Diss. ETH No. 16398

Stability of Timetables and Train Routings through Station Regions

A dissertation submitted to the

SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of

DOCTOR OF SCIENCES

presented by

THOMAS MICHAEL HERRMANN
Dipl. Math. ETH

born 31st July 1975
citizen of Rohrbach (BE)

accepted on the recommendation of

Prof. Dr. Hans-Jakob Lüthi, examiner
Prof. Dr. Ulrich Weidmann, co-examiner
Prof. Dr. Kay Axhausen, co-examiner

2006

To my parents

Acknowledgements

While working at the Institute for Operations Research at the Swiss Federal Institute of Technology I had the great opportunity to learn about the railway transportation sector and its challenges. I am very grateful to Prof. Hans-Jakob Lühti, the referee of my thesis, for his continuous supervision and the granted freedom during the course of this project.

I owe many thanks to Prof. Ulrich Weidmann and Prof. Kay Axhausen for their willingness to referee my thesis, their helpful suggestions and their friendly support.

I am also infinitely indebted to my project partner Dan Burkolter for his invaluable input which undoubtedly improved this thesis. I much appreciated his critical and constructive reviews and his sense of humor.

Special thanks go to Gabrio Caimi for his beneficial contributions and implementations. I am sure he will successfully continue to advance the railway project.

I wish to express my gratitude to Oskar Stalder, Dr. Felix Laube, Dr. Raimond Wüst and Thomas Graffagnino from the Swiss Federal Railways who raised interesting research problems and established an atmosphere for insightful discussions.

Many thanks also to Dr. Maurice Cochand for advising the project in the initial phase and Dr. Jens Henoch for numerous lively discussions and his indispensable assistance.

A very friendly word of thanks goes to Hilda Fritze-Vomvoris, Rico Zenklusen and Jörg Doege for proofreading the thesis.

I would like to thank all my present and former colleagues at the Swiss Federal Institute of Technology who enriched my time at IFOR.

Finally, my warmest thanks go to my wife Nathalie, my family and friends for their constant support and encouragement.

Abstract

During the last few years, railway traffic has increased considerably; moreover, it is expected that railroad transportation will further grow for both passenger and freight transportation. These developments create needs to optimize both the utilization of the existing infrastructure and the coordination tasks inside the railroad company. Thanks to developments in computer science, optimization techniques, and intelligent resource management, railway schedules with increased frequencies can be generated nowadays. Railway operators expect the capacity bottlenecks of the railway system especially near main stations, since main train lines are intersecting there. Tight timetables, however, are exposed to train delays to a greater extent than less dense schedules.

The thesis at hand describes stability measures of timetables and the highly related topic of the search for train routings within station regions. As the quality of service should not suffer when introducing new train services, the question of the stability of a timetable is of crucial interest while designing denser timetables. Unforeseen events may require partial modifications of the plans in real-time and therefore re-scheduling procedures should be already taken into account when designing a new timetable. Moreover, re-scheduling procedures should be as easy to implement as possible. Fundamentally, the timetable's ability to absorb some disturbances trades off against full exploitation of available capacity. With the help of evaluation functions, timetables can be examined regarding their likelihood to fail, their sensitivity against disruptions of the schedule, or their efficiency to recover from deviations.

Separating the problem of train routings in exact topologies from the saturation of the available capacity and the generation of a timetable in an aggregated topology results in a two level approach. On the upper level a timetable for an intended train

service is generated using an aggregated topology. The task on this level is to develop a timetable whose periodicity is as small as possible in order to use the network to full capacity while respecting safety restrictions and the intention of the train service. On the lower level, exact topologies are used in order to decide feasibility of the previously generated, tentative timetables and to analyze the derived schedules. As this thesis mainly deals with stability of timetables, it is consistently assumed that a timetable for the aggregated topology is available.

By examining the routing alternatives, which are tremendously high in station regions, the feasibility problem is modeled as an independent set problem. The node set corresponds to all possible routes of the trains and two nodes are connected by an edge, if their corresponding routes are mutually exclusive. The independent set problem is then solved by applying a fixed point heuristic.

