	0.42	3612.97	773386
18	66181	45882	254	439.78	458.45	438.08	0.39	3613.58	462670
20	93779	56951	257	456.57	458.45	451.76	1.06	3613.94	303575

Table 2: Solution statistic for model (APP) and variants of scenario REQ_36.

τ	*#rows*	*#cols*	*#trains*	ub^*	v(LP)	v^*	gap in %	$t_\sum$ in s	*#bbn*
0	835	920	29	37.10	**37.10**	**37.10**	–	8.44	1
2	2418	2895	67	99.92	**99.92**	**99.92**	–	8.41	1
4	6920	9345	121	219.05	**219.05**	**219.05**	–	8.63	1
6	11129	16329	143	238.67	242.72	238.67	–	9.39	1
8	17393	27470	161	260.77	269.25	260.77	–	13.29	280
10	24825	41517	185	309.54	314.04	309.54	–	46.41	577
12	33156	57149	198	334.93	342.66	334.93	–	110.98	528
14	42282	74862	220	373.94	381.45	373.94	–	259.62	780
16	53142	96729	239	399.81	405.33	399.81	–	1467.77	1485
18	65378	124115	254	438.08	450.48	438.08	–	2399.55	512
20	79697	156674	257	454.77	458.30	451.76	0.67	3618.53	421

Table 3: Solution statistic for model (ACP) and variants of scenario REQ_36.

Tables 2 and 3 show the results for model (APP) and (ACP), respectively. The tables list:

- ▷ τ length of the time interval,
- ▷ $\#rows$ number of rows (constraints) of the integer programming formulation,
- ▷ $\#cols$ number of columns (variables) of the integer programming formulation,
- ▷ $\#trains$ number of scheduled trains in the solution,
- ▷ ub^* proven upper bound,
- ▷ $v(LP)$ optimal value of the linear relaxation,
- ▷ v^* objective function value of (best) integral solution,
- ▷ optimality gap[1]
- ▷ $\#bbn$ number of processed branch and bound nodes,
- ▷ and, $t_\sum$ the total running time of `TS-OPT`.

[1] The relative gap is defined between the best integer objective $bestSol$ and the objective of the best node remaining $bestNode$ as $\frac{|bestNode - bestSol|}{10^{-10} + |bestSol|}$.

instance	#*trains*	#*reqs*	#*rows*	#*cols*	ub^*	v(LP)	v^*	gap in %	$t_{\sum}$ in s	#*bbn*
req01	8	8	510	555	70.00	**70.00**	**70.00**	–	1.28	1
req02	11	11	882	676	84.01	84.14	84.01	–	0.20	1
req03	8	8	451	538	68.00	**68.00**	**68.00**	–	0.21	1
req04	19	19	1287	1197	150.53	150.83	150.53	–	0.39	1
req05	15	15	1344	877	108.89	109.42	108.89	–	0.28	4
req06	14	14	967	916	115.74	115.83	115.74	–	0.28	8
req07	42	46	5812	2949	336.09	343.49	336.09	–	5.21	1754
req08	46	55	7140	3312	357.93	372.44	357.93	–	8.09	987
req09	62	106	25957	6661	519.00	545.16	508.00	2.17	3601.01	203976
req10	73	198	76700	12525	662.55	667.55	610.23	8.57	3601.57	25673
req11	62	288	7453	2304	526.00	526.67	526.00	–	1.16	1

Table 4: Solution statistic of model (APP) for WHEEL-instances.

It turns out that, in fact, model (APP)$'$ produces for all instances a significantly weaker LP-bound (upper bounds $v(LP)$ and ub^*) than model (ACP). In addition, we marked the instances where the LP-bound at the root is equal to the objective value of the optimal integer solutions.

With increasing flexibility τ the models become trivially larger. Although the extended formulation (ACP) produces in most cases the larger model, the produced results are almost always better for this testset. Model (ACP) was able to solve all instances to optimality except for the last one. Whereas model (APP) could only solve the first six instances during the time limit. However, the reason was that the dual bound could not be significantly improved during branch and bound even if the optimal primal solutions were found. We reported more results of similar experiments with 146, 285 and 570 train requests in Borndörfer & Schlechte (2007) [30] where the same effects can be observed.

1.2 Results for the TTPlib

In addition to the HAKAFU_SIMPLE instances the TTPlib contains artificial auction instances provided by our project partners, i.e., Andreas Tanner from WIP. Figure 3 shows the layout of the infrastructure for the 11 WHEEL instances. Furthermore, station capacities are considered, as well as minimum dwell time requirements for several trains, see Chapter II, Section 2.1.2 and Chapter III, Section 1.1.

For each run of `TS-OPT`, a time limit of one hour (3600 seconds) was used to solve the IPs. Table 4 and Table 5 show the results of the static models (APP)$'$ and (ACP).

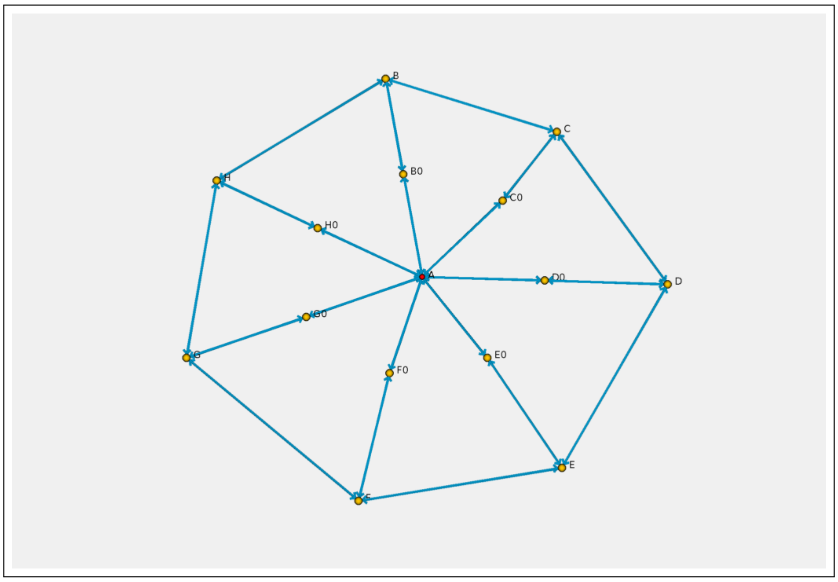


Figure 3: Artifical network WHEEL, see TTPlib [208]

instance	$\#trains$	$\#reqs$	$\#rows$	$\#cols$	ub^*	$v(\mathrm{LP})$	v^*	gap in %	$t_{\sum}$ in s	$\#bbn$
req01	8	8	1119	1202	70.00	**70.00**	**70.00**	–	1.88	1
req02	11	11	1273	1364	84.01	**84.01**	**84.01**	–	0.43	1
req03	8	8	1104	1175	68.00	**68.00**	**68.00**	–	0.27	1
req04	19	19	2351	2514	150.53	**150.53**	**150.53**	–	0.43	1
req05	15	15	1596	1706	108.89	**108.89**	**108.89**	–	0.34	1
req06	14	14	1816	1945	115.74	**115.74**	**115.74**	–	0.30	1
req07	42	46	5151	5512	336.09	**336.09**	**336.09**	–	0.66	1
req08	46	55	5747	6133	357.93	**357.93**	**357.93**	–	0.70	1
req09	62	106	9854	10553	508.00	**508.00**	**508.00**	–	1.29	1
req10	73	198	16263	17512	614.77	**614.77**	**614.77**	–	2.23	1
req11	62	288	6353	4912	526.00	526.67	526.00	–	1.96	1

Table 5: Solution statistic of model (ACP) for WHEEL-instances.

Obviously, model (ACP) has more variables than model (APP)′ because of the auxiliary track flows. But if the conflict constraints of the instance "explode" model (ACP) has significantly less rows than (APP)′, e.g., in case of instances REQ_07-REQ_10.

CPLEX was able to solve all 11 instances of model (ACP) to optimality already in the root node (in only some seconds!). In addition in 10 of 11 cases the value of the LP-relaxation equals the optimal value of the integer problem. In contrast, (APP)′ was only able to solve 9 problems within the time limit. For scenario REQ_09 and REQ_10 only a gap of approximately 2% and 8% were reached after 1 hour. Only in two cases the value of the LP-relaxation equals the optimal value of the integer problem. In addition, CPLEX needs to solve model (APP)′ a significant number of branch and bound nodes for 6 instances.

instance	#trains	#reqs	#rows	#cols	ub^*	v(LP)	v^*	gap in %	$t_\sum$ in s	#bbn
req01	198	285	3400	2563	393.72	**393.72**	**393.72**	–	11.86	1
req02	266	285	28810	19694	461.54	505.64	457.25	0.94	3612.41	1156299
req03	273	285	62908	35021	486.60	507.71	484.78	0.38	3614.86	367354
req04	285	285	349241	97135	512.37	512.51	511.95	0.08	3687.18	58421
req05	152	194	2216	1764	288.00	**288.00**	**288.00**	–	8.99	1
req06	204	213	17780	14512	348.92	370.55	348.92	–	40.62	8207
req07	178	184	33607	23450	331.41	336.23	327.82	1.10	3609.58	705190
req08	199	199	182442	68342	374.01	374.16	373.92	0.02	3639.15	147562
req09	93	114	1369	1112	166.82	**166.82**	**166.82**	–	7.93	1
req10	104	109	8147	7699	202.88	218.18	202.88	–	5.56	765
req11	97	98	12455	11902	162.40	162.44	162.40	–	10.67	2116
req12	113	113	66011	39167	245.33	245.37	245.33	–	37.73	906
req13	28	28	336	308	59.46	**59.46**	**59.46**	–	2.62	1
req14	33	33	1879	2544	69.53	**69.53**	**69.53**	–	2.41	1
req15	31	31	3406	4477	46.08	**46.08**	**46.08**	–	1.02	9
req16	30	30	9281	9436	91.62	91.64	91.62	–	2.47	352
req17	215	285	2417	1929	393.30	397.23	393.30	–	17.31	1
req18	274	285	28827	19638	482.91	504.94	474.59	1.75	3616.93	1460850
req19	278	285	62994	35116	488.32	507.88	488.31	–	3621.81	555884
req20	285	285	346438	99306	512.59	512.65	512.49	0.02	3699.88	90655
req21	170	209	1676	1382	296.92	**296.92**	**296.92**	–	6.87	1
req22	206	212	18394	14121	337.96	349.73	337.96	–	22.43	4057
req23	191	199	41456	26132	358.50	378.12	358.49	–	3622.67	892094
req24	194	194	184853	68282	371.86	371.93	371.86	–	1739.29	57430
req25	98	117	959	822	175.56	**175.56**	**175.56**	–	6.78	1
req26	117	118	8604	7952	191.75	192.54	191.75	–	5.17	181
req27	116	118	16268	13981	185.46	198.15	185.46	–	95.62	49763
req28	102	102	63468	35804	191.32	191.37	191.32	–	1473.04	213086
req29	20	20	154	144	40.71	**40.71**	**40.71**	–	3.47	1
req30	31	31	1439	1835	100.06	**100.06**	**100.06**	–	0.90	1
req31	363	1062	16844	15620	464.40	464.75	464.40	–	62.70	182
req32	261	1140	106091	44112	202.85	214.58	202.85	–	53.74	486
req33	151	570	34911	22056	105.33	109.86	105.33	–	25.29	505
req36	151	285	5907	5712	242.58	255.34	242.58	–	10.19	556
req37	257	334	201529	82937	365.73	371.25	365.73	–	190.72	523
req38	259	334	201529	82937	398.77	405.87	398.77	–	109.01	549
req39	272	358	245968	73324	505.18	521.02	505.18	–	492.91	6259
req40	272	358	245968	73324	541.10	556.99	534.43	1.25	3630.77	4650
req41	287	382	106728	56037	419.11	427.16	419.11	–	66.34	544
req42	288	382	106728	56037	442.27	453.22	442.27	–	75.16	555
req43	300	409	247756	87209	531.44	553.59	531.44	–	2226.43	38981
req44	300	409	247756	87209	554.97	585.29	554.97	–	3378.26	83002
req45	264	344	141976	51079	514.90	523.11	514.90	–	956.05	1852
req46	263	344	141976	51079	462.73	472.75	462.73	–	168.23	5000
req47	25	25	2304	3105	43.63	**43.63**	**43.63**	–	2.27	1
req48	41	41	11585	13314	76.81	**76.81**	**76.81**	–	5.54	1

Table 6: Solution statistic of model (APP) for HAKAFU_SIMPLE-instances.

We also performed this experiment for the remaining instances of the TTPlib, i.e., 50 instances for network HAKAFU_SIMPLE. The results of the experiment are shown in Table 6 and 7. For four instances CPLEX, i.e., REQ_34,REQ_35, REQ_49 and REQ_50, was not able to solve the integer program within 1 hour for both models. For the remaining 46 instances model (ACP) reached three times the time limit without any solution. For another three instances `TS-OPT` terminates for model (ACP) with a small optimality gap of approximately 1%. CPLEX was able to solve all other instances (40) to proven optimality. In addition, we marked 16 instances were the objective values of the LP relaxation for model (ACP) coincide with optimal integer solution.

CPLEX was able to produce solutions for model (APP)′ for all 46 instances, i.e., also for instances REQ_39, REQ_43, and REQ_44 within the time limit. However, in 8 cases the runs terminated after an hour

instance	$\#trains$	$\#reqs$	$\#rows$	$\#cols$	ub^*	v(LP)	v^*	gap in %	$t_\sum$ in s	$\#bbn$
req01	198	285	6880	9315	393.72	**393.72**	**393.72**	–	13.66	1
req02	266	285	37487	59637	457.25	483.48	457.25	–	46.61	518
req03	274	285	61607	105243	485.27	505.99	485.27	–	2419.21	13947
req04	284	285	153226	302292	512.14	512.20	508.78	0.66	3644.78	779
req05	152	194	4845	6330	288.00	**288.00**	**288.00**	–	9.98	1
req06	204	213	28212	44623	348.92	365.57	348.92	–	24.97	529
req07	178	184	41997	70415	327.82	336.23	327.82	–	41.17	527
req08	199	199	111069	216059	373.92	374.02	373.92	–	2272.67	1582
req09	93	114	3070	3795	166.82	**166.82**	**166.82**	–	8.72	1
req10	104	109	15209	22972	202.88	207.22	202.88	–	5.43	49
req11	97	98	21656	33715	162.40	166.55	162.40	–	11.25	622
req12	113	113	68197	130143	245.33	245.35	245.33	–	83.75	482
req13	28	28	916	915	59.46	**59.46**	**59.46**	–	2.88	1
req14	33	33	5061	6613	69.53	**69.53**	**69.53**	–	2.46	1
req15	31	31	8521	11935	46.08	**46.08**	**46.08**	–	1.34	1
req16	30	30	16894	26468	91.62	91.63	91.62	–	5.01	99
req17	215	285	5361	7318	393.30	394.92	393.30	–	17.64	1
req18	274	285	38118	62658	474.59	495.79	474.59	–	27.75	104
req19	278	285	63662	112602	488.31	498.03	488.31	–	163.09	531
req20	284	285	161313	329062	512.55	512.57	509.18	0.66	3651.25	840
req21	170	209	3909	5078	296.92	**296.92**	**296.92**	–	8.00	1
req22	206	212	27657	44998	337.96	346.90	337.96	–	21.07	351
req23	191	199	48054	85452	358.49	368.19	358.49	–	86.96	542
req24	194	194	115319	235679	371.86	371.91	371.86	–	3493.65	6446
req25	98	117	2395	3047	175.56	**175.56**	**175.56**	–	7.43	1
req26	117	118	15725	24117	191.75	**191.75**	**191.75**	–	5.65	1
req27	116	118	26196	44151	185.46	198.13	185.46	–	23.91	542
req28	102	102	62018	121071	191.32	191.35	191.32	–	882.22	6357
req29	20	20	479	472	40.71	**40.71**	**40.71**	–	1.80	1
req30	31	31	3797	4822	100.06	**100.06**	**100.06**	–	1.28	1
req31	368	1062	31754	43710	464.40	**464.40**	**464.40**	–	69.10	30
req32	297	1140	80183	126924	202.85	**202.85**	**202.85**	–	60.63	1
req33	171	570	42416	67443	105.33	**105.33**	**105.33**	–	26.13	1
req36	151	285	11855	16392	242.58	246.64	242.58	–	10.99	1
req37	257	334	130148	265556	365.73	366.74	365.73	–	3653.02	9
req38	259	334	130148	265556	398.77	401.44	398.77	–	3166.27	1
req39	0	358	114397	226407	507.54	507.54	–	∞	3642.14	1
req40	272	358	245968	73324	541.07	556.99	534.43	1.24	3631.86	4723
req41	287	382	97282	202892	419.11	422.47	419.11	–	1527.14	545
req42	288	382	97282	202892	442.27	446.56	442.27	–	2228.27	494
req43	0	409	140963	303446	538.55	538.55	–	∞	3651.79	1
req44	0	409	140963	303446	563.85	563.85	–	∞	3651.04	1
req45	264	344	85629	171420	514.90	516.25	514.90	–	744.86	149
req46	263	344	85629	171420	462.73	465.85	462.73	–	1252.49	223
req47	25	25	6163	8272	43.63	**43.63**	**43.63**	–	2.68	1
req48	41	41	24124	40722	76.81	**76.81**	**76.81**	–	8.21	1

Table 7: Solution statistic of model (ACP) for HAKAFU_SIMPLE-instances.

with an optimality gap of approximately 1%. The produced solution were already the optimal ones, nevertheless (APP)′ was not able to close the gap within the time limit. The other 38 instances were solved to optimality. In 12 cases the objective values of the LP relaxation for model (APP)′ coincide with optimal integer solution.

We increased the time limit to one day and solved again the hard instances. Let us explicitly point out that these computations would not be possible on a standard PC at the beginning of the project. However, thanks to the 16GB main memory we were able to produce these numbers to verify our novel algorithmic approach which will be discussed in the next section.