The probability that the routes of two different trains are incompatible can be calculated by assuming certain delay patterns for arriving and departing trains. The previous graph model is then extended by the introduction of additional edges whose weights are set to the probability that the corresponding routes are incompatible. Stability measures are then expressed as certain properties of the extended graph. Moreover, a stability measure that is independent of any delay distributions is also introduced. In a second step, these stability measure functions are used to state different optimization problems that are then solved by a random restart local search heuristic.

In order to test the methodology, the Bern station region has been used. Depending on the applied optimization problem and the underlying delay patterns, the trains are scheduled to travel on different routes through the network. Results show that the tighter the timetable becomes the more important is the design of the railway system and the coordination of suitable track topologies, meaningful train service intentions, the dense schedules, elaborate routings, and actively managed train delays.

Zusammenfassung

Die Beförderung von Personen und Gütern auf der Schiene hat in den letzten Jahren stark zugenommen; gemeinhin wird erwartet, dass dieser Trend in den nächsten Jahren anhält. Diese Entwicklung führt dazu, dass nicht nur die Auslastung der bestehenden Infrastruktur erhöht werden muss, sondern dass auch die Betriebsabläufe innerhalb eines Bahnunternehmens optimiert werden müssen. Dank den Fortschritten in der Computertechnologie, den Optimierungsmethoden und dem Management von Ressourcen können heute schon Fahrpläne mit erhöhten Zugfrequenzen generiert werden. Die Bahnunternehmen erwarten deshalb auf Grund der vielen Zugfahrten, dass inbesondere in der Nähe von Hauptbahnhöfen und Knotenpunkten die vorhandene Kapazität zu einem Engpass werden wird. Darüber hinaus sind dichtere Fahrpläne auch einem grösseren Risiko bezüglich Verspätungen ausgesetzt.

In dieser Arbeit werden Stabilitätsmasse für Fahrpläne eingeführt und das eng verwandte Thema der Routenplanung innerhalb von Bahnhofsregionen untersucht. Durch die Einführung von neuen Zugverbindungen soll die Qualität gegenüber den Passagieren nicht leiden. Deshalb ist die Stabilität eines Fahrplanes bei der Konstruktion von neuen und dichteren Fahrplänen von besonderem Interesse. Da unvorhersehbare Ereignisse die erarbeiteten Pläne stören können, müssen Umplanungsmassnahmen soweit wie möglich bereits bei der Erstellung der Pläne berücksichtigt werden und zusätzlich einfach zu implementieren sein. Die Fähigkeit eines Fahrplanes gewisse Störungen zu absorbieren, muss gegen die volle Auslastung der vorhandenen Kapazität abgewogen werden. Mit Hilfe von Bewertungsfunktionen können nun Fahrpläne auf ihre Eigenschaften analysiert werden. Von besonderem Interesse sind dabei die Wahrscheinlichkeit, dass ein Fahrplan neu errechnet werden muss, die Sensitivität gegenüber Verspätungen und die Effizienz beim Beheben von Störungen.

Es wird ein zweistufiges Modell eingeführt, in welchem die Berechnung eines Fahrplans und die Saturierung der vorhandenen Kapazität in einer aggregierten Topologie von der exakten Routenplanung für die Züge getrennt ist. In der oberen Stufe wird eine aggregierte Topologie verwendet, um für eine gegebene Leistungsabsicht Fahrpläne mit möglichst hoher Auslastung des Netzwerks zu berechnen. Deshalb wird unter Berücksichtigung von Sicherheitsmargen ein Fahrplan mit möglichst kleiner Periodizität entworfen. Auf der unteren Stufe hingegen werden lokale exakte Topologien verwendet, um zu entscheiden, ob ein provisorischer Fahrplan gültig ist, und um den Fahrplanablauf zu analysieren. Da das Hauptthema dieser Arbeit die Stabilität von Fahrplänen ist, wird durchweg angenommen, dass ein Fahrplan für die aggregierte Topologie zur Verfügung steht.