Tables 8 and 9 show the results for both models. For instances REQ_34, REQ_35, REQ_49, and REQ_50 the LP relaxation of model (APP)′ became too large, i.e., CPLEX abort with out of memory. The other

instance	#trains	#reqs	#rows	#cols	ub*	v(LP)	v*	gap in %	$t_{\sum}$ in s	#bbn
req34	0	285	3623973	305366	–	–	memout	∞	935.83	–
req35	0	285	7974708	514425	–	–	memout	∞	1100.57	–
req37	257	334	201529	82937	365.73	371.25	365.73	–	401.52	523
req38	259	334	201529	82937	398.77	405.87	398.77	–	258.78	549
req39	272	358	245968	73324	505.18	521.02	505.18	–	862.91	6259
req40	272	358	245968	73324	535.32	556.99	535.32	–	47203.67	28766
req43	300	409	247756	87209	531.44	553.59	531.44	–	5245.88	38981
req44	300	409	247756	87209	554.97	585.29	554.97	–	7604.45	83002
req49	0	285	2152600	232204	–	–	memout	∞	177.84	–
req50	0	285	7974708	514425	–	–	memout	∞	282.70	–

Table 8: Solution statistic of model (APP) for hard HAKAFU_SIMPLE-instances.

instance	#trains	#reqs	#rows	#cols	ub*	v(LP)	v*	gap in %	$t_{\sum}$ in s	#bbn
req34	0	285	384563	873904	512.67	512.67	–	∞	86478.83	1
req35	0	285	587570	1394454	512.75	512.75	–	∞	86540.46	1
req37	257	334	130148	265556	365.73	366.74	365.73	–	3683.60	10
req38	259	334	130148	265556	398.77	401.44	398.77	–	2448.81	1
req39	272	358	114397	226407	505.18	507.54	505.18	–	6901.22	174
req40	272	358	114397	226407	535.32	541.55	535.32	–	17688.08	645
req43	300	409	140963	303446	531.44	538.55	531.44	–	18162.80	553
req44	300	409	140963	303446	554.97	563.85	554.97	–	27764.54	624
req49	0	285	311772	703252	504.68	504.68	–	∞	86493.50	1
req50	0	285	587570	1394454	512.75	512.75	–	∞	86744.36	1

Table 9: Solution statistic of model (ACP) for hard HAKAFU_SIMPLE-instances.

instances could be solved to optimality within an hour. In contrast to that, CPLEX was able to solve all relaxations of model (ACP) within one day and produced stronger upper bounds for all hard scenarios. However, CPLEX needed more time producing an optimal integer solution for model (ACP) than for model (APP)′ for almost all hard instances. Although CPLEX needs less branch and bound nodes to solve model (ACP), the time needed per node, i.e., to solve the linear relaxation, was significantly higher than for model (APP)′.

1.3 Conclusion

We have compared the static model formulation (APP)′ and (ACP) for a huge set and variants of instances which are free available at TTPlib. First of all, CPLEX was able to solve model (APP)′ and (ACP) for instances of reasonable size to proven optimality, i.e., `TS-OPT` was only used to construct the (preprocessed) graphs and models. Only for some very large scale instances the larger LP relaxation of the extended formulation had a negative effect on the total running time. We have observed that even if the extended formulation (ACP) tends in most cases to larger LP relaxations than (APP)′, the benefit from a better global upper bound transfers often directly to a higher solution quality and shorter running times. In particular, these effects are in-

model	(APP)	(ACP)
	REQ_36-instances	
# produced best upper bound	2	11
# no integrality gap	2	3
# optimal solution found	11	11
# optimal solution proven	6	10
	WHEEL-instances	
# produced best upper bound	3	11
# no integrality gap	2	10
# optimal solution found	10	11
# optimal solution proven	9	11
	HAKAFU_SIMPLE-instances	
# produced best upper bound	25	50
# no integrality gap	12	16
# optimal solution found	38	40
# optimal solution proven	39	40

Table 10: Comparison of results for differrent models on the TTPlib-instances.

tensified if the flexibility of the train requests are high, e.g., if the time windows of the events are large, or if the capacity is rare, e.g., if several trains compete for the same track resources.

The results of our computational experiments made us conclude that model (ACP) outperforms model (APP)′. Table 10 gives a short summary and lists the number of instances for which the models produced an optimal solution, number of instances for which the root upper bound has no integrality gap, and the number of instances for which the upper bound of the root LP relaxation was better or equal than the one produced by the other model. If we would establish a system of point scoring, model (ACP) will be most likely the winner on “points”. Hence, (ACP) is suited best for our particular problem instances and real world application.

2 Algorithmic Ingredients for the (PCP)

In this section we want to analyze our different solution approaches to solve model (PCP), which we all integrated or implemented in our module TS-OPT. We start with a comparison of our approach with computational results from the literature in Section 2.1. Section 2.2 discusses experiments and results for the bundle method. Finally, we present computational results of the rapid branching heuristic to solve large scale track allocation problems in Section 2.3.

2.1 Results from the Literature

Let us discuss computational results for a variation of the rather "simple" WHEEL instances. The reason is that Cacchiani, Caprara & Toth (2010) [54] present results for modified versions of these TTPlib instances by excluding station capacities. In addition, their implemented model cannot handle train type specific headway times. Hence, they only considered instances of the TTPlib with one train type, i.e., the WHEEL instances. However, let us thank them (and all others) for using our instances in their studies, which verifies that the TTPlib provides an useful, modular, and easily understandable standard format for track allocation problems.

They used a (PPP) formulation of the problem, produced upper bounds by solving the Lagrangian relaxation using standard subgradient optimization and column generation, and constructed solutions by a greedy heuristic based on Lagrangian profits and some refinement procedure. They were able to solve instances REQ_1-REQ_8 to proven optimality within a second. For instances REQ_9-REQ_11 they could produce almost optimal solutions, i.e., the produced upper bounds prove a gap within 2% of the optimum. The time needed to produce solutions for problem REQ_9 and REQ_10 is comparatively high (57 and 602 seconds), as well as we already observed for the static model (APP), see Section 1 and Table 4. However, in 5 of 11 cases the presented solutions are also feasible (and hence optimal) in presence of the station capacity constraints.

Table 11 lists the statistic of our column generation approach using the bundle method and the rapid branching heuristic. We want to mention that our listed absolute values (bounds and objectives) differ to the published ones on TTPlib due to a problem specific scaling inside of TS-OPT. In fact, we scale all objective values such that the best path has profit of 10.0. Furthermore, we used as a stopping criteria an optimality gap of 1.0%. It can be seen that we only need a very small number of branch and bound nodes to produce almost optimal solutions (gap

instance	#trains	#reqs	#rows	#cols	ub^*	v(LP)	v^*	gap in %	$t_\sum$ in s	#bbn
req01	8	8	235	369	70.00	**70.00**	**70.00**	–	0.41	1
req02	11	11	253	594	84.01	**84.01**	**84.01**	–	0.20	1
req03	8	8	237	403	68.00	**68.00**	**68.00**	–	0.13	1
req04	19	19	474	1149	150.67	**150.67**	**150.67**	–	0.30	1
req05	15	15	304	972	108.92	108.92	108.89	0.03	0.39	3
req06	14	14	375	765	115.80	115.80	115.74	0.05	0.33	2
req07	42	46	919	3587	337.22	337.22	337.16	0.02	2.07	3
req08	46	55	1014	4296	359.44	359.44	359.36	0.02	5.90	3
req09	62	106	1422	6173	511.28	511.28	511.20	0.02	6.32	3
req10	77	198	1879	8645	644.68	644.68	644.61	0.01	17.32	2
req11	66	288	1176	3014	556.16	556.16	556.00	0.03	29.03	6

Table 11: Solution statistic of `TS-OPT` and model (PCP) for WHEEL-instances.

below 0.05%). However, the re-scaled upper bounds and solutions are conform to the results presented by Cacchiani, Caprara & Toth (2010) [54]. There are minor deviations for the solutions values because are numerical ones respecting the given tolerances, see Table 11.

To demonstrate that even such small instances have to be solved via exact optimization approaches, we only run the bundle method to solve the relaxation and used afterwards a simple greedy heuristic in `TS-OPT` to produce a feasible integral solution. It can be seen that even for these simple instances it is not trivial to produce high quality solutions. For some of the instance the produced solutions have a gap larger than 15% to the optimum. Finally, Table 12 compares the (PPP)-results of Cacchiani, Caprara & Toth (2010) [54], the (bundle and) greedy approach, and the (bundle and) rapid branching approach to solve model (PCP) with `TS-OPT`. Already this rather easy subset of the TTPlib indicates that our configuration model has computational advantages, both the static variant (ACP), see Section 1 and Table 4, and dynamic version (PCP). In particular, if the instance give rise to many conflicts, e.g. instances REQ_9 and REQ_10.

	bundle and greedy			(PPP)			TS-OPT		
instance	v^*	gap in %	$t_\sum$ in s	v^*	gap in %	$t_\sum$ in s	v^*	gap in %	$t_\sum$ in s
req01	3500.00	–	1	3500.00	–	1	3500.00	–	1
req02	3600.00	16.7	1	4221.02	–	1	4200.50	–	1
req03	3400.00	–	1	3400.00	–	1	3400.00	–	1
req04	7533.50	–	1	7533.29	–	1	7533.50	–	1
req05	4733.50	15.1	1	5456.78	–	1	5444.50	–	1
req06	5786.00	0.1	1	5787.24	–	1	5787.00	–	1
req07	14615.50	15.4	2	16910.72	–	2	16858.00	–	2
req08	16528.00	8.7	2	17957.08	–	17	17968.00	–	6
req09	23869.00	7.1	4	26046.44	1.8	57	26046.00	–	6
req10	30391.00	6.1	9	32299.96	1.2	602	32230.50	–	17
req11	25800.00	7.8	16	27800.00	0.6	8	27800.00	–	29

Table 12: Comparison of results for model (PPP) from Cacchiani, Caprara & Toth (2010) [54] for modified WHEEL-instances.

2.2 Bundle Method

We evaluated our algorithmic approaches presented in Chapter III, Section 3 on the benchmark library TTPlib, see Erol et al. (2008) [85]. They are associated with the macroscopic railway network model HAKAFU_SIMPLE already described in Section 1.

Figure 4 illustrates the column generation process for solving instance REQ_05 with the barrier method of CPLEX. For each iteration the current value of the RMLP is shown as well as the upper bound $\beta(\gamma, \pi, \lambda)$, see Lemma 2.11. The general effects of "heading in" and "tailing off" can be observed, i.e., we need many column generation iterations to get an upper bound value of 289. Obviously, one could try to improve the performance or convergence of a standard column generation approach by using stabilization techniques or sophisticated strategies for the generation of columns, see Lübbecke & Desrosiers (2005) [156].

Figure 5 shows exemplary the progress of the bundle method 5, i.e., it can be seen that a dual bound of 289 is reported after one second. Together with Figure 4 it gives an intuition of the progress and convergence of the bundle method and the standard column generation approach for solving instance REQ_05. The mere fact that the time-scales are significantly different prevent us from plotting both runs together. The reason for the significant smaller solution time is that in case of the bundle method in each iteration only a very small QP and

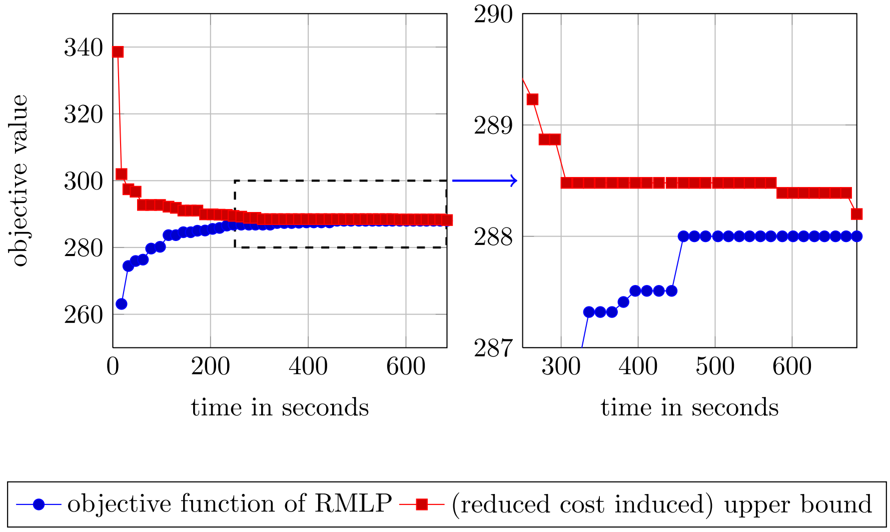


Figure 4: Solving the LP relaxation of model (PCP) with column generation and the barrier method.

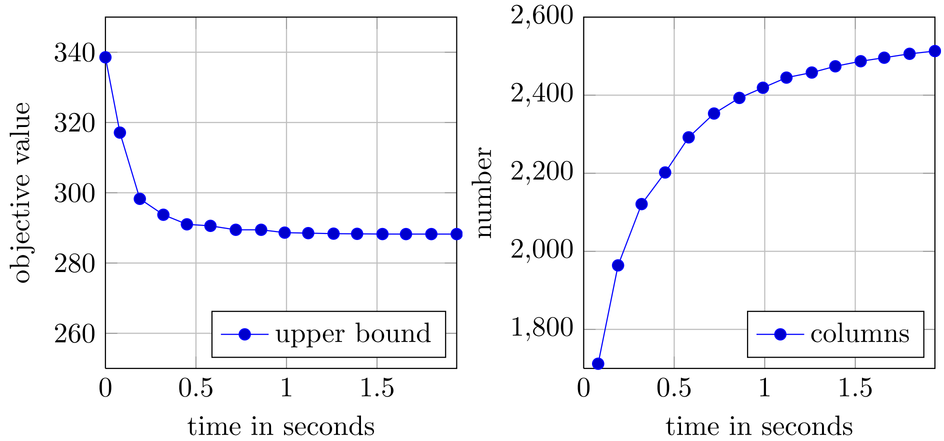


Figure 5: Solving the LP relaxation of model (PCP) with the bundle method.

several shortest path problems are successively solved. In case of the column generation approach with the barrier method, as well as with the primal or dual simplex method, solving large linear programs and also solving shortest path problems are alternated.

Table 13 compares different solution approaches to solve the linear or Lagrangean relaxation of model (PCP) for an arbitrary selection of request scenarios of network HAKAFU_SIMPLE. On the one hand, we solve the linear relaxation by column generation and by using different algorithms to solve the LP relaxation, i.e., the rows "dual" contain the results of the dual simplex algorithm, "barrier" stands for barrier algorithm, and "primal" for the primal simplex algorithm. On the other hand, the rows "bundle" show the results for the bundle method. The sizes, i.e., $\#reqs$, $\#rows$, and $\#cols$, of the finally generated models are listed, as well as the solution time $t_{\sum}$. Column ub^* shows the value of the upper bound $\beta(\gamma, \pi, \lambda)$ induced by the reduced cost during the column generation method, see Lemma 2.11 or the best upper bound produced by the Lagrangean relaxation. Column $v^*(LP)$ states the value of the produced fractional primal solution. We mark this value in case of the bundle method, because the produced fractional vector might violate the relaxed constraints, i.e., the coupling constraints of model (PCP).

We can observe that the standard column generation approach for solving LPs needs much more columns until the relaxation is solved to optimality for most of the instances. In each iteration a noticeable larger LP is solved. The number of column generation iterations ($\#iter$) is very high, i.e., several hundreds, if we solve the (MLP) to proven op-

solver	#*reqs*	#*rows*	#*cols*	*ub**	v^*(LP)	$t_\sum$ in *s*	#*iter*
				REQ_02			
primal	285	7914	138450	488.06	482.41	> week	761
dual	285	7914	147831	487.22	482.77	> day	1000
barrier	285	7914	145146	489.29	482.77	> 4hours	1000
bundle	285	7914	146415	484.13	*484.13*	449	1514
				REQ_05			
primal	194	1157	36691	288.04	287.81	454	116
dual	194	1157	37087	288.00	288.00	566	187
barrier	194	1157	37448	288.20	288.00	683	230
bundle	194	1157	2521	288.24	*288.24*	2	157
				REQ_17			
primal	285	1393	24185	395.29	394.92	450	130
dual	285	1393	25344	395.15	394.83	616	241
barrier	285	1393	25901	395.12	394.92	842	330
bundle	285	1393	3692	395.29	*395.29*	35	234
				REQ_21			
primal	209	1032	14932	296.92	296.83	242	179
dual	209	1032	13886	296.92	296.69	148	86
barrier	209	1032	14274	297.15	296.92	287	184
bundle	209	1032	1991	297.28	*297.28*	25	142
				REQ_25			
primal	117	645	6058	175.56	175.56	107	229
dual	117	645	5410	175.56	175.35	29	38
barrier	117	645	5433	175.56	175.56	83	36
bundle	117	645	1268	175.73	*175.73*	14	122

Table 13: Statistic for solving the LP relaxation of model (PCP) with column generation and the bundle method.

timality. That is no column with positive reduced cost is left. Besides the higher memory consumption for the larger LPs, we observed a convergence problem with the primal and dual simplex as well as with the barrier method.

In contrast, the bundle method solves the relaxation (RMLP) in an algorithmically integrated and sparse way. No "real" column generation is needed because the function evaluation step of algorithm 5 can be solved exactly. Only in the direction finding step the generated paths and configurations are used. However, the produced solutions of the shortest path problems can be seen as generated columns of the bundle method, i.e., these are the columns that we store during the bundle algorithm to construct a restricted version of model (PCP) and produce an integral solution in the end. In addition, we keep also the paths and configurations induced by columns that leave the bundle set during the algorithm.

Therefore, the generation of columns seems to be more guided and only a small portion of the paths and configurations compared with the other approaches is needed to solve the relaxation, see Figure 5 and, Table 13. The very large instance REQ_02 is one of a few exceptions, for which the bundle method also needs a comparable high number of columns similar to the other approaches. However, the solution time is always significantly smaller without losing quality. In case of REQ_02, the column generation approach is stopped after a fixed limit of 1000 iterations with a bound even worse than produced by the bundle approach.

For our type of problem, i.e., the Lagrangean dual of model (PCP), the parameter calibration of the the bundle method was rather uncomplicated and straight-forward. Figure 6 compares exemplary the effect of different choices for the size of the bundle (2, 5, 10, 15, 20, 25) on the solution of the Lagrangean relaxation of some test instances. It can be seen that larger bundles lead in general to a reduction in the number of iterations to a certain limit. However, larger bundles also produce larger and more difficult quadratic programs in algorithm 5, such that the total solution time and the number of iterations increases after a certain point. A default bundle size of 15 seems to be a good choice for our specific problem instances.