Im Zentrum der Routenplanung steht die Untersuchung der verschiedenen Planungsalternativen, die in Bahnhofsregionen typischerweise sehr zahlreich sind. Das entsprechende Entscheidungsproblem wird auf das Problem des Auffindens unabhängiger Mengen zurückgeführt. Dabei besteht die Knotenmenge aus der Menge aller möglicher Routen der Züge und zwei Knoten werden miteinander verbunden, falls sich die entsprechenden Routen gegenseitig ausschliessen. Dieses Problem der unabhängigen Menge wird mit Hilfe einer Fixpunktiterations-Heuristik gelöst.

Falls zusätzlich Verteilungen der Verspätungen von ankommenden und abfahrenden Zügen verfügbar sind, so kann die Wahrscheinlichkeit, dass zwei Routen inkompatibel werden, berechnet werden. Das vorherige Graphenmodell wird erweitert, indem zusätzliche Kanten eingeführt werden, deren Gewichte den Wahrscheinlichkeiten inkompatibler Routen entsprechen. Stabilitätsmasse können dann mit Hilfe von Eigenschaften des erweiterten Graphes ausgedrückt werden. Zusätzlich wird auch ein Stabilitätsmass eingeführt, das unabhängig von Verspätungswahrscheinlichkeiten ist. In einem zweiten Schritt werden diese Stabilitätsmasse als Zielfunktionen in verschiedene Optimierungsprobleme eingeführt, die wiederum mit Hilfe einer Heuristik (randomisierte Nachbarschaftssuche) gelöst werden.

An Hand der Bahnhofsregion Bern wird diese Methodik getestet. Je nach verwendetem Optimierungskriterium und verwendeten Verspätungsmustern verkehren die Züge auf verschiedenen Routen. Die Resultate zeigen: Je dichter ein Fahrplan ist, desto wichtiger sind Design und Koordination von angemessener Infrastruktur, sinnvollen Angebotsszenarien, dichten Fahrplänen, durchdachten Routenplanungen und dem aktiven Bewirtschaften von Zugsverspätungen.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Background and Motivation

> At school, new ideas are thrust at you every day. Out in the world, you'll have to find the inner motivation to search for new ideas on your own.
>
> *Bill Watterson (*1958)*

Since December 12th 2004 a trip from Zurich to Bern has taken 58 minutes—one hundred years ago two and a half hours were needed. This is the first time in the history of Swiss railway that a train needs less than an hour to connect these two main cities (see Figure 1.1).

Nowadays the main challenge is not to further reduce the travel time, but to provide good service to customers. Today the Swiss Federal Railways (SBB) transports about 250 million passengers and 55 million tons of freight each year. The overall length of Switzerland's public transportation network is almost 25'000 kilometers, while the SBB's track network is 3'000 kilometers long. In spite of this modest fraction of the network, the SBB trains account for 87% of all passenger-kilometers and 90% of tonnage-kilometers of railway transportation. Overall in public transportation, the SBB performs about the half of 24 billion passenger-kilometers per year.

At almost 800 stations an hourly or half-hourly regular-interval train service is provided. Two-thirds of all freight passing through the Swiss Alps is transported by train. Freight cars can be loaded and dispatched at over 650 stations.

Almost 30'000 employees work every day to make the trains run safely and punctually. So not only is the SBB the largest travel and transport company in Switzerland, it is also one of the biggest employers.

14% more trains were introduced on the track network with the timetable change in December 2004 and 90% of all train departures have changed as well (see

[Stalder and Laube, 2004]). At the same time the rail infrastructure has grown by 160 kilometers of new track. Usually the trains travel on a regular basis of 30 minutes, which means that every event repeats on a half-hourly basis. In contrast, 20 years ago—1982, at the introduction of a nationwide periodic timetable—the frequency interval was everywhere an interval of practically one hour.