Table 14 shows the results of our implementation of the bundle method on solving the Lagrange relaxation of the the model (PCP). Additional to the columns we have already introduced in former tables, column *#iter* displays the number of iterations of the bundle method to solve the Lagrangean relaxation, see algorithm 5. We denoted the optimal value of the Lagrangean dual (LD) by $v^*(\text{LD})$. After that, we performed a trivial greedy heuristic to find an integer solution for the

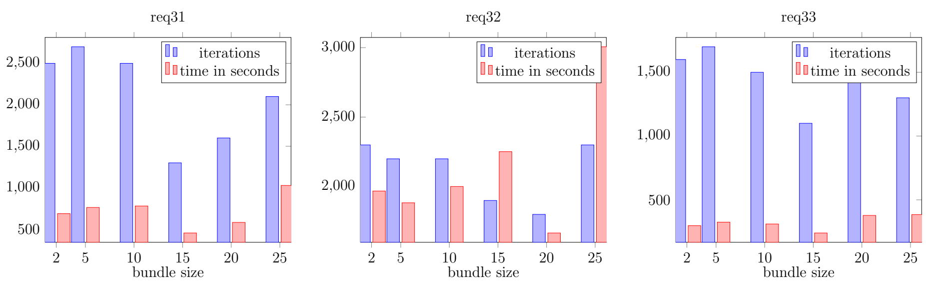


Figure 6: Testing different bundle sizes.

instance	*#trains*	*#reqs*	*#rows*	*#cols*	v^*(LD)	v^*	gap in %	$t_{\sum}$ in s	*#iter*
req01	197	285	1618	4613	393.95	393.55	0.10	18.22	214
req02	207	285	7914	146415	484.13	362.13	33.69	935.63	1514
req03	224	285	12848	202773	507.09	392.63	29.15	1988.87	1540
req04	208	285	31615	138989	512.37	385.29	32.98	1353.57	320
req05	152	194	1157	2521	288.24	288.00	0.08	11.35	157
req06	175	213	6032	118056	366.31	312.89	17.07	528.08	1866
req07	158	184	8878	154847	336.41	295.11	13.99	877.83	1865
req08	155	199	23308	68767	374.31	282.15	32.66	289.63	120
req09	93	114	746	1392	166.99	166.82	0.10	9.02	119
req10	98	109	3303	48171	207.48	194.11	6.89	67.94	753
req11	78	98	4633	15224	166.62	134.65	23.74	13.19	65
req12	95	113	14856	31302	245.39	194.35	26.26	64.58	68
req13	28	28	244	321	59.46	59.46	–	3.39	14
req14	32	33	1133	2125	69.53	69.30	0.34	2.86	9
req15	30	31	1909	3409	46.09	43.43	6.12	1.58	10
req16	29	30	3759	5148	91.65	87.20	5.10	2.64	12
req17	211	285	1393	3692	395.29	389.78	1.41	22.81	234
req18	220	285	8218	140206	496.05	360.90	37.45	861.04	1520
req19	216	285	13576	45652	512.72	341.27	50.24	180.73	145
req20	194	285	34094	70786	512.77	312.09	64.30	568.75	122
req21	168	209	1032	1991	297.28	295.41	0.63	8.93	142
req22	173	212	6003	20820	355.02	262.32	35.34	36.46	92
req23	150	199	10370	28203	381.18	280.11	36.08	62.84	80
req24	140	194	24925	38751	372.00	255.94	45.34	125.11	45
req25	98	117	645	1268	175.73	175.56	0.10	8.08	122
req26	101	118	3460	8396	193.34	162.03	19.32	7.10	32
req27	85	118	5692	11737	198.18	120.97	63.82	12.60	23
req28	81	102	13612	18026	191.38	149.19	28.28	25.07	19
req29	20	20	145	189	40.71	40.71	–	2.26	5
req30	30	31	867	1709	100.06	96.39	3.80	1.57	8
req31	352	1062	6913	28318	464.78	458.02	1.48	347.06	828
req32	292	1140	16489	28191	203.05	192.62	5.42	1196.45	752
req33	171	570	9036	12566	105.69	100.78	4.87	159.85	459
req34	149	285	76842	138994	512.75	310.90	64.92	2366.24	108
req35	137	285	116303	49772	512.77	243.78	110.34	1677.37	44
req36	127	285	2602	28385	247.00	178.23	38.58	99.10	684
req37	169	334	28694	133626	367.10	221.61	65.65	2462.74	864
req38	167	334	28694	145328	401.65	249.44	61.02	2882.28	1090
req39	142	358	24329	158428	507.89	198.83	155.44	3694.50	1112
req40	140	358	24329	176134	541.89	251.92	115.10	3929.45	1331
req41	144	382	22035	135959	422.67	213.62	97.86	2621.83	1140
req42	134	382	22035	138510	446.81	194.58	129.63	2626.67	1225
req43	151	409	30978	170834	538.79	294.09	83.21	4613.30	1204
req44	154	409	30978	176552	564.14	274.85	105.25	4556.10	1177
req45	163	344	18694	112021	516.49	339.18	52.28	1449.44	1137
req46	151	344	18694	112122	466.09	259.29	79.76	1595.59	1122
req47	24	25	1402	2212	43.63	43.42	0.49	2.97	8
req48	39	41	5456	5567	76.81	71.71	7.12	6.48	9
req49	139	285	63963	105681	512.74	300.85	70.43	1860.76	120
req50	137	285	116303	49772	512.77	243.78	110.34	1673.98	44

Table 14: Solution statistic of bundle method and greedy heuristic for model (PCP) for HAKAFU_SIMPLE-instances.

constructed sub-problems. The objective value is denoted by v^* in Table 14.

We could observe that the upper bounds produced by our bundle implementation for model (PCP) have the same quality as the ones obtained by model (ACP), i.e., better bounds than model (APP)′. There are only slight differences because of the numerical tolerances. In addition, the bundle approach and model (PCP) is faster than static models for very large scale instances, e.g., REQ_40, REQ_49 or REQ_50. In addition, solving the static models (ACP) and (APP) for instances REQ_34, REQ_35, and REQ_50 is critical from a memory point of view. At least 16GB of main memory is required to solve the root relaxtion.

In contrast to that our bundle approach uses only 2 GB of memory to solve the relaxtion of these instances.

However, for the produced integer solutions of the greedy heuristic no solution quality can be guaranteed. Obviously, there are easy instances, e.g., REQ_01,REQ_05,REQ_09,REQ_13 or REQ_29, where a greedy approach is able to produce an optimal or almost optimal solution. But there are also many instances for which the greedy solution is far away from optimality, e.g., REQ_39, REQ_40 or REQ_42 have a gap larger than 100%.

Finally, we conclude that the bundle method is the most efficient approach to produce high quality upper bounds for model (PCP). It outperforms standard column generation approaches using the simplex or interior point methods, i.e., the total running time is order of magnitudes smaller and the quality of the upper bounds is roughly comparable. Furthermore, we were able with this approach to produce non-trivial upper bounds much faster than with the static model variant (ACP) for very large scale instances.

2.3 Rapid Branching

We tested our implementation of the rapid branching heuristic, see Algorithm 6 presented in Section 3 of Chapter III, on instances from the benchmark library TTPlib, see the macroscopic railway network model HAKAFU_SIMPLE described in Section 1.1, and some larger request sets.

Figure 8 shows an ideal run of our code `TS-OPT`, i.e., the run of scenario REQ_31 and network HAKAFU_SIMPLE. On the left hand side the objective value of the primal solution, the upper bound, and the objective of the fixation evaluated by the rapid branching heuristic is illustrated. In the initial LP stage (dark blue), a global upper bound is computed by solving the Lagrangean dual using the bundle method after approximately 400 seconds. In that scenario one can see the improvement of the upper bound during the bundle method. Furthermore, in that stage the most important path and configuration variables are generated. On the right hand side of the figure, the development of the number of generated columns, the number of fixed to 1 columns, and the number of integer infeasibilities, i.e., the number of integer variables that still have a fractional value in the primal solution of the current relaxation, is shown. In the first phase (dark blue) the column

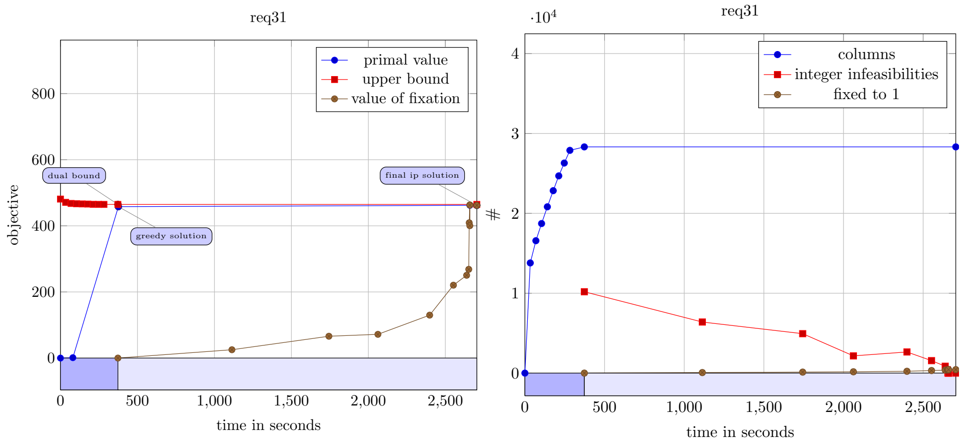


Figure 7: Solving a track allocation problem with `TS-OPT`; dual (LP) and primal (IP) stage.

generation process during the bundle method can be seen and that fixing a large number of the "right" variables at once (to 1) decreases the integer infeasibilities significantly, but not monotonously. In fact, the rapid branching heuristic produced a solution with 0.61% and was able to improve the greedy solution computed directly after the first phase with a gap of 1.48%.

Figure 8 shows another run of our code `TS-OPT`, i.e., scenario REQ_48 of network HAKAFU_SIMPLE. On the left hand side the objective value of the primal solution, the upper bound, and the objective of the fixation evaluated by the rapid branching heuristic is plotted again. In the initial LP stage (dark blue), a global upper bound is computed by solving the Lagrangean dual using the bundle method after approximately 15 seconds. In that scenario the upper bound is only slightly below the trivial upper bound, i.e., the sum of all maximum profits. In the succeeding IP stage (light blue) an integer solution is constructed by the greedy heuristic and improved by the rapid branching heuristic. It can be seen that the final integer solution has virtually the same objective value as the LP relaxation and the method is able to close the gap between greedy solution and the proven upper bound. On the right hand side of the figure, one can see that indeed often large numbers of variables are fixed to one and several backtracks are performed throughout the course of the rapid branching heuristic until the final solution was found. In addition, we plotted the development of the integer infeasibilities, i.e., the number of integer variables that still have a fractional value.

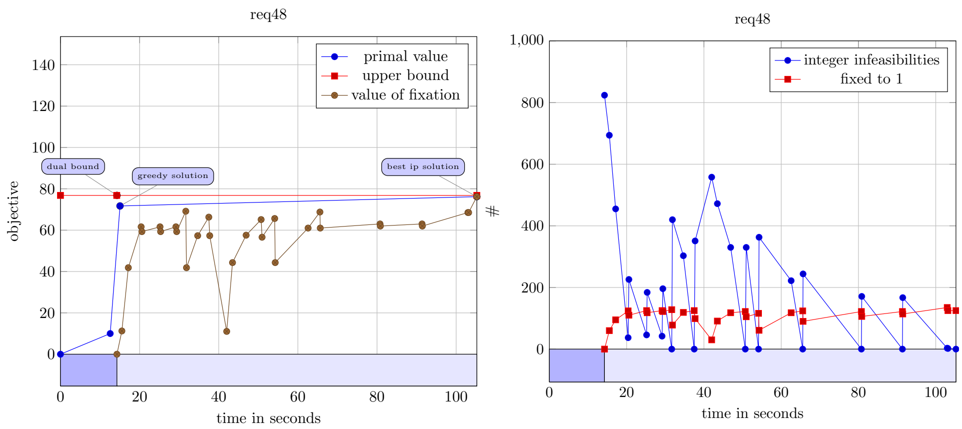


Figure 8: Solving track allocation problem REQ_48 with `TS-OPT`.

Tables 15, 16 and 17 show results for solving the test instances by our code `TS-OPT` in order to calibrate our method. Furthermore, we set a limit on the number of backtrack for rapid branching of 5. The tables list the number of scheduled trains in the best solution found, the number of requested train, the size of the model in terms of number of rows and columns, the upper bound produced by the bundle method, the solution value of rapid branching heuristic, the optimality gap, the total running time in CPU seconds, and the number of (rapid) branching nodes. The computations in Table 15 have been performed with an aggressive choice of the rapid branching integrality tolerance of $\epsilon = 0.4$; Table 17 shows the results for a cautious choice of $\epsilon = 0.1$ and Table 17 for the default choice of $\epsilon = 0.25$. It can be seen that the aggressive choice tends to be faster, because more variables are fixed at once to explore fewer rapid branching nodes, but the solution quality is lower. However, there are a few exceptions, e.g., instance REQ_07 explores less nodes and terminates with a better solution. Choosing a very moderate setting leads to larger computation times and more evaluation of rapid branching nodes with the adavantage that the solution quality is in general higher. In addition, one can see that the rapid branching heuristic sometimes fails to produce solutions, e.g., for instance REQ_11 with aggressive or moderate settings. However, with choosing $\epsilon = 0.25$, high quality solutions for large-scale track allocation problems involving hundreds of train requests can be computed.

The benefit of the our algorithmic approach can be seen for very large scale instances. In Table 18 we list the results for instances with more than 500 requests through the network HAKAFU_SIMPLE. In addition, these instances have much more coupling rows than the instances of

instance	*#trains*	*#reqs*	*#rows*	*#cols*	v(LP)	v^*	gap in %	$t_{\sum}$ in s	*#bbn*
req06	198	213	6032	118056	366.31	330.64	10.79	27035.56	26
req07	171	184	8878	154847	336.41	297.91	12.92	9933.44	21
req08	160	199	23308	68767	374.31	271.78	37.73	41718.71	23
req11	0	98	4633	15224	166.62	271.78	-	2017.58	29
req12	98	113	14856	31302	245.39	193.17	27.03	3273.55	22
req17	216	285	1393	3692	395.29	392.76	0.64	37.81	15
req18	253	285	8218	140206	496.05	415.28	19.45	27577.64	39
req31	360	1062	6913	28318	464.78	461.97	0.61	2675.49	13
req32	257	1140	16489	28191	203.05	202.44	0.30	2628.38	21
req33	138	570	9036	12566	105.69	105.33	0.34	560.39	8

Table 15: Solution statistic of rapid branching with *aggressive* settings.

instance	*#trains*	*#reqs*	*#rows*	*#cols*	v(LP)	v^*	gap in %	$t_{\sum}$ in s	*#bbn*
req06	0	213	6032	118056	366.31	-	∞	30204.58	49
req07	172	184	8878	154847	336.41	289.58	16.17	10916.31	39
req08	166	199	23308	68767	374.31	280.01	33.68	33771.72	24
req11	0	98	4633	15224	166.62	176.17	-	3385.86	28
req12	0	113	14856	31302	245.39	176.17	39.29	2393.54	34
req17	216	285	1393	3692	395.29	392.76	0.64	46.87	5
req18	254	285	8218	140206	496.05	374.53	32.45	51728.37	42
req31	359	1062	6913	28318	464.78	453.37	2.52	3026.06	8
req32	257	1140	16489	28191	203.05	202.40	0.32	2735.25	13
req33	138	570	9036	12566	105.69	105.33	0.34	1209.50	23

Table 16: Solution statistic of rapid branching with *moderate* settings.

instance	*#trains*	*#reqs*	*#rows*	*#cols*	v(LP)	v^*	gap in %	$t_{\sum}$ in s	*#bbn*
req06	201	213	6032	118056	366.31	327.77	11.76	16124.86	29
req07	172	184	8878	154847	336.41	282.75	18.98	33985.51	31
req08	168	199	23308	68767	374.31	287.79	30.06	29716.57	27
req11	88	98	4633	15224	166.62	145.22	14.74	703.81	16
req12	96	113	14856	31302	245.39	182.13	34.73	2712.82	22
req17	216	285	1393	3692	395.29	392.76	0.64	48.87	9
req18	253	285	8218	140206	496.05	385.21	28.78	35562.13	42
req31	357	1062	6913	28318	464.78	456.39	1.84	3024.38	9
req32	256	1140	16489	28191	203.05	199.59	1.73	2838.52	24
req33	138	570	9036	12566	105.69	105.66	0.02	653.47	9

Table 17: Solution statistic of rapid branching with *default* settings.

instance	*#trains*	*#reqs*	*#rows*	*#cols*	v(LP)	v^*	gap in %	$t_{\sum}$ in s	*#bbn*
REQ_506	218	506	30213	282463	274.55	266.79	2.91	70186.90	2188
REQ_567	247	567	30595	259003	369.47	360.58	2.46	63573.24	1875
REQ_813	215	813	32287	225482	441.45	418.58	5.46	37627.05	157
REQ_875	239	875	36206	248922	395.10	368.22	7.30	46128.19	228
REQ_906	235	906	35155	265837	441.16	409.06	7.85	51234.58	471

Table 18: Solution statistic of `TS-OPT` for model (PCP) for very large instances.

the TTPlib. The associated graphs and static models are too big and cannot be solved on machines with 16GB main memory. Using default settings of rapid branching in `TS-OPT` and a limit maximum backtracks of 100 leads to the shown results. This demonstrates that rapid branching is a powerful heuristic to solve large scale track allocation problems and is able to produce high quality solution with a small optimality gap.

2.4 Conclusion

We showed that the bundle method and the rapid branching heuristic is a competitive approach to tackle large scale (PCP) formulations that are originating from railway track allocation problems. Furthermore, this illustrates that this solution approach has potential to be further generalized for solving large scale mixed integer programs. In particular, if the model formulation allows for a strong Lagrangean relaxation the bundle method has a lot of advantages in comparison to standard LP solvers, e.g., running time and total memory consumption. Moreover, our novel approach produced much faster high quality primal solutions and global upper bounds for several unsolved large scale track allocation instances of the TTPlib.

3 Auction Results

We consider in this section the results of a theoretical design of an auction-based allocation mechanism for railway slots in order to establish a fair and non-discriminatory access to a railway network. In this setting, railway undertakings (RU) compete for the use of a shared railway infrastructure by placing bids for trains that they intend to run. The main motivation and argumentation of that idea can be found in Borndörfer et al. (2006) [34]. The trains consume infrastructure capacity, such as track segments between and inside stations, over certain time intervals, and they can exclude each other due to safety and other operational constraints, even if they would not meet physically as we already define in detail in Chapter II. The auctioneer, i.e., an infrastructure manager, chooses from the bids a feasible subset, namely, a timetable, that maximizes the auction proceeds. Such a mechanism is desirable from an economic point of view because it can be argued that it leads to the most efficient use of a limited resource. However, it is clear that this vision can only become reality if the railway industry accepts sophisticated and modern technologies to support their planning and operational challenges. Figure 9 shows a general auction mechanism that has to be stated more precisely, e.g., definition of rounds, activity rules, definition and rules on bids and many more. Starting point is always the submission of initial bids by the participants. In the next step the winner determination problem is solved until the prede-

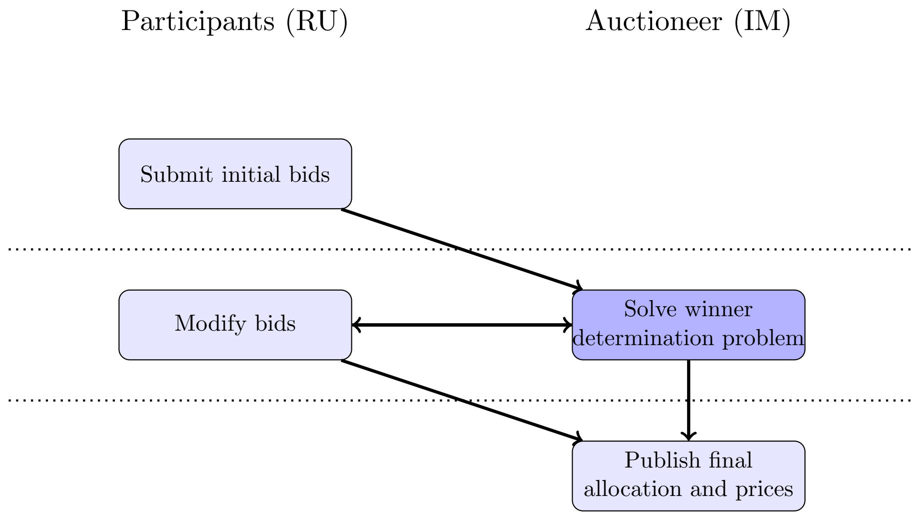


Figure 9: Auction procedure in general

fined conditions on termination are fulfilled, e.g., the maximal number of rounds is reached or there was no activity of the participants.

In the final stage the winner, i.e., the allocation of goods to bidders, and the corresponding prices are determined and published. A central question in mechanism design is whether there exists mechanisms ensuring *efficient* allocation, i.e., auctions that ensure that resources wind up in the hands of those who value them most. In other words an auction game is efficient, if in equilibrium the winner are the buyers with the highest valuation. The precise concept of equilibrium with respect to well-defined terminology of bids and valuations can be found in Milgrom (2004) [162].

In other industries well defined and implementable auction variants are an established mechanism to allocate scarce goods, e.g., energy market, telecommunication frequencies, airport slots, and ticketing of major events. However, the technical complexity and size of the railway resources act as a barrier to establish an auction based capacity allocation procedure. The winner determination problem of a railway auction is then to solve the track allocation problem discussed in Chapter III. Obviously, this procedure has to be defined and controlled by an independent agency, i.e., the Federal Network Agency in case of Germany.

In the following sections, we will define and discuss different auction designs. Some more from theoretical others from a computational and practically implementable point of view.

3.1 The Vickrey Track Auction

Vickrey (1961) [209] argued in his seminal paper for the importance of incentive compatibility in auction design, and he showed that a *second price auction* has this property, as well as efficiency. In a second price auction the bidder who submitted the highest bid is awarded the object being sold and pays a price equal to the second highest amount bid.

William Vickrey was awarded the Nobel Memorial Prize in Economics together with James Mirrlees for their research on the economic theory of incentives under asymmetric information. He, and independently Clarke (1971) [68] and Groves (1973) [106], also proposed a sealed-bid auction that generalizes the simple Vickrey auction for a single item to the multi-item case, the so-called Vickrey-Clarke-Groves (VCG) mechanism, which is also *incentive compatible*. Incentive compatibility is a concept originally proposed by Hurwicz (1972) [118] to describe any set of rules or procedures for which individuals find it in their own best interest to behave non-strategically in particular, truthfully. This is important in a variety of contexts, such as creating the mechanism for electing representatives or for deciding who receives benefits within a welfare state. Moreover the field of mechanism design is a rather new and fruitful mathematical research area.

This classical result pertains to a combinatorial auction, in which bids are placed for bundles of items, and two bundles can be allocated if and only if they do not contain the same item. This is, however, not sufficient for a railway track auction, in which more general constraints on the compatibility of slots arise, e.g., from minimum headway constraints. Whatever these constraints may be, a second price auction can of course also be conducted in such a setting. However, it is a priori not clear if such an auction is incentive compatible.

In Borndörfer, Mura & Schlechte (2009) [40] we formally defined such a Vickrey Track Auction (VTA) and showed that this is indeed the case by straight-forward modification of the original proof. The proof of Mura (2006) [164] does not depend on the concrete structure of the TTP, i.e., it generalizes to combinatorial Vickrey auctions with arbitrary combinatorial winner determination problems. For example, it follows that a VTA with additional constraints on the number of slots that can be allocated to a bidder is also incentive compatible, because this rule can be dealt with by adding constraints to the specific winner determination problem.

Even if the VTA is only a one-shot auction, i.e., only one round is performed, the definition of the prices causes the solution of several winner determination problem, i.e., all winner determination problems with each of the winners excluded. Erdogan (2009) [83] focuses on the computational tractability of this algorithmic mechanism design by extending a branch and bound approach to a branch and remember algorithm that exploit several information of the original winner determination problem, i.e., usage of still valid cuts and solutions as warmstart information for the MIP solving. For artificial auction scenarios based on the instances of the TTPlib he reported an acceleration ratio of two for the Vickrey payment computations, i.e., as well as for the measured geometric mean of the total number of branch and bound nodes and simplex iterations needed.

Indeed this shows that the VTA has theoretically all desired properties and even the computation of the payments may be reasonably practicable with great efforts. Nevertheless it is really challenging to establish such an auction design in reality due to the complex and hardly transparent price determination process in particular for combinatorial auctions with a lot of participants. Furthermore, it is known, that the "generalized" Vickrey auction suffers from several severe practical drawbacks, see Ausubel & Milgrom (2005) [14].

- ▷ It does not allow for price discovery, that is, discovery of the market price if the buyers are unsure of their own valuations.
- ▷ It is vulnerable to collusion by losing bidders.
- ▷ It is vulnerable to shill bidding with respect to the buyers.
- ▷ It does not necessarily maximize seller revenues; seller revenues may even be zero in VCG auctions.
- ▷ The seller's revenues are non-monotonic with regard to the sets of bidders and offers.

In these auctions, several criteria besides incentive compatibility merit the attention of a practical mechanism designer. Revenues are an obvious one. Auctions are commonly run by an expert auctioneer on behalf of the actual seller and any failure to select a core allocation with respect to reported values implies that there is a group of bidders who have offered to pay more in total than the winning bidders, yet whose offer has been rejected. Imagine trying to explain such an outcome to the actual seller or, in a government sponsored auction, to a skeptical public. Monotonicity of revenues with respect to participation is another important property of auction mechanisms, because its failure could allow a seller to increase sales revenues by disqualifying bidders after the bids are received. Another important desideratum is that a bidder should not profit by entering and playing as multiple bidders, rather than as a single one.

3.2 A Linear Proxy Auction

Designing an auction for the usage of railway infrastructure resources is nothing novel. Brewer & Plott (1996) [45] suggest a model where feasibility of a train schedule is based on the *binary exclusion* property, which says that a schedule of trains is feasible if any two trains are conflict-free. Parkes & Ungar (2001) [175] present an auction-based track allocation mechanism for the case that single-track, double-track, and yard segments have to be concatenated to form a single line. They suggest a hybrid mechanism that combines elements of the simultaneous and the combinatorial auction formats. However, these approaches are mainly driven by economic questions and assume almost trivial railway track allocation models and artificial data sets.

In that section we will present results of a more practically implementable iterative auction design with linear prices, i.e., the *Linearized Proxy Auction* (LPA). We will briefly discuss the main focus of that work. The precise auction design can be found in Schlechte & Tanner (2010) [189]. It generalizes the Ausubel Milgrom Proxy Auction presented by Ausubel & Milgrom (2002) [15]. Indeed no efficiency can be ensured but at least the resulting allocation lies in the *core*. An individually rational outcome is in the core of an auction game if and only if there is no group of bidders who would strictly prefer an alternative deal that is also strictly better for seller. Consequently, an auction mechanism that delivers core allocations has the advantage that there is no individual or group that would want either to renege after the auction is run in favor of some allocation that is feasible for it and the any non-core agreement made before the auction risks being unwound afterwards.

Our generalized variant (LPA) leads to the possibility of prices lying above the bidder-optimal core frontier, in contrast to the general Ausubel Milgrom Proxy auction. Some examples are discussed in Schlechte & Tanner (2010) [189]. However, main advantage of the design is the use of dual prices, i.e., the dual solution of the LP relaxation of model (ACP), to enforce *activity* in the iterative auction to decrease the number of auction rounds without loosing too much efficiency.

Table 19 lists the results of an auction simulation for real world demand data of the railway network HAKAFU_SIMPLE. The statistic basis of that data and the explicit auction rules, e.g., minimum increment, starting time of a bid, etc. can also be found in Schlechte & Tanner (2010) [189]. Furthermore, we scaled the profit values of the bidders with a constant scaling factor α to analyse the sensitivity of our auctioning approach.

	profit			auction rounds		
α	trivial	dual	efficiency	trivial	dual	speedup
0.8	2983	2932	0.983	17.65	13.61	25%
1.0	3658	3597	0.984	19.43	14.11	27%
1.5	4941	4843	0.980	20.06	15.4	23%
2.0	6144	5967	0.971	21.53	17.2	20%
2.5	7272	7065	0.972	21.77	18.23	16%
4.0	9720	9374	0.964	22.96	19.84	14%
6.0	12233	11879	0.971	23.12	19.59	15%

Table 19: Incremental auction with and without dual prices: profit and number of rounds until termination

Table 19 compares two versions of the LPA auction. The first version of the LPA, denoted as trivial, does not know any minimum price rule for newly introduced slots, so bidders start bidding for slots from price zero. The second version of LPA uses the dual-based minimum price rule and is therefore labeled with dual. We compare the results in efficiency and convergence rate. The second and the third column of Table 19 show the outcome for both LPA versions: one can see that the minimum price rule does not essentially affect efficiency in the next column. However, the last columns demonstrate that the number of rounds is significantly lower with the dual minimum price rule. We observe that using dual prices as minimum prices may speedup the auction while the efficiency loss is moderate for our test cases.

3.3 Conclusion

We presented and discussed several aspects of different theoretical auctioning procedures for the use of railway infrastructure resources. We want to point out explicitly that because of the character of the experiments and several assumptions on the auction setting most of our contributions are theoretic ones. Our experience from discussions with several European railway infrastructure managers is that “real” auctioning is a visionary idea that is hardly imaginable and implementable in the near future. However, the iterative resolution of resource conflicts in the coordination phase, see again Figure 8, can obviously be exchanged by more efficient procedures using an automatic track allocation tool embedded in an appropriate auction design. Still a lot of decision makers have to be convinced until the railway industry will agree on such an procedure. An adequate auction design with specified rules for “railway capacity” as for instance in the telecommunication market for frequencies, see Brunner et al. (2007) [47] and Ausubel & Milgrom (2002) [15], has to be defined and supported by the majority of railway actors.

4 The Simplon Corridor

In this section we present the results of the developed models and algorithms of Chapter II and III for a real world application, i.e., the Simplon corridor in Switzerland. The scenarios are extensively described from Section 4.1 to Section 4.3. Finally, Section 4.5. provides a capacity analysis of the Simplon tunnel using our optimization framework for railway track allocation.

4.1 Railway Network

There are only two north-south railway connections through the Alps in Switzerland, namely, the Gotthard corridor and the Lötschberg-Simplon corridor. The Simplon connects Switzerland and Italy and is therefore of strategic importance for the international railway freight traffic. It has a length of approximately 45 km and 12 stations. This may sound like a rather small network at first glance, but the routing possibilities at the terminals Brig and Domodossola, the routing possibilities in the intermediate stations Iselle and Varzo, and a rather unusual slalom routing for certain types of cargo trains lead to very complex planning situations. An `OpenTrack` network data export for the part from Brig (BR) in Switzerland to Domodossola (DO) in Italy

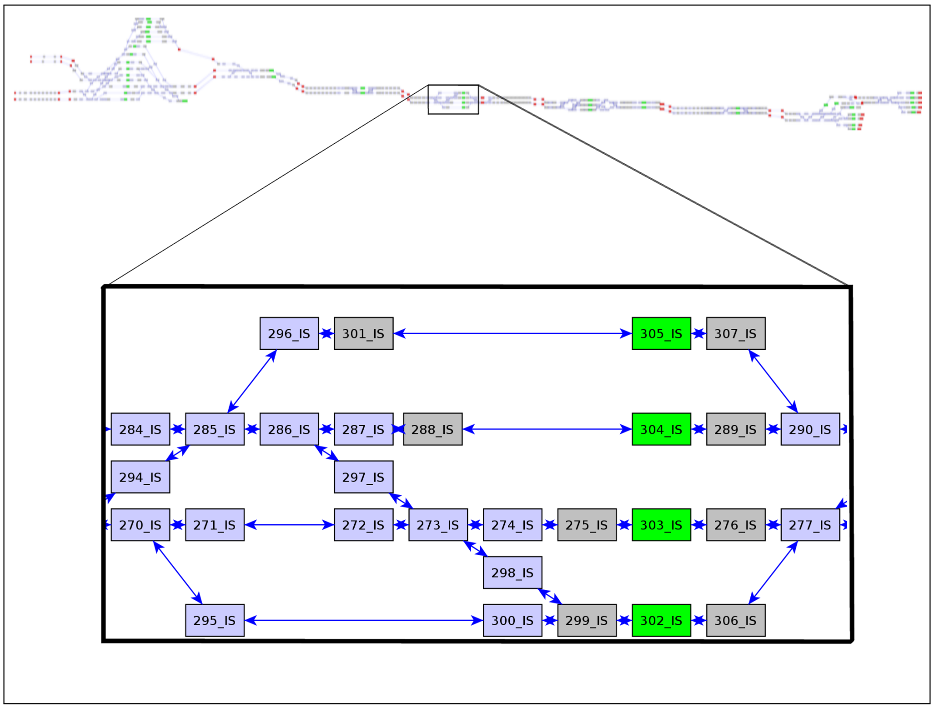


Figure 10: Micro graph representation of Simplon and detailed representation of station Iselle given by `OpenTrack`.

was provided by the SBB Schweizerische Bundesbahnen. The microscopic network consists of 1154 nodes and 1831 arcs including 223 signals, see Figure 10. Even if this network consists of only 12 stations and has a length of approximately 45 km, it is an important corridor in the European railway network. According to geographical conditions there are only two north-south railway corridors in Switzerland, the Gotthard corridor and the Lötschberg-Simplon corridor. This is in conflict with the fact that Switzerland is an very important country for the traffic transit between central und southern Europe. To that effect there is a huge and increasing demand on slots through this corridor. The Simplon tunnel is in fact a bottleneck in the European railway network.

This data was macrotized in two steps. The first step is resort to standardized train driving dynamics that lead to the definition of a handful of *train types*; these are used to compute standardized driving and headway times. This allows to amalgamate larger parts of the microscopic infrastructure network to a *macroscopic network* in the second step. The following subsections describe this process for the Simplon application.

4.2 Train Types

The decision which train types to consider is a crucial point, because a more detailed consideration of driving dynamics allows the construction of tighter schedules. For a capacity analysis, however, a modelling strategy is appropriate that captures the main characteristics, but abstracts from minor special characteristics of individual trains. We use six different types, two for passenger trains and four for freight trains.

The different, but invariable stopping patterns of *regional trains* (R) and *intercity trains* (EC) and their very different driving dynamics (due to the different engines used) result in considerable differences in running and headway times for such trains. They are therefore considered as two train types. We do, however, ignore different train compositions, i.e., in length and in the number of wagons. Hence, R and EC are the two types of *passenger trains* that we consider.

Freight trains come in four different types. GV Auto are special train services that transport passengers and their automobiles from Brig (BR) to the next station after the Simplon tunnel, which is Iselle (IS). There these trains cross all other tracks to reach an isolated ramp. Because of these unique routing requirements at Iselle, we consider them as belonging to an individual freight train type on their own.

GV RoLa and GV SIM are train types that transport freight vehicles (GV RoLa) and containers (GV SIM). They have a larger height and width than standard freight trains, and they can use only one of the tracks in the tunnel between Iselle and Preglia. This results in a so-called "slalom route" that these trains have to take from Brig. In Iselle they have to change to the right track[2] until Preglia, i.e., it is possible to change again to the standard side in the intermediate station Varzo to let other trains pass. Furthermore, the running times of these trains types, especially for the direction from Brig to Domodossola, differ significantly, namely, a GV RoLa needs about 7 minutes more than a GV SIM. They also use different routes in the area of Domodossola. Thus separate train types GV RoLa and GV SIM are introduced. Finally, GV MTO are standard freight trains which use the standard tracks in the Iselle-Preglia tunnel.

SBB was interested in running additional freight trains through the Simplon such that we concentrated on freight traffic. We assume, in

[2]In Switzerland trains are usually running on the left side.

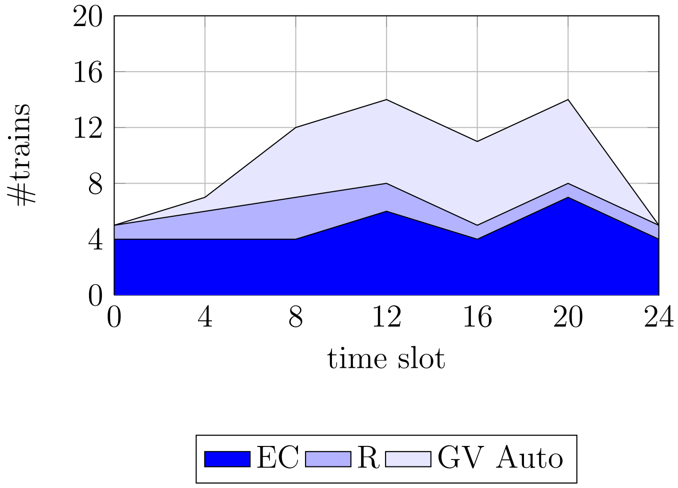


Figure 11: Given distribution of passenger or fixed traffic in the Simplon corridor for both directions.

particular, that the passenger trains are given and cannot be changed. Hence the slots for passenger trains R and EC from Brig to Domodossola and vice versa are fixed. In addition, the GV Auto trains, which are not operated all day, are also fixed. All these trains must, however, be considered with respect to their influence on the remaining traffic, i.e., with respect to their headways and with respect to station capacities. Figure 11 shows the passenger train distribution across the day.

4.3 Network Aggregation

The train types introduced in Section 4.2 can run on 28 different routes through $G = (V, E)$. The routes differ in their stopping pattern and in various ways to pass through Varzo. These routes are the basis of the aggregation of the microscopic network. They partition the network into segments, on which driving and headway times can be computed individually. In other words, if a train route runs on a track segment and no other routes cross, one can compute the parameters that are relevant for a slot allocation on this segment beforehand, and compress the segment.

Clearly, the routes meet at the stations, such that the macroscopic network must necessarily contain a node for each of the twelve stations. Some more macroscopic *pseudo nodes* are needed to model all train route interactions correctly, i.e., divergences, convergences, and crossings. Applying the NETCAST Micro-Macro Transformation algorithm described in Chapter II and in Schlechte et al. (2011) [190] produces a macroscopic network $N = (S, J)$ with 55 nodes and 87 tracks. 32 of these nodes are pseudo stations. Most of them are located directly in the front area of stations. The other 23 nodes are possible start, end, or waiting nodes along the corridor.

This automatically constructed network was further aggregated in a second step by applying some reductions that are not yet genericly implemented in NETCAST. We kept only those pseudo stations that handle crossing conflicts, namely, for GV Auto on the route from Brig to Iselle and those for a detailed modeling of the station Varzo. The reason for this detailed treatment of Varzo is that the routing through this station is crucial for the capacity of the whole corridor. In Varzo the over-width freight trains can pass each other, such that a locking of the entire area between Iselle and Preglia can be avoided for GV SIM and GV RoLa trains from the other direction, when one of them runs through the tunnel. All other potential pseudo nodes were aggregated to the closest station node in a conservative manner, i.e., the headway times for the incident tracks had to be slightly overestimated. In addition, some nodes that represent different platforms at the same station were aggregated. After these modifications the network consists of 18 stations and 40 tracks. For comparison, we also consider a "traditional" macroscopic network that is solely based on station nodes; clearly, a conservative model based on such an aggregation will employ oversized buffers and therefore waste capacity. Let us list the macroscopic networks, that we constructed by NETCAST on the basis of microscopic `OpenTrack` data:

- ▷ network with station area aggregation (18 stations and 40 tracks), SIMPLON_BIG,
- ▷ network with full station aggregation (12 stations and 28 tracks), SIMPLON_SMALL.