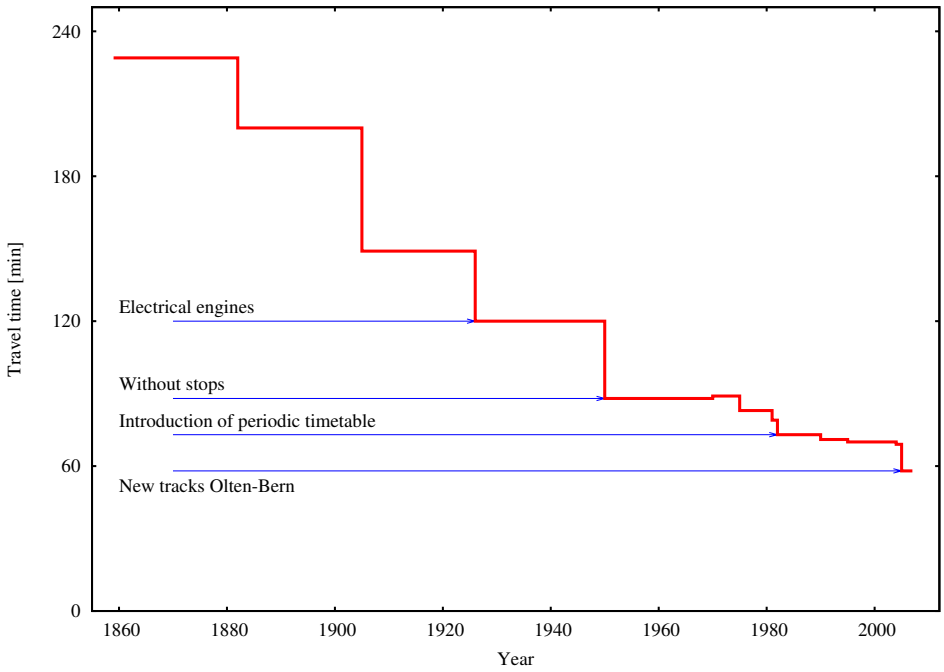

Figure 1.1: Travel time between Zurich and Bern. The first connection between Zurich and Bern by stagecoaches was introduced in 1735. Around 1840 the first regular connection by stagecoaches took 14 hours. In 1859 the first non-stop railway track was opened and a steam engine was able to cover the distance of about 120 kilometers in less than 4 hours. With the introduction of electrical engines the time was cut in half. Since 1950 and the introduction of new infrastructure, new engines, and new coaches the travel time could be halved again (see [SBB, 2004]).

The volume of rail traffic is expected to increase throughout the coming years. A sound service is required, which is essential in order for the railway companies to survive. The means to achieve such a level of service are manyfold: Investment in engines and coaches would increase the comfort and shorten travel time for passengers and cargo transport. Upgrading the track network however, would have an impact for both passenger and freight trains and their travel time, although new tracks are very expensive and should thus be avoided. Regardless of which investment is made, it is

done in order to increase the quality of service, *i. e.* to extend existing services and lines or to increase the capacity in certain regions.

Today Switzerland's public transportation network is more or less complete and there is no need to open many new stations. Hence infrastructure investment is undertaken to improve process flows and service level. The demand for more frequent services within different conurbations and between main cities is still increasing. Figure 1.2 shows the number of trains passing the stations before and after the December 2004 timetable change.

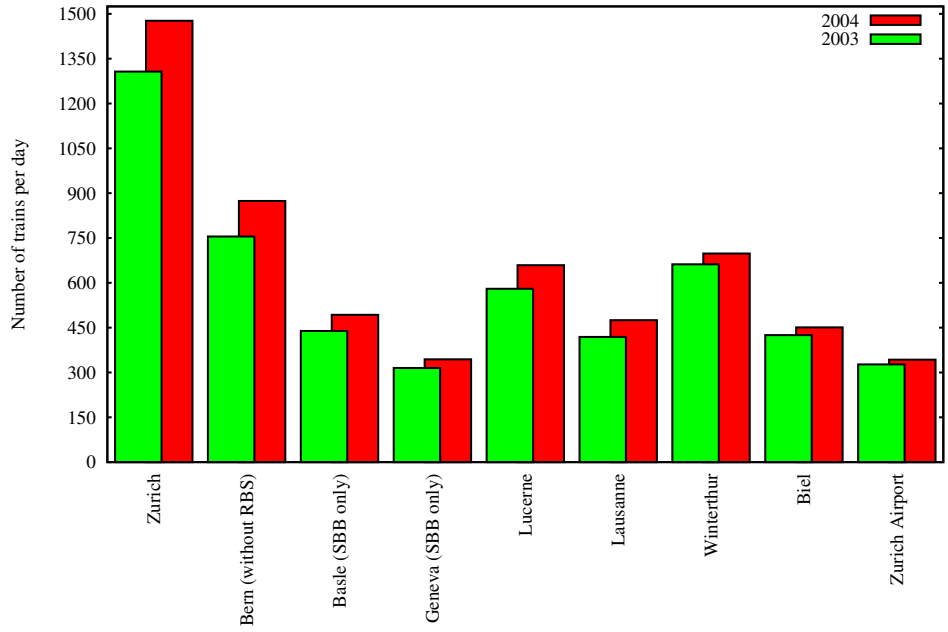

Figure 1.2: Train load in nine Swiss main stations, ordered by percentage of difference between the 2003 and 2004 timetables (see [SBB, 2004]).

As the traffic on the network is further increased and condensed, timetable planners will have to focus more and more on the interaction between the *utilization of the tracks* and the *stability of the timetable*. Fundamentally, the timetable's ability to absorb train delays trades off against the full exploitation of the available capacity. Hence the following question arises: *Can the number of trains be increased further without decreasing the quality of service?*

Track systems for railways and urban traffic are assumed to increase but the main stations—which are usually located in the center of a city—cannot expand. Therefore, an increase in the number of trains will cause problems in the important stations as they approach their capacity limit. However, train stations are also the main source of delays—either trains have to wait for an incoming connecting train and hence have

a departure delay or the boarding of passengers is too slow. Hence, it is important to investigate operations in train stations and include the conclusions already in the planning phase. The more trains running in the system, the more vulnerable the timetable becomes when disruptions occur. Yet, the measures to quantify the different stability aspects of a timetable—especially for timetables in station regions—have to be identified. As capacity aspects of a track network are strongly correlated to the generation of timetables, a distinct separation between track utilization and stability considerations has to be made, since stability can only be measured if a timetable is available. Nevertheless, conclusions about the stability of a timetable may again be included in a new timetable generation procedure.

1.1 The Train Scheduling Problem

The planning of public rail transport systems raises many challenges, as many tasks have to be coordinated in order to develop convenient plans. Moreover, the tasks are highly dependent upon each other. The process of setting up a public rail transport system can be divided into several phases (see for example [Lindner, 2000], [Bussieck et al., 1998], and [Bussieck et al., 1997]).

The first phase is concerned with the layout of a suitable track network that meets the requirements for both passenger and cargo transport. Before considering a schedule, a railway company must know the needs of their customers. The cargo division wants to transport freight from various facilities to particular addressees. Often both the origins and the destinations are not located in the center of the cities but on the outskirts or in industrial areas. In contrast, the passenger division wants to transport people between city centers. Hence, the rail track network has to be designed in a way to meet both interests. However, the track networks are often almost sufficient and can thus be considered fixed today. Hence, track extensions are often built to improve rail traffic locally.

Once the track network has been designed, a train service has to be defined in a second phase. A line plan is constructed by aggregating the demand and by specifying the origins and destinations as well as intermediate stops (see for example [Bussieck et al., 1996] or [Lindner, 2000]). For public transport this system of train service is crucial for success. If the lines do not connect cities in an appropriate way, *i. e.* if the passenger has to change trains too often, then he or she chooses not to travel by train and goes by the car instead (see [Goverde, 1998b] and [IRSE, 2004]). Usually the passenger line systems are based on surveys from which so-called origin-destination matrices are constructed, which describe passenger volume between any two stations. The same is true for the cargo transport: If a company is not able to handle the requested service in a sufficiently short time, customers will not use rail transport.