After some experiments with these networks, the expertise of SBB about the operational conditions in the Simplon corridor led to the introduction an additional technical blocking time for combinations of GV RoLa trains with any other trains in the front area of Domodossola. The headway times of cargo trains were set to a fixed value of some minutes instead of the simulation values in order to guarantee certain departure and arrival distances in the marshaling yard of Brig. We further improved the macroscopic model by adding buffer times for

		type		direction		freight trains		
name	train requests	passenger	freight	BR-DO	DO-BR	GV RoLa	GV SIM	GV MTO
4h-tp-as-d	41	15	26	23	18	4	9	13
4h-tp-as-n	36	8	28	20	16	7	10	11
4h-tp-s-d	42	15	27	23	19	4	8	15
4h-f20-s	38	14	24	22	16	6	12	6
4h-f15-s	46	14	32	26	20	8	16	8
4h-f12-s	54	14	40	30	24	10	20	10
4h-f10-s	62	14	48	34	28	12	24	12
4h-f7.5-s	78	14	64	42	36	16	32	16
24h-tp-as	390	63	327	203	187	69	108	150
24h-tp-s-n	219	63	156	110	109	48	54	54
24h-tp-s	297	63	234	149	148	60	78	96
24h-f24-s	183	63	120	92	91	30	60	30
24h-f20-s	207	63	144	104	103	36	72	36
24h-f15-s	255	63	192	128	127	48	96	48
24h-f12-s	303	63	240	152	151	60	120	60
24h-f10-s	351	63	288	176	175	72	144	72

Table 20: Statistics of demand scenarios for the Simplon case study

standard headways and headways for the opposite direction. In this way, two more macroscopic networks were generated with **NETCAST**:

- ▷ with station area aggregation (18 stations and 40 tracks) and technical times, SIMPLON_TECH,
- ▷ with station area aggregation (18 stations and 40 tracks) and technical and buffer times SIMPLON_BUF.

4.4 Demand

In order to evaluate and analyze the Micro-Macro Transformation introduced in Chapter II and the optimization models discussed in Chapter III we considered various train request scenarios. The capacity of the Simplon corridor is estimated by saturating it with freight trains, that are selected from fictional request sets. To this purpose, we have constructed 16 train request sets listed in Table 20. The first eight request sets cover a four hour time horizon (prefix "4h" in the request set name) either from 8am to 12am (suffix "d" for day) or from 0am to 4am (suffix "n" for night). The other request sets are used to calculate a timetable for an entire day (24h).[3]

Three of the 4h request sets are called "testplan" (tp), which means that they are used to evaluate the correctness of the Micro-Macro Transformation on the basis of a microscopically feasible timetable that has been generated manually by the authors. The same applies to the three "testplan" request sets that cover the whole day. Some of the test

[3]The "n" in the second 24h request is a reminder that freight trains drive more frequently at night.

	Brig-Domodossola		Domodossola-Brig	
Δ (sec.)	running	headway	running	headway
1	1778	272	1794	251
6	297	46	299	42
12	158	23	149	21
30	60	10	60	9
60	30	5	30	5
300	6	1	6	1

Table 21: Running and headway times for EC with respect to Δ

request sets, e.g., 24h-tp-as, have the disadvantage that the requests are not symmetrically distributed with respect to both directions. We therefore distinguish between asymmetric (as) and symmetric (s) request sets, which do not have this drawback.

We also remark that almost all "tp" request sets do not match the train type distribution that is desired by SBB. Namely, traffic demand in practice takes the form that every second request is a GV SIM, while the others are GV RoLa and GV MTO in equal parts. To approximate this characteristic, we generated some more requests using a uniform distribution according to the desired train demand pattern. The resulting request sets are named with the infix "fx", where x denotes the period time of the freight trains. We remark that we are aware of the fact that in practice traffic demand is not uniformly distributed, however, for want of better data and for the purpose of demonstrating the principal viability of our model in an analysis of a theoretical capacity of the corridor, we deem this data good enough.

Observation 4.1. *We will briefly discuss the impact of discretization on the real world data of the Simplon. The best usage from a simple capacity point of view without considering realistic traffic assumptions is trivially to use only the fastest train as much as possible. For the given Simplon corridor this is an EC train with times for both directions listed in Table 21. We denote by d the rounded running time with respect to Δ and by h the technical minimal rounded headway time, respectively.*

Even this trivial consideration of the corridor as a network of only two stations and two tracks documents the sensitivity of the macroscopic model with respect to the chosen discretization Δ. Assuming a coarse unit of 5 minutes, it is only possible to run 12 $\left(= \frac{3600}{300}\right)$ trains in each direction per hour. Only when Δ is smaller than 12 or 6 seconds, a *maximum capacity of* 13 *or* 14 $\left(= \frac{3600}{42 \cdot 6}\right)$ *trains per direction and per hour is theoretically available.*

4.5 Capacity Analysis based on Optimization

We provide in this section a capacity analysis of the Simplon corridor using our micro-macro aggregation approach. The goal of this study is to saturate the residual capacity of the corridor by running a maximum number of fictitious freight trains (GV MTO, GV SIM, GV RoLa) between the passenger trains (remember the passenger trains are given as fixed).

We remark that there are a lot of side-constraints for such additional trains that we do not consider. Requirements such as desired arrival or departure time windows at certain stations, dwell time requirements, the balance of train traffic in opposite directions, and other constraints are ignored, partly because of lack of data, partly because there is no point for such constraints in an analysis of a theoretical capacity maximum. These considerations are also the reason for using the following simple objective function:

- a basis value for each scheduled train depending on type and direction,
- a penalty for deviations from optimal arrival and departure times,
- and very small penalties for travel time increases or avoidable stops.

We constructed the macroscopic scenarios associated with all request sets and with all four macroscopic networks, namely SIMPLON_SMALL, SIMPLON_BIG, SIMPLON_TECH, and SIMPLON_BUF. Furthermore, we varied the time discretization of the model using step sizes of 6, 10, 30, and 60 seconds. The resulting macroscopic track allocation problems were solved using the integer programming based track allocation optimizer `TS-OPT` presented in Chapter III, the solutions were disaggregated using `NETCAST` and verified by `OpenTrack`. For each run of `TS-OPT`, a time limit of one day (86400 seconds) was used.

Table 22 lists exemplary solution statistics for all request scenarios and network SIMPLON_BIG using a discretization of 10s. The tables gives:

- number of trains (*#trains*),
- number of columns of the integer program (*#cols*),

instance	#trains	#cols	#rows	v(LP)	ub^*	v^*	gap	t_{LP}	t_{IP}
4h-tp-as-d	35	70476	30432	149.35	147.27	147.27	–	0.00	18.68
4h-tp-as-n	27	35859	17136	151.21	146.39	146.39	–	0.03	14.60
4h-tp-s	36	106201	45873	90.77	70.57	70.57	–	23.28	2054.04
4h-f20-s	30	173929	69531	152.52	145.97	145.97	–	54.23	2397.83
4h-f15-s	34	110920	46870	151.76	136.90	136.90	–	18.82	1440.07
4h-f12-s	36	211745	84107	189.57	186.36	186.36	–	107.78	12508.98
4h-f10-s	37	235430	93501	206.09	200.33	200.33	–	153.58	12124.92
4h-f7.5-s	37	135746	56968	79.26	72.15	72.15	–	37.97	11856.11
24h-tp-as	203	462769	196238	1035.94	984.77	984.77	–	102.73	63588.77
24h-tp-s-n	154	284038	117208	794.62	760.63	760.63	–	40.45	1609.42
24h-tp-s	174	403017	167548	888.97	843.30	843.30	–	76.02	27391.87
24h-f24-s	143	444199	178162	722.29	697.12	697.12	–	92.60	4454.76
24h-f20-s	156	471759	195167	791.31	752.49	752.49	–	93.70	3779.25
24h-f15-s	174	660642	250673	919.22	885.43	861.84	2.74	235.06	86400.40
24h-f12-s	179	662236	259676	985.46	958.76	958.76	–	213.54	79497.37
24h-f10-s	193	791285	312943	1090.47	1069.70	1041.08	2.75	426.75	86400.71

Table 22: IP-Solution analysis of network SIMPLON_BIG with time discretization of 10s and a time limit of 24h

- ▷ number of rows of the integer program (#*rows*),
- ▷ optimal value of the linear relaxation (v(LP)),
- ▷ (best) proven upper bound (ub^*),
- ▷ (best) objective function value of integral solution (v^*),
- ▷ optimality gap in percent,
- ▷ time needed to solve the linear relaxation (t_{LP}),
- ▷ and, the total running time of `TS-OPT`.

A first important result is that `TS-OPT` is indeed able to compute a feasible, i.e., conflict free, slot allocation for all instances within one day. Figure 12 shows an example of a resulting train diagram with a valid block occupation for request set 24h-tp-as, network SIMPLON_BUF, and a discretization of 30s. The tractability of these instances is to do the network aggregation algorithm of **`NETCAST`** presented in Chapter II, which produces reasonably sized macroscopic networks that give rise to reasonably sized track allocation problems. There is no instance where `TS-OPT` needs more than 600 MB of main memory, and `TS-OPT` can therefore compute feasible solutions for almost all problems. This give evidence that our micro-macro aggregation approach and our extended formulation works very well.

Not every instance could be solved to proven optimality for each network and time setting. But the 4h-requests never took more than three and a half hours to be solved to optimality, and even for the really complex uniformly distributed daily scenarios feasible solutions with small optimality gaps could be computed. Moreover, the instance with the maximum number of train requests (24h-tp-as with 390 train requests) could be solved to optimality for each network and all time discretizations of 30 seconds and more. Table 22 shows that such an instance produces a timetable with 203 trains, which means that 140 freight

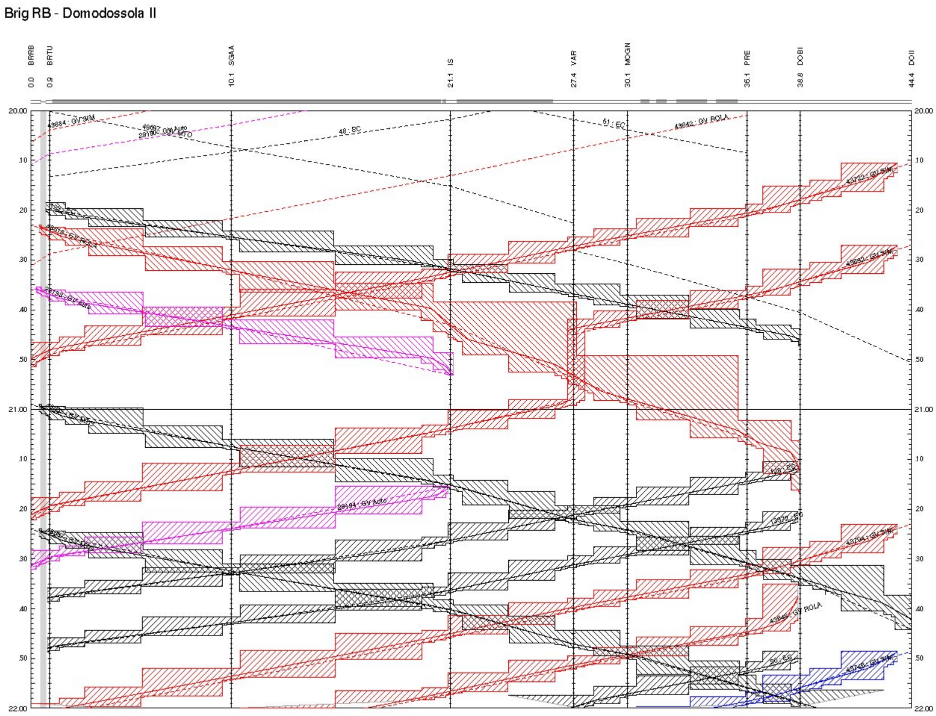


Figure 12: Traffic diagram in `OpenTrack` with block occupation for request 24h-tp-as

train slots out of the requested potential 327 train slots are routed in the optimal schedule. This establishes a theoretical capacity of the Simplon corridor of more than 200 trains per day. Adding technical and buffer times in network SIMPLON_BUF, it is still possible to schedule 170 trains. This number is almost identical to the saturation in the timetable that is currently in operation and can be taken as an indication of both the accuracy of the model as well as the quality of the current timetable. We can also observe that not every request set produces a saturated timetable that runs between 160 and 200 trains per day. This highlights the fact that the demand, i.e., the number of requested trains of different types and the degrees of freedom in routing them have a crucial effect on the capacity of a corridor.

We also analyzed the effects of different time discretizations. Table 23 and 24 give an overview on the sizes of the resulting track allocation problems for two test instances. We distinguish two different discretization parameters, namely, we denote by dep_steps the step size for train departure events and by wait_steps the step size for train dwell activi-

	24h-tp-as				
discretization (sec.)	6	10	30	30	60
dep_steps (sec.)	30	50	150	30	60
wait_steps (sec.)	60	100	300	60	60
#cols	504314	318303	114934	370150	178974
#rows	222096	142723	53311	170525	81961
t(lp) (sec.)	135.67	48.88	17.77	54.13	151.67
t(ip) (sec.)	72774.55	12409.19	110.34	81683.02	2411.20
size of IP (MB)	50	30	10	36	18
#trains	196	187	166	188	180

Table 23: Solution data of instance 24h-tp-as with respect to the chosen time discretization for SIMPLON_SMALL

	24h-f15-s				
discretization (sec.)	6	10	30	30	60
dep_steps (sec.)	30	50	150	30	60
wait_steps (sec.)	60	100	300	60	60
#cols	649494	375694	115293	392146	172462
#rows	234529	146044	49458	163388	74200
t(lp) (sec.)	190.36	64.59	2.83	47.44	103.50
t(ip) (sec.)	2923.76	2639.62	34.83	8265.71	1043.48
size of IP (MB)	64	36	10	38	16
#trains	176	163	143	155	145

Table 24: Solution data of instance 24h-f15-s with respect to the chosen time discretization for SIMPLON_SMALL

ties, respectively. As expected, problem sizes normally[4] decrease with coarser time discretizations, and the same holds for the running times. Anyway, `TS-OPT` can solve even instances with more than 500.000 variables.

An exception to the rule – coarser time discretization implies a decrease in problem size – can be observed by comparing the 30s and the 60s instance. This irregularity originates from a different parameter setting with respect to possible departure and waiting times, see Table 23. In the first 30s discretization scenario a train can only depart at times that are multiples of 150 seconds, see definition of dep_steps, and the waiting times must be a multiple of five minutes, see definition of wait_steps.

[4]There is no general relation between problem size and solution time as one can see by a comparison of the 6s-discretization runs.

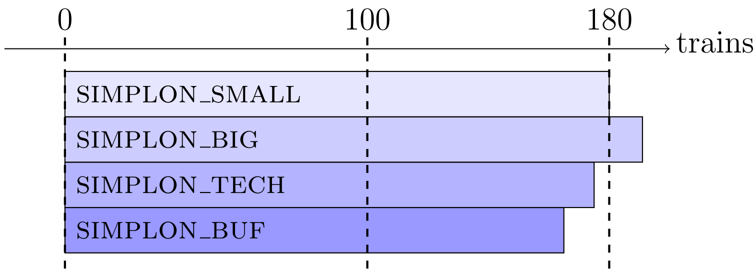


Figure 13: Comparison of scheduled trains for different networks (SIMPLON_) for instance 24h-tp-as in a 60s discretisation

	24h-tp-as		24h-f15-s	
	scheduled	requested	scheduled	requested
GV RoLa	30	69	21	48
GV SIM	41	108	51	96
GV MTO	69	150	39	48
all freight trains	140	327	111	192

Table 25: Distribution of freight trains for the requests 24h-tp-as and 24h-f15-s by using network SIMPLON_BIG and a rounding to 10 seconds

That is a rather rough model with a limited degree of freedom. We therefore changed the parameters for the 60s runs, such that the time steps are narrower and more similar to the 6s case. We also did 30s runs with departure and waiting times similar to the 6s cases, such that the influence of those two parameters could be analyzed. It turns out that there is not only a connection between time discretization and the number of scheduled trains, but there is also an often even stronger connection between departure and waiting time steps and the number of scheduled trains. We therefore also must pay attention to these parameters. We finally remark that the combinatorial complexity and/or the computational tractability of a particular track allocation instance can not be reliably predicted or estimated by looking at simple scenario statistics.

Another important point is the influence of network aggregation on the number of scheduled trains. As shown in Figure 13, a more detailed network model leads to a major increase in the number of scheduled trains. But by introducing specific headway times, we again loose about 8% of the trains and an additional 6% by also considering buffer times.

Up to now we only considered the total number of scheduled trains as a measure for the corridor capacity. But it is also important to keep the *structure* of the computed timetable in mind. Figure 14 shows the train type distribution of the three freight train types for two requests. This little example is representative for the general observation that the train type distribution associated with uniformly distributed requests is much closer to the desired distribution, see Figure 14, than that of the requests based on a testplan timetable. The latter timetables feature a higher fraction of GV MTO requests than desired; in fact, these trains do not run on a slalom route in the corridor and are therefore easier to schedule. The higher percentage of GV SIM and GV RoLa trains in the uniformly distributed request sets often leads to bigger problems than that resulting from the testplan request sets, see Table 23 and Table 24.

Another observation is that the majority of timetables schedules more trains from Domodossola to Brig than vice versa. This is not surprising as the models due not contain any symmetry constraints. We did, however, try to achieve some balance by manipulating the objective function. Introduce such global constraints could be an interesting aspect of future work.

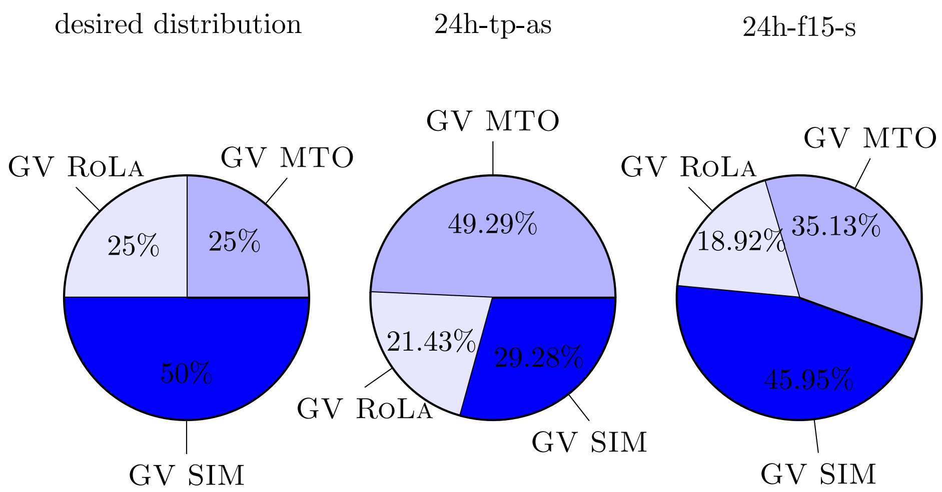


Figure 14: Distribution of freight trains for the requests 24h-tp-as and 24h-f15-s by using network SIMPLON_BIG and a rounding to 10 seconds

4.6 Conclusion

To the best knowledge of the author and confirmed by several railway practitioners this was the first time that automatically produced track allocations (on a macroscopic scale) fulfill the requirements of the original microscopic model. Furthermore, we strongly believe that our models and algorithmic solution approaches are already able to support the mid-term and long-term planning of track allocations, i.e., the creation of the annual time table. Finally, we want to complete the thesis with an excerpt from the project conclusions of our industry partners from SBB:

> "The produced timetables from this project are qualitatively better than all previous results of other projects. For the first time it was possible to simulate an algorithmic generated timetable in the simulation tool `OpenTrack` without conflicts .... We would expect a benefit (by introducing such a tool) on a strategic middle-term and long-term level. Because we estimate that we could decrease the planning time needed for freight train allocation from 2-3 weeks to only one week. In addition much more scenario variations could be considered and results could be produced much faster." (translation by the author).