After the lines have been fixed, the timetable, which is the backbone of every public rail transport service, has to be built in a third step. A timetable consists of arrival and departure times of the lines at certain points of the network (*e. g.* stations). This scheme is repeated periodically (usually hourly in Switzerland) and is constructed in such a way that no conflicts occur. This task is very difficult, as the complexity is tremendous. If an entire network is considered in detail, then the problem of finding a feasible schedule is too complicated—even for small networks. Hence, the network is often aggregated and feasibility is checked for local detailed topologies. Nowadays the timetable is usually constructed by hand, using some software support and a lot of experience. Much research deals with computer aided timetabling, such as [Burkolter, 2005], [Lindner, 2000], [Heusch et al., 1997], and [Serafini and Ukovich, 1989].

In order to construct a timetable, the planner assigns rolling stock to the served lines. Furthermore, he has to assure interchange possibilities at main stations in order to provide complete service. All this is summarized as the train service intention: A set of lines, each having fixed start and end points within the track network, their rolling stock, characterized by acceleration and braking profiles, and eventually some predefined passing or stopping points (*e. g.* platforms at stations). Finally, the arrival sequence of the lines at main stations might be fixed to allow for connections.

A timetable then defines a passing-time for each start, intermediate and end point of a line such that safety and interchange restrictions are met.

1.2 Topic of the Thesis

The thesis at hand examines the stability of draft timetables that are based on some given network layout and train service intentions. Henceforth, network and service intention are assumed to be fixed. The intended lines share common resources on the network. As track elements are utilized by many different trains, potential conflicts have to be detected as well as respected while generating timetables. However, recognition of conflicts is not trivial, as the following example shows.

For the purpose of illustration of the concepts, a small toy example has been generated, the so-called MATHWORLD. It includes many features that occur in larger examples taken from the real world. The network layout and the lines are shown in Figure 1.3. For safety reasons, it is assumed that two trains must have a time gap of at least one minute if they pass the same point in the network.

In detail, the train service intention for MATHWORLD consists of six lines that could be served by five trains. Nevertheless, there are six different itineraries: An intercity route (short IC+ and IC−) connects Abel-Heim and Cauchy-Ville via Dantzig-Stadt (no stop in Euler-Hausen). A first commuter train route (S1+) goes from Abel-Heim via several stations (not shown) to Euler-Hausen and to Dantzig-Stadt and back

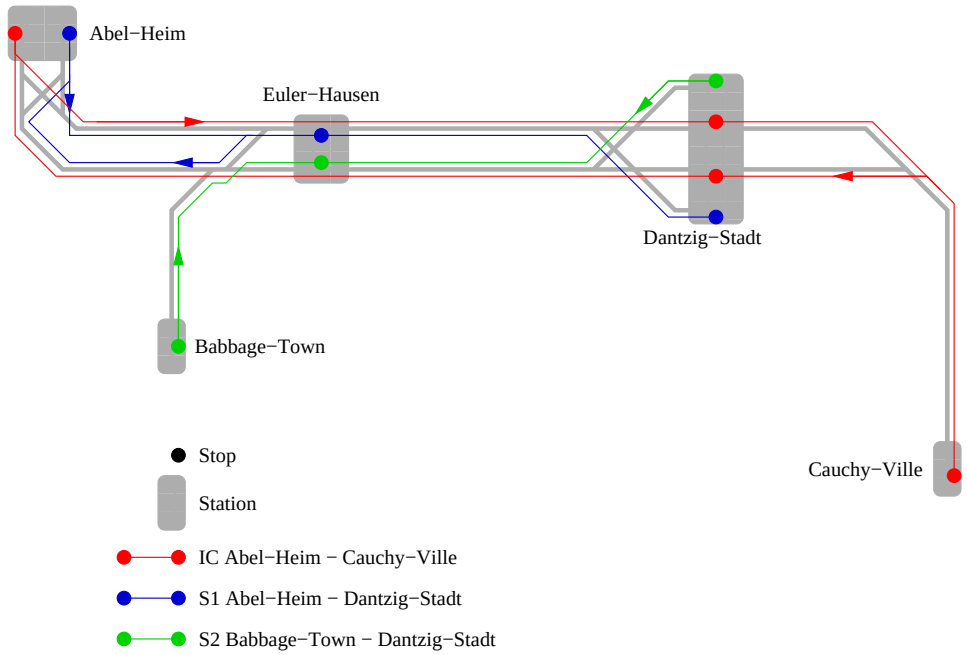

Figure 1.3: Layout and lines of MATHWORLD.