References

[1] E. J. W. ABBINK, M. FISCHETTI, L. G. KROON, G. TIMMER & M. J. C. M. VROMANS. *Reinventing crew scheduling at netherlands railways*. Interfaces **35**(5):393–401, 2005. Cited on page 15.

[2] M. ABRIL, M. A. SALIDO & F. BARBER. *Distributed search in railway scheduling problems*. Eng. Appl. Artif. Intell. **21**(5):744–755, 2008. ISSN 0952-1976. Cited on page 36.

[3] T. ACHTERBERG. *Constraint Integer Programming*. PhD thesis, Technische Universität Berlin, 2007. Cited on pages 9, 51, 132.

[4] T. ACHTERBERG, T. KOCH & A. MARTIN. *MIPLIB 2003*. Operations Research Letters **34**(4):1–12, 2006. URL http://www.zib.de/Publications/abstracts/ZR-05-28. ZIB-Report 05-28. Cited on page 64.

[5] R. K. AHUJA, T. L. MAGNANTI & J. B. ORLIN. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1993. Cited on page 107.

[6] R. K. AHUJA, K. C. JHA & J. LIU. *Solving real-life railroad blocking problems*. INTERFACES **37**(5):404–419, 2007. URL http://interfaces.journal.informs.org/cgi/content/abstract/37/5/404. Cited on pages 18, 19.

[7] R. K. AHUJA, R. H. MÖHRING & C. D. ZAROLIAGIS, (Eds.). *Robust and Online Large-Scale Optimization: Models and Techniques for Transportation Systems*, vol. 5868 of *Lecture Notes in Computer Science*. Springer, 2009. ISBN 978-3-642-05464-8. Cited on pages 199, 202, 205.

[8] S. ALBERS. *Online algorithms: a survey*. Math. Program. **97**(1-2): 3–26, 2003. Cited on page 10.

[9] S. ALBERS & S. LEONARDI. *On-line algorithms*. ACM Comput. Surv. **31**(3es):4, 1999. Cited on page 10.

[10] T. ALBRECHT. *Railway Timetable and Traffic*, chap. Energy-Efficient Train Operation, pp. 83–105. Eurailpress/ DVV Media, 2008. Cited on page 93.

[11] Altenbeken. *Altenbeken*, 2009. URL http://www.db.de/site/bahn/de/geschaefte/infrastruktur__schiene/netz/netzzugang/dokumente/Bahnhof/SNB/H/HA__NBS.pdf. In German available at http://www.db.de/site/bahn/de/geschaefte/infrastruktur__schiene/netz/netzzugang/dokumente/Bahnhof/SNB/H/HA__NBS.pdf. Cited on pages xxv, 58.

[12] L. ANDEREGG, S. EIDENBENZ, M. GANTENBEIN, C. STAMM, D. S. TAYLOR, B. WEBER & P. WIDMAYER. *Train routing algorithms:*

Concepts, design choises, and practical considerations. In R. E. LADNER, (Ed.), *ALENEX*, pp. 106–118. SIAM, 2003. ISBN 0-89871-542-3. Cited on page 14.

[13] D. APPLEGATE, R. BIXBY, V. CHVÁTAL & W. COOK. Finding cuts in the TSP (a preliminary report). Technical report, Center for Discrete Mathematics and Theoretical Computer Science (DIMACS), March 1995. DIMACS Technical Report 95-05. Cited on page 145.

[14] L. M. AUSUBEL & P. MILGROM. Ascending proxy auctions. Levine's bibliography, UCLA Department of Economics, 2005. URL http://econpapers.repec.org/RePEc:cla:levrem:122247000000000785. Cited on page 173.

[15] L. M. AUSUBEL & P. R. MILGROM. *Ascending auctions with package bidding*. Frontiers of Theoretical Economics **1**(1):1–42, 2002. Cited on pages 174, 176.

[16] E. BALAS. *Projection, lifting and extended formulation in integer and combinatorial optimization*. Annals OR **140**(1):125–161, 2005. Cited on page 123.

[17] C. BARNHART & G. LAPORTE. *Handbooks in Operations Research & Management Science: Transportation*. North-Holland, 2007. Cited on pages 9, 10.

[18] C. BARNHART, E. L. JOHNSON, G. L. NEMHAUSER, M. W. P. SAVELSBERGH & P. H. VANCE. *Branch-and-price: Column generation for solving huge integer programs*. Oper. Res. **46**(3):316–329, 1998. ISSN 0030-364X. Cited on pages 48, 133.

[19] C. BARNHART, H. JIN & P. H. VANCE. *Railroad blocking: A network design application*. Oper. Res. **48**(4):603–614, 2000. ISSN 0030-364X. Cited on pages 18, 19.

[20] C. BARNHART, P. BELOBABA & A. R. ODONI. *Applications of operations research in the air transport industry*. Transportation Science **37** (4):368–391, 2003. Cited on page 44.

[21] N. BARNIER, P. BRISSET, T. RIVIRE & T. R. ERE. *Slot allocation with constraint programming: Models and results*. In *In Proc. of the Fourth International Air Traffic Management R&D Seminar ATM*, 2001. Cited on page 10.

[22] J. E. BEASLEY & N. CHRISTOFIDES. *An algorithm for the resource constrained shortest path problem*. Networks **19**:379–394, 1989. Cited on page 50.

[23] A. BEN-TAL & A. NEMIROVSKI. *Robust convex optimization*. Mathematics of Operations Research **23**(4):769–805, 1998. Cited on page 128.

[24] L. BENGTSSON, R. GALIA, T. GUSTAFSSON, C. HJORRING & N. KOHL. *Railway crew pairing optimization*. In F. GERAETS,

L. Kroon, A. Schoebel, D. Wagner & C. Zaroliagiis, (Eds.), *Algorithmic Methods for Railway Optimization*, LNCS, pp. 126–144. Springer-Verlag, 2007. Cited on page 15.

[25] D. Bertsimas & M. Sim. *Robust discrete optimization and network flows*. Mathematical Programming **98**:49–71, 2003. Cited on page 128.

[26] G. Birkhoff. *Lattice theory*, vol. 25. American Mathematical Society, Providence, R.I., 3 edition, 1967. Cited on page 68.

[27] K. S. Booth & G. S. Lueker. *Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms*. J. Comput. Syst. Sci. **13**(3):335–379, 1976. Cited on page 108.

[28] R. Borndörfer & C. Liebchen. *When periodic timetables are suboptimal*. In *OR*, pp. 449–454, 2007. Cited on page 35.

[29] R. Borndörfer & M. Neumann. Models for line planning with transfers. ZIB Report 10-11, ZIB, Takustr. 7, 14195 Berlin, 2010. Cited on page 26.

[30] R. Borndörfer & T. Schlechte. *Models for railway track allocation*. In C. Liebchen, R. K. Ahuja & J. A. Mesa, (Eds.), *ATMOS 2007 - 7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*, vol. 07001 of *Dagstuhl Seminar Proceedings*. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2007. Cited on pages ii, 90, 115, 153.

[31] R. Borndörfer & T. Schlechte. *Solving railway track allocation problems*. In J. Kalcsics & S. Nickel, (Eds.), *OR*, pp. 117–122. Springer, 2007. ISBN 978-3-540-77902-5. Cited on pages ii, 90.

[32] R. Borndörfer, M. Grötschel & A. Löbel. *Duty scheduling in public transit*. In W. Jäger & H.-J. Krebs, (Eds.), *MATHEMATICS – Key Technology for the Future*, pp. 653–674. Springer Verlag, Berlin, 2003. URL `http://www.zib.de/PaperWeb/abstracts/ZR-01-02`. ZIB Report 01-02. Cited on page 50.

[33] R. Borndörfer, U. Schelten, T. Schlechte & S. Weider. *A column generation approach to airline crew scheduling*. In *OR*, pp. 343–348, 2005. Cited on pages ii, 46, 52.

[34] R. Borndörfer, M. Grötschel, S. Lukac, K. Mitusch, T. Schlechte, S. Schultz & A. Tanner. *An auctioning approach to railway slot allocation*. Competition and Regulation in Network Industries **1**(2):163–196, 2006. URL `http://www.zib.de/PaperWeb/abstracts/ZR-05-45`. ZIB Report 05-45. Cited on pages ii, 90, 91, 126, 127, 170.

[35] R. Borndörfer, M. Grötschel & M. E. Pfetsch. *A column-generation approach to line planning in public transport*. Transporta-

tion Science **41**(1):123–132, 2007. ISSN 1526-5447. Cited on pages 10, 25.

[36] R. BORNDÖRFER, M. GRÖTSCHEL & U. JAEGER. *Planungsprobleme im öffentlichen Verkehr.* In M. GRÖTSCHEL, K. LUCAS & V. MEHRMANN, (Eds.), *PRODUKTIONSFAKTOR MATHEMATIK – Wie Mathematik Technik und Wirtschaft bewegt*, acatech DISKUTIERT, pp. 127–153. acatech – Deutsche Akademie der Technikwissenschaften und Springer, 2008. ISBN 978-3-540-89434-6. URL http://opus.kobv.de/zib/volltexte/2008/1103/. ZIB Report 08-20. Cited on page 10.

[37] R. BORNDÖRFER, A. LÖBEL & S. WEIDER. *A bundle method for integrated multi-depot vehicle and duty scheduling in public transit.* In M. HICKMAN, P. MIRCHANDANI & S. VOSS, (Eds.), *Computer-aided Systems in Public Transport (CASPT 2004)*, vol. 600 of *Lecture Notes in Economics and Mathematical Systems*, pp. 3–24. Springer-Verlag, 2008. Cited on pages x, xiv, 48, 141, 143.

[38] R. BORNDÖRFER, B. EROL & T. SCHLECHTE. *Optimization of macroscopic train schedules via TS-OPT.* In I. HANSEN, E. WENDLER, U. WEIDMANN, M. LÜTHI, J. RODRIGUEZ, S. RICCI & L. KROON, (Eds.), *Proceedings of the 3rd International Seminar on Railway Operations Modelling and Analysis - Engineering and Optimisation Approaches*, Zürich, Switzerland, 2009. Cited on pages ii, 90.

[39] R. BORNDÖRFER, M. GRÖTSCHEL & U. JAEGER. Planning problems in public transit. ZIB Report 09-13, ZIB, Takustr. 7, 14195 Berlin, 2009. URL http://opus.kobv.de/zib/volltexte/2009/1174/. To appear in English translations of acatech book. Cited on page 10.

[40] R. BORNDÖRFER, A. MURA & T. SCHLECHTE. *Vickrey auctions for railway tracks.* In B. FLEISCHMANN, K. H. BORGWARDT, R. KLEIN & A. TUMA, (Eds.), *Operations Research Proceedings 2008*, pp. 551–556. Springer-Verlag, 2009. URL http://opus.kobv.de/zib/volltexte/2008/1122/. ZIB Report 08-34. Cited on pages ii, 172.

[41] R. BORNDÖRFER, I. DOVICA, I. NOWAK & T. SCHICKINGER. Robust tail assignment. Technical Report ZIB Report 10-08, Zuse-Institut Berlin, Takustr. 7, 14195 Berlin, 2010. URL http://opus.kobv.de/zib/volltexte/2010/1231/. Cited on page 6.

[42] R. BORNDÖRFER, B. EROL, T. GRAFFAGNINO, T. SCHLECHTE & E. SWARAT. Optimizing the simplon railway corridor. ZIB Report 10-24, ZIB, Takustr. 7, 14195 Berlin, 2010. submitted to Annals of Operations Research 4.11.2010. Cited on pages ii, 55, 58.

[43] R. BORNDÖRFER, T. SCHLECHTE & S. WEIDER. *Railway track allocation by rapid branching.* In T. ERLEBACH & M. LÜBBECKE, (Eds.), *Proceedings of the 10th Workshop on Algorithmic Approaches*

for Transportation Modelling, Optimization, and Systems, vol. 14 of *OpenAccess Series in Informatics (OASIcs)*, pp. 13–23, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum gr Informatik. ISBN 978-3-939897-20-0. doi: http://dx.doi.org/10.4230/OASIcs.ATMOS.2010.13. URL http://drops.dagstuhl.de/opus/volltexte/2010/2746. Cited on pages ii, 90.

[44] U. Brännlund, P. Lindberg, A. Nou & J.-E. Nilsson. *Railway timetabling using langangian relaxation.* Transportation Science **32** (4):358–369, 1998. Cited on pages 36, 38, 40, 103, 106.

[45] P. J. Brewer & C. R. Plott. *A binary conflict ascending price (BICAP) mechanism for the decentralized allocation of the right to use railroad tracks.* International Journal of Industrial Organization **14**(6): 857–886, 1996. Cited on page 174.

[46] O. Brünger & E. Dahlhaus. *Running time estimation*, chap. 4, pp. 58–82. Railway Timetable & Traffic. Eurailpress, 2008. Cited on page 60.

[47] C. Brunner, J. Goersee, C. Holt & J. Ledyard. An experimental test of combinatorial fcc spectrum auctions. Technical report, California Institute of Technology, Pasadena, 2007. URL http://www.hss.caltech.edu/~jkg/fcc_smrpb.pdf. Cited on page 176.

[48] D. Burkolter, T. Herrmann & G. Caimi. *Generating dense railway schedules.* In A. Jaszkiewicz, M. Kaczmarek, J. Zak & M. Kubiak, (Eds.), *Advanced OR and AI Methods in Transportation*, pp. 290–297. Publishing House of Poznan University of Technology, 2005. URL http://euro2005.cs.put.poznan.pl/eproc.html. Cited on page 34.

[49] M. R. Bussieck. *Optimal lines in public rail transport.* PhD thesis, TU Braunschweig, 1997. Cited on pages 10, 25.

[50] M. R. Bussieck, T. Winter & U. T. Zimmermann. *Discrete optimization in public rail transport.* Mathematical Programming **79B** (1–3):415–444, 1997. Cited on pages xxiii, 9, 10.

[51] V. Cacchiani. *Models and Algorithms for Combinatorial Optimization Problems arising in Railway Applications.* PhD thesis, DEIS, Bologna, 2007. Cited on pages 39, 90, 103.

[52] V. Cacchiani, A. Caprara & P. Toth. *A column generation approach to traintimetabling on a corridor.* 4OR , 2007. To appear. Cited on pages 39, 40, 90, 142.

[53] V. Cacchiani, A. Caprara, L. Galli, L. G. Kroon & G. Maróti. *Recoverable robustness for railway rolling stock planning.* In *ATMOS*, 2008. Cited on page 34.

[54] V. Cacchiani, A. Caprara & P. Toth. *Scheduling extra freight trains on railway networks.* Transportation Research Part B: Method-

ological **44**(2):215–231, 2010. URL http://econpapers.repec.org/RePEc:eee:transb:v:44:y:2010:i:2:p:215-231. Cited on pages xxiv, 39, 40, 90, 159, 160.

[55] X. CAI & C. J. GOH. *A fast heuristic for the train scheduling problem*. Comput. Oper. Res. **21**(5):499–510, 1994. ISSN 0305-0548. Cited on pages 36, 38.

[56] X. CAI, C. J. GOH & A. MEES. *Greedy heuristics for rapid scheduling of trains on a single track*. IIE Transactions **30**(5):481 – 493, 1998. URL http://www.springerlink.com/content/v9t27h636427t066. Cited on pages 36, 38.

[57] G. CAIMI. *Algorithmic decision support for train scheduling in a large and highly utilised railway network*. PhD thesis, ETH Zurich, 2009. Cited on pages xxv, 2, 18, 33, 34, 41, 57, 90, 91.

[58] G. CAIMI, D. BURKOLTER & T. HERRMANN. *Finding delay-tolerant train routings through stations*. In *OR*, pp. 136–143, 2004. Cited on page 129.

[59] G. C. CAIMI, M. FUCHSBERGER, M. LAUMANNS & K. SCHÜPBACH. *09. periodic railway timetabling with event flexibility*. In C. LIEBCHEN, R. K. AHUJA & J. A. MESA, (Eds.), *ATMOS 2007 - 7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*, Dagstuhl, Germany, 2007. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany. ISBN 978-3-939897-04-0. URL http://drops.dagstuhl.de/opus/volltexte/2007/1173. Cited on page 34.

[60] A. CAPRARA, M. FISCHETTI & P. TOTH. *Algorithms for the set covering problem*. Annals of Operations Research **98**:2000, 1998. Cited on page 147.

[61] A. CAPRARA, M. FISCHETTI, P. L. GUIDA, M. MONACI, G. SACCO & P. TOTH. *Solution of real-world train timetabling problems*. In *HICSS 34*. IEEE Computer Society Press, 2001. Cited on pages 38, 108.

[62] A. CAPRARA, M. FISCHETTI & P. TOTH. *Modeling and solving the train timetabling problem*. Operations Research **50**(5):851–861, 2002. Cited on pages 38, 40, 103, 108.

[63] A. CAPRARA, M. MONACI, P. TOTH & P. L. GUIDA. *A lagrangian heuristic algorithm for a real-world train timetabling problem*. Discrete Appl. Math. **154**(5):738–753, 2006. ISSN 0166-218X. Cited on pages 36, 106.

[64] A. CAPRARA, L. KROON, M. MONACI, M. PEETERS & P. TOTH. *Passenger railway optimization*. In C. BARNHART & G. LAPORTE, (Eds.), *Handbooks in Operations Research and Management Science*, vol. 14, chap. 3, pp. 129–187. Elsevier, 2007. Cited on pages 12, 103.

[65] M. CAREY & D. LOCKWOOD. *A model, algorithms and strategy for train pathing*. The Journal of the Operational Research Society **46**, 1995. Cited on page 38.

[66] L. CASTELLI, P. PELLEGRINI & R. PESENTI. Airport slot allocation in europe: economic efficiency and fairness. Working Papers 197, Department of Applied Mathematics, University of Venice, 2010. URL `http://econpapers.repec.org/RePEc:vnm:wpaper:197`. Cited on page 10.

[67] A. CHARNES & M. MILLER. *A model for the optimal programming of railway freight train movements*. Management Science **3**(1):74–92, 1956. Cited on pages xxv, 2, 5, 6.

[68] E. H. CLARKE. *Multipart pricing of public goods*. Public Choice **2**: 19–33, 1971. Cited on page 172.

[69] J. CLAUSEN, A. LARSEN, J. LARSEN & N. J. REZANOVA. *Disruption management in the airline industry-concepts, models and methods*. Comput. Oper. Res. **37**(5):809–821, 2010. ISSN 0305-0548. Cited on page 15.

[70] J.-F. CORDEAU, P. TOTH & D. VIGO. *A Survey of Optimization Models for Train Routing and Scheduling*. TRANSPORTATION SCIENCE **32**(4):380–404, 1998. URL `http://transci.journal.informs.org/cgi/content/abstract/32/4/380`. Cited on page 38.

[71] F. CORMAN, R. M. GOVERDE & A. D'ARIANO. *Rescheduling Dense Train Traffic over Complex Station Interlocking Areas*, pp. 369–386. Springer-Verlag, Berlin, Heidelberg, 2009. ISBN 978-3-642-05464-8. doi: http://dx.doi.org/10.1007/978-3-642-05465-5_16. Cited on page 15.

[72] A. D'ARIANO, F. CORMAN, D. PACCIARELLI & M. PRANZO. *Reordering and local rerouting strategies to manage train traffic in real time*. Transportation Science **42**(4):405–419, 2008. ISSN 1526-5447. Cited on page 15.

[73] DB Netze AG. *DB Netze AG-Homepage*, 2010. URL `http://www.dbnetze.com`. http://www.dbnetze.com. Cited on page 91.

[74] X. DELORME, X. GANDIBLEUX & J. RODRIGUEZ. *Stability evaluation of a railway timetable at station level*. European Journal of Operational Research **195**(3):780–790, 2009. Cited on pages 57, 129.

[75] J. DESROSIERS, F. SOUMIS & M. DESROCHERS. Routes sur un réseau espace-temps. Technical Report 236, Centre de recherche sur les transports, Université de Montréal, 1982. Cited on page 96.

[76] M. J. DORFMAN & J. MEDANIC. *Scheduling trains on a railway network using a discrete event model of railway traffic*. Transportation Research Part B: Methodological **38**(1):81 – 98, 2004. ISSN 0191-2615. URL `http://www.sciencedirect.com/science/article/`

B6V99-484SFYN-2/2/e474b988e5fca3c08b20c1cf991a960b. Cited on page 36.

[77] J. ECKSTEIN & M. NEDIAK. *Pivot, cut, and dive: a heuristic for 0-1 mixed integer programming*. J. Heuristics **13**(5):471–503, 2007. Cited on page 145.

[78] M. EHRGOTT. *Multicriteria Optimization*. Springer Verlag, Berlin, 2 edition, 2005. Cited on pages 129, 131, 132.

[79] M. EHRGOTT & D. RYAN. *Constructing robust crew schedules with bicriteria optimization*. Journal of Multi-Criteria Decision Analysis **11**: 139–150, 2002. Cited on page 129.

[80] S. EIDENBENZ, A. PAGOURTZIS & P. WIDMAYER. *Flexible train rostering*. In T. IBARAKI, N. KATOH & H. ONO, (Eds.), *ISAAC*, vol. 2906 of *Lecture Notes in Computer Science*, pp. 615–624. Springer, 2003. ISBN 3-540-20695-7. Cited on page 14.

[81] L. EL-GHAOUI, F. OUSTRY & H. LEBRET. *Robust solutions to uncertain semidefinite programs*. SIAM J. Optim. **9**:33–52, 1998. Cited on page 128.

[82] D. EMERY. *Enhanced ETCS Level 3 train control system*. In A. TOMII, J. ALLAN, E. ARIAS, C. BREBBIA, C. GOODMAN, A. RUMSEY & G. SCIUTTO, (Eds.), *Computers in Railways XI*. WIT Press, 2008. Cited on page 63.

[83] C. ERDOGAN. Computing prices for track allocations. Diploma thesis, TU Berlin, 2009. Cited on page 173.

[84] B. EROL. Models for the train timetabling problem. Diploma thesis, TU Berlin, 2009. URL http://www.zib.de/Optimization/Projects/TrafficLogistic/TrassenB/thesis_erol.pdf. Cited on pages 57, 117, 121.

[85] B. EROL, M. KLEMENZ, T. SCHLECHTE, S. SCHULTZ & A. TANNER. *TTPlib 2008 - A library for train timetabling problems*. In A. TOMII, J. ALLAN, E. ARIAS, C. BREBBIA, C. GOODMAN, A. RUMSEY & G. SCIUTTO, (Eds.), *Computers in Railways XI*. WIT Press, 2008. URL http://opus.kobv.de/zib/volltexte/2008/1102/. Cited on pages ii, 64, 91, 150, 161.

[86] B. FEDERAL MINISTRY OF TRANSPORT & HOUSING. *Regulation for the use of railway infrastructure*, 2005. URL http://www.gesetze-im-internet.de/eibv_2005/BJNR156610005.html. Cited on page 32.

[87] FEDERAL TRANSPORT INFRASTRUCTURE PLANNING PROJECT GROUP. *Federal transport infrastructure plan 2003*, 2003. URL http://www.bmvbs.de/Anlage/original_17121/Federal-Transport-Infrastructure-Plan-2003.pdf. Cited on pages xxv, 1, 2.

[88] P.-J. FIOOLE, L. G. KROON, G. MARÓTI & A. SCHRIJVER. *A rolling stock circulation model for combining and splitting of passenger trains*. European Journal of Operational Research **174**(2):1281–1297, 2006. Cited on page 14.

[89] F. FISCHER & C. HELMBERG. *Dynamic graph generation and dynamic rolling horizon techniques in large scale train timetabling*. In T. ERLEBACH & M. LÜBBECKE, (Eds.), *Proceedings of the 10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, vol. 14 of *OpenAccess Series in Informatics (OASIcs)*, pp. 45–60, Dagstuhl, Germany, 2010. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-939897-20-0. doi: http://dx.doi.org/10.4230/OASIcs.ATMOS.2010.45. URL http://drops.dagstuhl.de/opus/volltexte/2010/2749. Cited on pages 39, 40, 90, 133.

[90] F. FISCHER, C. HELMBERG, J. JANSSEN & B. KROSTITZ. *Towards solving very large scale train timetabling problems by lagrangian relaxation*. In M. FISCHETTI & P. WIDMAYER, (Eds.), *ATMOS 2008 - 8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems*, Dagstuhl, Germany, 2008. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany. URL http://drops.dagstuhl.de/opus/volltexte/2008/1585. Cited on pages 39, 40, 90, 103, 109, 142, 149.

[91] M. FISCHETTI, D. SALVAGNIN & A. ZANETTE. *Fast approaches to improve the robustness of a railway timetable*. Transportation Science **43**(3):321–335, 2009. ISSN 1526-5447. Cited on pages 34, 128.

[92] B. A. FOSTER & D. M. RYAN. *An integer programming approach to scheduling*. In *Computer Aided Scheduling of Public Transport*. Springer Verlag, Berlin, 1991. Cited on pages 48, 142.

[93] A. FRANGIONI. *About lagrangian methods in integer optimization*. Annals of Operations Research **139**:163–193, 2005. ISSN 0254-5330. URL http://dx.doi.org/10.1007/s10479-005-3447-9. 10.1007/s10479-005-3447-9. Cited on page 135.

[94] M. FUCHSBERGER. Solving the train scheduling problem in a main station area via a resource constrained space-time integer multi-commodity flow. Master's thesis, Institut for Operations Research ETH Zürich, 2007. Cited on pages 57, 103.

[95] A. FÜGENSCHUH, H. HOMFELD, A. HUCK, A. MARTIN & Z. YUAN. *Scheduling Locomotives and Car Transfers in Freight Transport*. Transportation Science **42**(4):1 – 14, 2008. Cited on page 19.

[96] A. FÜGENSCHUH, H. HOMFELD & H. SCHÜLLDORF. *Single car routing in rail freight transport*. In C. BARNHART, U. CLAUSEN, U. LAUTHER & R. MÖHRING, (Eds.), *Dagstuhl Seminar Proceedings 09261,*

Schloss Dagstuhl – Leibniz-Zentrum fr Informatik, Deutschland, 2009. Cited on page 19.

[97] M. GAREY & D. JOHNSON. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979. Cited on page 104.

[98] K. GHOSEIRI, F. SZIDAROVSZKY & M. J. ASGHARPOUR. *A multi-objective train scheduling model and solution*. Transportation Research Part B: Methodological **38**(10):927 – 952, 2004. ISSN 0191-2615. URL http://www.sciencedirect.com/science/article/B6V99-4C0053J-1/2/e37583200d0d67abec74538df41f1909. Cited on page 36.

[99] G.ILGMANN. *The essence of railways*. G.Ilgmann, 2007. Cited on page 8.

[100] A. GILLE, M. KLEMENZ & T. SIEFER. *Applying multiscaling analysis to detect capacity resources in railway networks*, chap. A 7, pp. 73–82. Timetable Planning and Information Quality. WIT Press, 2010. Cited on page 56.

[101] J.-W. GOOSSENS, S. P. M. VAN HOESEL & L. G. KROON. *On solving multi-type railway line planning problems*. European Journal of Operational Research **168**(2):403–424, 2006. Cited on page 25.

[102] M. F. GORMAN. *Statistical estimation of railroad congestion delay*. Transportation Research Part E **45**(3):446–456, 2009. Cited on page 4.

[103] M. GRÖNKVIST. *The Tail Assignment Problem*. PhD thesis, Chalmers University of Technology and Göteborg University, 2005. Cited on pages 10, 14.

[104] M. GRÖTSCHEL, L. LOVÁSZ & A. SCHRIJVER. *Geometric Algorithms and Combinatorial Optimization*, vol. 2 of *Algorithms and Combinatorics*. Springer, 1988. ISBN 3-540-13624-X, 0-387-13624-X (U.S.). Cited on pages 9, 119.

[105] M. GRÖTSCHEL, S. O. KRUMKE & J. RAMBAU. *Online Optimization of Large Scale Systems*. Springer, Sept. 2001. ISBN 3-540-42459-8. Cited on page 10.

[106] T. GROVES. *Incentives in Teams*. Econometrica **41**:617–631, 1973. Cited on page 172.

[107] M. HABIB, R. M. MCCONNELL, C. PAUL & L. VIENNOT. *Lex-bfs and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing*. Theor. Comput. Sci. **234**(1-2):59–84, 2000. Cited on page 109.

[108] T. HANNE & R. DORNBERGER. *Optimization problems in airline and railway planning - a comparative survey*. In *Proceedings of the The Third International Workshop on Advanced Computational Intelligence*, 2010. Cited on page 10.

[109] I. HANSEN. *State-of-the-art of railway operations research*, chap. A 4, pp. 35–47. Timetable Planning and Information Quality. WIT Press, 2010. Cited on page 57.

[110] O. HAPPEL. *Ein Verfahren zur Bestimmung der Leistungsfähigkeit der Bahnhöfe*. PhD thesis, RWTH Aachen, 1950. Cited on page 61.

[111] O. HAPPEL. *Sperrzeiten als Grundlage für die Fahrplankonstruktion*. Eisenbahntechnische Rundschau (ETR) pp. 79–90, 1959. Cited on page 61.

[112] S. HARROD. *Modeling network transition constraints with hypergraphs*. Transportation Science **10**:293–310, 2010. Cited on page 4.

[113] E. HELLY. *Über Mengen konvexer Körper mit gemeinschaftlichen Punkten*. Jahresber. Deutsch. Math. Verein. **32**:175–176, 1923. Cited on page 108.

[114] C. HELMBERG. Semidefinite programming for combinatorial optimization. Technical report, Zuse Institute Berlin, October 2000. also habilitation thesis. Cited on page 135.

[115] A. HIGGINS, E. KOZAN & L. FERREIRA. *Heuristic techniques for single line train scheduling*. Journal of Heuristics **3**(1):43–62, 1997. ISSN 1381-1231. Cited on page 36.

[116] J.-B. HIRIART-URRUTY & C. LEMARÉCHAL. *Convex Analysis and Minimization Algorithms I*, vol. 305 of *A Series of Comprehensive Studies in Mathematics*. Springer-Verlag, 1993. Cited on page 135.

[117] J.-B. HIRIART-URRUTY & C. LEMARÉCHAL. *Convex Analysis and Minimization Algorithms II*, vol. 306 of *A Series of Comprehensive Studies in Mathematics*. Springer-Verlag, 1993. Cited on page 135.

[118] L. HURWICZ. *On informationally decentralized systems*. In C. B. MCGUIRE & R. RADNER, (Eds.), *Decision and Organization: A Volume in Honor of Jacob Marschak*. North-Holland, Amsterdam, 1972. Cited on page 172.

[119] CPLEX 12.2.0.2. *IBM ILOG CPLEX Optimization Studio*. IBM, 2011. URL http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/. Cited on page 149.

[120] S. IRNICH & G. DESAULNIERS. *Shortest Path Problems with Resource Constraints*, chap. 2, pp. 33–65. GERAD 25th Anniversary Series. Springer, 2005. Cited on page 50.

[121] S. IRNICH, G. DESAULNIERS, J. DESROSIERS & A. HADJAR. *Path-reduced costs for eliminating arcs in routing and scheduling*. INFORMS Journal on Computing **22**(2):297–313, 2010. Cited on page 50.

[122] J. JESPERSEN-GROTH, D. POTTHOFF, J. CLAUSEN, D. HUISMAN, L. KROON, G. MAROTI & M. NIELSEN. Disruption management in passenger railway transportation. Econometric Institute Report EI

2007-05, Erasmus University Rotterdam, Econometric Institute, Jan. 2007. URL http://ideas.repec.org/p/dgr/eureir/1765008527.html. Cited on page 15.

[123] J. JESPERSEN-GROTH, D. POTTHOFF, J. CLAUSEN, D. HUISMAN, L. G. KROON, G. MARÓTI & M. N. NIELSEN. *Disruption management in passenger railway transportation.* In Ahuja, Möhring & Zaroliagis (2009) [7], pp. 399–421. ISBN 978-3-642-05464-8. Cited on page 11.

[124] K. C. JHA, R. K. AHUJA & G. ŞAHIN. *New approaches for solving the block-to-train assignment problem.* Networks **51**(1):48–62, 2008. ISSN 0028-3045. Cited on pages 18, 19.

[125] D. JOVANOVIC & P. T. HARKER. *Tactical scheduling of rail operations.* Transportation Science **25**:46–64, 1991. Cited on pages 37, 38.

[126] D. KIM & C. BARNHART. *Transportation service network design: Models and algorithms.* In N. H. M. WILSON, (Ed.), *Proc. of the Seventh International Workshop on Computer-Aided Scheduling of Public Transport (CASPT), Boston, USA, 1997*, vol. 471 of *Lecture Notes in Economics and Mathematical Systems*, pp. 259–283. Springer-Verlag, Berlin, Heidelberg, 1997. Cited on page 18.

[127] K. C. KIWIEL. *Proximal bundle methods.* Mathematical Programming **46**(123):105–122, 1990. Cited on pages 136, 141.

[128] K. C. KIWIEL. *Approximation in proximal bundle methods and decomposition of convex programs.* Journal of Optimization Theory and applications **84**(3):529–548, 1995. Cited on pages 136, 141.

[129] S. G. KLABES. *Algorithmic railway capacity allocation in a competitive European railway market.* PhD thesis, RWTH Aachen, 2010. Cited on pages xxv, 11, 13, 29, 31, 39, 40, 57, 61, 62, 90.

[130] M. KLEMENZ & S.SCHULTZ. *Modelling aspects of a railway slot allocation.* In *2nd International Seminar on Railway Operations Modelling and Analysis*, 2007. Cited on page 41.

[131] W. KLEMT & W. STEMME. *Schedule synchronization for public transit networks.* In *Computer-Aided Transit Scheduling*, pp. 327–335. Springer-Verlag, New York, 1988. Cited on page 34.

[132] N. KLIEWER, T. MELLOULI & L. SUHL. *A time-space network based exact optimization model for multi-depot bus scheduling.* European Journal of Operational Research **175**(3):1616–1627, December 2006. URL http://ideas.repec.org/a/eee/ejores/v175y2006i3p1616-1627.html. Cited on pages 43, 96.

[133] V. KLIMA & A. KAVICKA. *Simulation support for railway infrastructure design and planning processes.* In *In Computers in Railways VII*, pp. 447–456. WIT Press, 2000. Cited on page 18.

[134] T. KOCH, A. MARTIN & T. ACHTERBERG. *Branching rules revisited.* Operations Research Letters **33**:42–54, 2004. Cited on page 142.

[135] A. KOKOTT & A. LÖBEL. Experiments with a dantzig-wolfe decomposition for multiple-depot vehicle scheduling problems. Technical Report ZIB Report 97-16, Zuse-Institut Berlin, Takustr. 7, 14195 Berlin, 1997. URL `http://www.zib.de/PaperWeb/abstracts/SC-97-16`. Cited on page 43.

[136] S. C. KONTOGIANNIS & C. D. ZAROLIAGIS. *Robust line planning under unknown incentives and elasticity of frequencies.* In *ATMOS*, 2008. Cited on page 90.

[137] C. KOPPER. *Zu lange, zu groß, zu teuer*, 2010. URL `http://www.zeit.de/2010/42/Bahn-Neubaustrecken`. Cited on page 17.

[138] C. KRÄUCHI & U. STÖCKLI. *Mehr Zug fr die Schweiz. Die Bahn-2000-Story (More train for Switzerland. The Rail 2000-Story).* Zürich: AS-Verlag, 2004. Cited on page 18.

[139] L. KROON, R. DEKKER, G. MAROTI, M. RETEL HELMRICH & M. J. VROMANS. *Stochastic improvement of cyclic railway timetables.* SSRN eLibrary , 2006. Cited on pages 34, 128.

[140] L. KROON, D. HUISMAN, E. ABBINK, P.-J. FIOOLE, M. FISCHETTI, G. MARÓTI, A. SCHRIJVER, A. STEENBEEK & R. YBEMA. *The new dutch timetable: The or revolution.* Interfaces **39**(1):6–17, 2009. ISSN 0092-2102. Cited on pages 2, 17, 34.

[141] L. G. KROON & L. W. P. PEETERS. *A variable trip time model for cyclic railway timetabling.* Transportation Science **37**(2):198–212, May 2003. Cited on page 34.

[142] L. G. KROON, R. DEKKER & M. J. C. M. VROMANS. *Cyclic railway timetabling: A stochastic optimization approach.* In F. GERAETS, L. G. KROON, A. SCHÖBEL, D. WAGNER & C. D. ZAROLIAGIS, (Eds.), *ATMOS*, vol. 4359 of *Lecture Notes in Computer Science*, pp. 41–66. Springer, 2004. ISBN 978-3-540-74245-6. Cited on page 34.

[143] A. LAMATSCH. *An approach to vehicle scheduling with depot capacity constraints.* In M. DESROCHERS & J.-M. ROUSSEAU, (Eds.), *Computer-Aided Transit Scheduling*, Lecture Notes in Economics and Mathematical Systems. Springer Verlag, 1992. Cited on page 96.

[144] S. LAN, J.-P. CLARKE & C. BARNHART. *Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions.* Transportation Science **40**(1):15–28, 2006. Cited on page 6.

[145] A. LANDEX, B. SCHITTENHELM, A. KAAS & J. SCHNEIDER-TILLI. *Capacity measurement with the UIC 406 capacity method.* In A. TOMII, J. ALLAN, E. ARIAS, C. BREBBIA, C. GOODMAN, A. RUMSEY & G. SCIUTTO, (Eds.), *Computers in Railways XI.* WIT Press, 2008. Cited on page 57.

[146] Y. LEE & C.-Y. CHEN. *A heuristic for the train pathing and timetabling problem.* Transportation Research Part B: Methodological **43**(8-9):837 – 851, 2009. ISSN 0191-2615. URL http://www.sciencedirect.com/science/article/B6V99-4VXT0P3-1/2/1cce3f2565ca4b86cb04a608124b7c36. Cited on page 36.

[147] C. LEMARÉCHAL. *Lagrangian relaxation.* In *Computational Combinatorial Optimization*, pp. 112–156, 2001. Cited on page 135.

[148] C. LIEBCHEN. *Periodic Timetable Optimization in Public Transport.* PhD thesis, Technische Universität Berlin, 2006. Cited on pages 10, 13, 33, 34.

[149] C. LIEBCHEN. *The first optimized railway timetable in practice.* Transportation Science **42**(4):420–435, 2008. Cited on pages 2, 34.

[150] C. LIEBCHEN & R. H. MÖHRING. *The modeling power of the periodic event scheduling problem: Railway timetables - and beyond.* In *ATMOS*, pp. 3–40, 2004. Cited on page 34.

[151] C. LIEBCHEN, M. SCHACHTEBECK, A. SCHÖBEL, S. STILLER & A. PRIGGE. Computing delay resistant railway timetables. Technical report, ARRIVAL Project, October 2007. Cited on page 128.

[152] C. LIEBCHEN, M. E. LÜBBECKE, R. H. MÖHRING & S. STILLER. *The concept of recoverable robustness, linear programming recovery, and railway applications.* In Ahuja, Möhring & Zaroliagis (2009) [7], pp. 1–27. ISBN 978-3-642-05464-8. Cited on pages 34, 128.

[153] C. LIEBCHEN, M. SCHACHTEBECK, A. SCHÖBEL, S. STILLER & A. PRIGGE. *Computing delay resistant railway timetables.* Comput. Oper. Res. **37**(5):857–868, 2010. ISSN 0305-0548. Cited on page 34.

[154] T. LINDNER. *Train schedule optimization in public rail transport.* PhD thesis, TU Braunschweig, 2000. Cited on page 34.

[155] A. LÖBEL. *Optimal Vehicle Scheduling in Public Transit.* Shaker Verlag, Aachen, 1997. URL http://www.shaker.de/Online-Gesamtkatalog/Details.idc?ISBN=3-8265-3504-9. Ph.D. thesis, Technische Universität Berlin. Cited on page 14.

[156] M. LÜBBECKE & J. DESROSIERS. *Selected topics in column generation.* Oper. Res. **53**(6):1007–1023, 2005. Cited on pages 143, 161.

[157] S. G. LUKAC. Holes, antiholes and maximal cliques in a railway model for a single track. Technical Report ZIB Report 04-18, Zuse-Institut Berlin, Takustr. 7, 14195 Berlin, 2004. URL http://www.zib.de/PaperWeb/abstracts/ZR-04-18. Cited on page 109.

[158] R. LUSBY. *Optimization Methods for Routing Trains Through Railway Junctions.* PhD thesis, The University of Auckland, 2008. Cited on pages 39, 40, 88, 90, 103, 142.

[159] R. LUSBY, J. LARSEN, M. EHRGOTT & D. RYAN. *Railway track allocation: models and methods*. OR Spectrum , December 2009. URL http://dx.doi.org/10.1007/s00291-009-0189-0. Cited on pages 10, 33, 57, 90.

[160] R. MARSTEN. *Crew planning at delta airlines*. Talk at the 15th Int. Symp. Math. Prog., 1994. Cited on page 48.

[161] D. MIDDELKOOP & M. BOUWMAN. *Train network simulator for support of network wide planning of infrastructure and timetables*. In *In Computers in Railways VII*, pp. 267–276. WIT Press, 2000. Cited on page 18.

[162] P. MILGROM. *Putting Auction Theory to Work*. Cambridge University Press, 2004. URL http://econpapers.repec.org/RePEc:cup:cbooks:9780521536721. Cited on page 171.

[163] M. MONTIGEL. *Modellierung und Gewährleistung von Abhängigkeiten in Eisenbahnsicherungsanlagen*. PhD thesis, ETH Zurich, 1994. Cited on page 58.

[164] A. MURA. Trassenauktionen im schienenverkehr. Diploma thesis, Technische Universität Berlin, 2006. URL http://www.zib.de/Optimization/Projects/Traffic/TrassenB/diplom_mura.ps.gz. Cited on pages 29, 127, 150, 172.

[165] K. NACHTIGALL & J. OPITZ. *Solving periodic timetable optimisation problems by modulo simplex calculations*. In *ATMOS*, 2008. Cited on page 34.

[166] A. NASH & D. HUERLIMANN. *Railroad simulation using OpenTrack*. In *Computer Aided Design, Manufacture and Operation in the Railway and other Advanced Transit Systems*, 2004. Cited on page 60.

[167] G. L. NEMHAUSER & L. A. WOLSEY. *Integer and Combinatorial Optimization*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, New York, 1988. Cited on pages 9, 132.

[168] F. NIEKERK & H. VOOGD. *mpact assessment for infrastructure planning: some dutch dilemmas*. Environmental Impact Assessment Review **19**:21–36, 1999. Cited on page 18.

[169] M. A. ODIJK. *Railway Timetable Generation*. PhD thesis, TU Delft, 1997. Cited on page 34.

[170] M. A. ODIJK, H. E. ROMEIJN & H. VAN MAAREN. *Generation of classes of robust periodic railway timetables*. Computers & OR **33**: 2283–2299, 2006. Cited on page 34.

[171] E. OLIVEIRA & B. M. SMITH. *A combined constraint-based search method for single-track railway scheduling problem*. In P. BRAZDIL & A. JORGE, (Eds.), *EPIA*, vol. 2258 of *Lecture Notes in Computer Science*, pp. 371–378. Springer, 2001. ISBN 3-540-43030-X. Cited on page 36.

[172] OpenTrack. *OpenTrack*, 2010. URL http:\www.opentrack.ch. http:\www.opentrack.ch. Cited on page 58.

[173] J. PACHL. *Systemtechnik des Schienenverkehrs*. Teubner Verlag, Stuttgart-Leipzig-Wiesbaden, 3 edition, 2002. Cited on page 60.

[174] J. PACHL. *Timetable design principles*, chap. 2, pp. 9–42. Railway Timetable & Traffic. Eurailpress, 2008. Cited on page 61.

[175] D. C. PARKES & L. H. UNGAR. *An auction-based method for decentralized train scheduling*. In *Proc. 5th International Conference on Autonomous Agents (AGENTS-01)*, pp. 43–50, 2001. Cited on page 174.

[176] M. PEETERS & L. G. KROON. *Circulation of railway rolling stock: a branch-and-price approach*. Computers & OR **35**(2):538–556, 2008. Cited on page 14.

[177] D. POTTHOFF, D. HUISMAN & G. DESAULNIERS. Column generation with dynamic duty selection for railway crew rescheduling. Econometric Institute Report EI 2008-28, Erasmus University Rotterdam, Econometric Institute, Dec. 2008. URL http://ideas.repec.org/p/dgr/eureir/1765014423.html. Cited on pages 11, 15, 52.

[178] G. POTTHOFF. *Verkehrsströmungslehre, Band 1- Die Zugfolge auf Strecken und in Bahnhöfen, 3. Auflage*. transpress Verlag Berlin, 1980. Cited on page 56.

[179] ProRail. *ProRail*, 2010. URL http://www.prorail.nl. http://www.prorail.nl. Cited on page 91.

[180] A. RADTKE. *Infrastructure modelling*, chap. 3, pp. 43–57. Railway Timetable & Traffic. Eurailpress, 2008. Cited on pages 55, 56.

[181] G. REINELT. *Tsplib - a traveling salesman problem library*. ORSA Journal on Computing **3**:376–384, 1991. Cited on page 64.

[182] N. J. REZANOVA & D. M. RYAN. *The train driver recovery problem- a set partitioning based model and solution method*. Comput. Oper. Res. **37**(5):845–856, 2010. ISSN 0305-0548. Cited on pages 11, 15.

[183] J. H. RODRIGUEZ. *A constraint programming model for real-time train scheduling at junctions*. Transportation Research Part B: Methodological **41**(2):231–245, 2007. URL http://econpapers.repec.org/RePEc:eee:transb:v:41:y:2007:i:2:p:231-245. Cited on page 36.

[184] A. ROMEIN, J. TRIP & J. DE VRIES. *The multi-scalar complexity of infrastructure planning: evidence from the dutch-flemish eurocorridor*. Journal of Transport Geography **3**(11):205–213, 2003. Cited on page 18.

[185] R. SAUDER & W. WESTERMAN. *Computer aided train dispatching: Decision support through optimization*. Interfaces **13**(6):24–37, 1983. Cited on page 37.

[186] B. SCHITTENHELM. *Quantitative methods to evaluate timetable attractiveness*. In I. HANSEN, E. WENDLER, U. WEIDMANN, M. LÜTHI, J. RODRIGUEZ, S. RICCI & L. KROON, (Eds.), *Proceedings of the 3rd International Seminar on Railway Operations Modelling and Analysis - Engineering and Optimisation Approaches*, Zürich, Switzerland, 2009. Cited on pages 25, 92.

[187] T. SCHLECHTE. Das Resource-Constraint-Shortest-Path-Problem und seine Anwendung in der ÖPNV-Dienstplanung. Master's thesis, Technische Universität Berlin, 2003. Cited on page 120.

[188] T. SCHLECHTE & R. BORNDÖRFER. *Balancing efficiency and robustness - a bi-criteria optimization approach to railway track allocation*. In M. EHRGOTT, B. NAUJOKS, T. STEWART & J. WALLENIUS, (Eds.), *MCDM for Sustainable Energy and Transportation Systems*. Lecture Notes in Economics and Mathematical Systems, 2008. URL http://opus.kobv.de/zib/volltexte/2008/1105/. Cited on pages ii, 90, 126, 129, 131.

[189] T. SCHLECHTE & A. TANNER. *Railway capacity auctions with dual prices*. In *Selected Proceedings of the 12th World Conference on Transport Research*, 2010. ISBN 978-989-96986-1-1. URL http://opus.kobv.de/zib/volltexte/2010/1233/. submitted to Special Issue of Research in Transportation Economics 24.2.2011. Cited on pages ii, 12, 174, 175.

[190] T. SCHLECHTE, R. BORNDÖRFER, B. EROL, T. GRAFFAGNINO & E. SWARAT. *Aggregation methods for railway networks*. In I. HANSEN, E. WENDLER, S. RICCI, D. PACCIARELLI, G. LONGO & J. RODRIGUEZ, (Eds.), *Proceedings of 4th International Seminar on Railway Operations Modelling and Analysis (IAROR)*, vol. 4, 2011. Cited on pages ii, 55, 180.

[191] A. SCHÖBEL & A. KRATZ. *A bicriteria approach for robust timetabling*. In Ahuja, Möhring & Zaroliagis (2009) [7], pp. 119–144. ISBN 978-3-642-05464-8. Cited on page 129.

[192] A. SCHÖBEL & S. SCHOLL. *Line planning with minimal traveling time*. In L. G. KROON & R. H. MÖHRING, (Eds.), *5th Workshop on Algorithmic Methods and Models for Optimization of Railways*, Dagstuhl, Germany, 2006. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany. ISBN 978-3-939897-00-2. URL http://drops.dagstuhl.de/opus/volltexte/2006/660. Cited on page 25.

[193] A. SCHRIJVER. *Theory of Linear and Integer Programming*. Interscience series in discrete mathematics and optimization. Wiley, 1998. Cited on page 132.

[194] A. SCHRIJVER & A. STEENBECK. Dienstregelingontwikkeling voor railned (timetable construction for railned. Technical report, Center

for Mathematics and Computer Science, 1994. Cited on page 34.

[195] K. Schultze. *Modell für die asynchrone Simulation des Betriebes in Teilen des Eisenbahnnetzes*. PhD thesis, RWTH Aachen, 1985. Cited on page 56.

[196] W. Schwanhäusser. *Die Bemessung der Pufferzeiten im Fahrplangefüge der Eisenbahn*. PhD thesis, RWTH Aachen, 1974. Cited on page 56.

[197] W. Schwanhäusser, I. Gast, K. Schultze & O. Brünger. Programmfamilie SLS Benutzerhandbuch. Technical report, Deutsche Bundesbahn, 1992. Cited on page 62.

[198] Y. Semet & M. Schoenauer. *An efficient memetic, permutation-based evolutionary algorithm for real-world train timetabling*. In *Congress on Evolutionary Computation*, pp. 2752–2759. IEEE, 2005. ISBN 0-7803-9363-5. Cited on page 36.

[199] P. Serafini & W. Ukovich. *A mathematical for periodic scheduling problems*. SIAM J. Discret. Math. **2**(4):550–581, 1989. ISSN 0895-4801. Cited on page 34.

[200] B. Sewcyk. *Makroskopische Abbildung des Eisenbahnbetriebs in Modellen zur langfristigen Infrastrukturplanung*. PhD thesis, Leibniz U Hannover, 2004. Cited on page 41.

[201] E. Silva de Oliveira. *Solving single-track railway scheduling problem using constraint programming*. PhD thesis, University of Leeds, School of Computing, 2001. Cited on page 36.

[202] A. Soyster. *Convex programming with set-inclusive constraints and applications to inexact linear programming*. Oper. Res. **21**:1154–1157, 1973. Cited on page 128.

[203] I. Steinzen, V. Gintner, L. Suhl & N. Kliewer. *A time-space network approach for the integrated vehicle-and crew-scheduling problem with multiple depots*. Transportation Science **44**(3):367–382, 2010. ISSN 1526-5447. Cited on page 43.

[204] R. Subramanian, R. Sheff, J. Quillinan, D. Wiper & R. Marsten. *Coldstart: Fleet assignment at delta air lines*. Interfaces **24**(1):104–120, 1994. Cited on pages 142, 147.

[205] L. Suhl, V. Dück & N. Kliewer. *Increasing stability of crew schedules in airlines*. In C. Barnhart, U. Clausen, U. Lauther & R. H. Möhring, (Eds.), *Models and Algorithms for Optimization in Logistics*, number 09261 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2009. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany. URL http://drops.dagstuhl.de/opus/volltexte/2009/2178. Cited on page 6.

[206] B. Szpigel. *Optimal train scheduling on a single track railway*. In *Proceedings of IFORS Conference on Operational Research'72*, number 72 in 6, pp. 343–352, 1973. Cited on pages 36, 37, 38, 40.

[207] Trasse Schweiz AG. *Business report 2009*, 2009. URL http://www.trasse.ch. http://www.trasse.ch. Cited on pages 13, 91.

[208] TTPlib. *TTPlib-Homepage*, 2008. URL http://ttplib.zib.de. http://ttplib.zib.de. Cited on pages xxvii, 3, 154.

[209] W. VICKREY. *Counterspeculation, auctions, and competitive sealed tenders*. The Journal of Finance **16**(1):8–37, 1961. URL http://www.jstor.org/stable/2977633. Cited on page 172.

[210] D. VILLENEUVE, J. DESROSIERS, M. E. LÜBBECKE & F. SOUMIS. *On compact formulations for integer programs solved by column generation*. Annals OR **139**(1):375–388, 2005. Cited on page 133.

[211] D. WEDELIN. *An algorithm for a large scale 0-1 integer programming with application to airline crew scheduling*. Annals of Operations Research **57**:283–301, 1995. Cited on pages 48, 142, 145.

[212] O. WEIDE, D. RYAN & M. EHRGOTT. *An iterative approach to robust and integrated aircraft routing and crew scheduling*. Comput. Oper. Res. **37**(5):833–844, 2010. ISSN 0305-0548. Cited on page 129.

[213] S. WEIDER. *Integration of Vehicle and Duty Scheduling in Public Transport*. PhD thesis, TU Berlin, 2007. Cited on pages x, xiv, xxvii, 3, 10, 16, 96, 120, 135, 141, 142, 143, 146.

[214] E. WENDLER. *Influence of ETCS on the capacity of lines*. In *Compendium on ERTMS*, Compendium on ERTMS: European Rail Traffic Management System. Eurailpress, 2009. Cited on page 63.

[215] T. WHITE & A. KRUG, (Eds.). *Managing Railroad Transportation*. VTD Rail Publishing, 2005. ISBN 0-9719915-3-7. Cited on page 4.

[216] J. W. ZHENG, H. T. KIN & M. B. HUA. *A study of heuristic approach on station track allocation in mainline railways*. International Conference on Natural Computation **4**:575–579, 2009. Cited on page 36.

[217] X. ZHOU & M. ZHONG. *Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds*. Transportation Research Part B: Methodological **41**(3):320–341, March 2007. URL http://ideas.repec.org/a/eee/transb/v41y2007i3p320-341.html. Cited on page 36.

[218] E. ZHU, T. G. CRAINIC & M. GENDREAU. Integrated service network design in rail freight transportation. Research Report CIRRELT-2009-45, CIRRELT, Montréal, Canada, 2009. Cited on page 18.

[219] G. M. ZIEGLER. *Lectures on Polytopes*. Springer, 1995. Cited on page 121.

[220] P. J. ZWANEVELD, L. G. KROON, H. E. ROMEIJN, M. SALOMON, S. DAUZERE-PERES, S. P. M. VAN HOESEL & H. W. AMBERGEN. *Routing Trains Through Railway Stations: Model Formulation and Algorithms*. Transportation Science **30**(3):181–194,

1996. URL http://transci.journal.informs.org/cgi/content/abstract/30/3/181. Cited on page 57.

[221] P. J. Zwaneveld, L. G. Kroon & S. P. M. van Hoesel. *Routing trains through a railway station based on a node packing model*. European Journal of Operational Research **128**(1):14–33, January 2001. URL http://ideas.repec.org/a/eee/ejores/v128y2001i1p14-33.html. Cited on page 57.

Lebenslauf

Thomas Schlechte

geboren am 10.03.1979 in Halle an der Saale

1985 bis 1986:	Besuch der Grundschule in Halle
1986 bis 1991:	Besuch der Grundschule in Berlin
1991 bis 1998:	Besuch des Descartes Gymnasiums in Berlin
1998 bis 2004:	Studium der Mathematik an der Technischen Universität Berlin
Seit 2004:	Wissenschaftlicher Mitarbeiter am Zuse Institute Berlin (ZIB)