(S1−). The second commuter train route (S2+) starts in Babbage-Town, stops at Euler-Hausen, turns in Dantzig-Stadt and returns via Euler-Hausen to Babbage-Town (S2−). Moreover, for each track segment, travel times for the different rolling stock types are given; *e. g.* it takes 5 minutes for an intercity train to travel from Euler-Hausen to Dantzig-Stadt, whereas it takes 6 minutes for a commuter train. The train service intention includes the following connection possibilities in Dantzig-Stadt:

(i) from S1+ to IC+ with at least 4 minutes transfer time (change of platform)

(ii) from S2+ to IC− with at least 4 minutes transfer time (change of platform)

(iii) from IC− to S2− with at least 4 minutes transfer time (change of platform)

(iv) from S2+ to IC+ with at least 1 minute transfer time (same platform)

(v) from IC+ to S2− with at least 1 minute transfer time (same platform)

Is there a feasible timetable for these specifications meeting the resource restrictions of the network layout? A timetable is said to be feasible, if for each train there is a path through the network such that the timetable, interchange and safety rules are satisfied. Assume that the timetable shown in Table 1.1 has been proposed by an oracle using a simplified network topology only. Assume further that this timetable assures that at any time there is only one train per platform.

		IC+	IC−	S1+	S1−	S2+	S2−
Abel-Heim	arr	-	57	-	16	⋮	⋮
Abel-Heim	dep	0	-	30	-	⋮	⋮
Babbage-Town	arr	⋮	⋮	⋮	⋮	-	58
Babbage-Town	dep	⋮	⋮	⋮	⋮	3	-
Euler-Hausen	arr	*(20)*	*(37)*	14	30	19	40
Euler-Hausen	dep	*(20)*	*(37)*	16	32	21	42
Dantzig-Stadt	arr	25	28	22	-	27	-
Dantzig-Stadt	dep	29	32	-	24	-	34
Cauchy-Ville	arr	57	-	⋮	⋮	⋮	⋮
Cauchy-Ville	dep	-	0	⋮	⋮	⋮	⋮

Table 1.1: A draft timetable for MATHWORLD. The numbers in brackets are train passing times without stops.

Since the method does not use the exact topology, one cannot be sure that this timetable fulfills all requirements. Obviously, the proposed timetable serves the intended lines, and the change of trains in Dantzig-Stadt is possible for all five designated interchange possibilities. Yet safety restrictions are *not* satisfied! This can be shown as follows. S2+ arrives at Euler-Hausen at time 19, waits then for two minutes before it continues its journey to Dantzig-Stadt at time 21. IC+ passes Euler-Hausen at time 20, while S2+ waits in Euler-Hausen. Therefore two different tracks are needed in Euler-Hausen for these two train services. The travel time between Euler-Hausen and Dantzig-Stadt is five to six minutes for both trains. Furthermore, S1− starts its itinerary from Dantzig-Stadt to Euler-Hausen at time 24. Since between these two stations three trains share the same two parallel tracks at the same time (between time 24 and 27), a conflict arises—no matter how the exact routing is chosen.

A feasible train schedule serving the shown lines with a cycle of 60 minutes is shown in Table 1.2. Note the unconventional line management for S1− (see Figure 1.3). Using this approach, it is possible for IC+ and IC− to pass S1− and S2+ respectively at Euler-Hausen.

Assume that such a timetable generation method would provide two feasible but different timetables satisfying the same train service intention. *Is there a substantial difference between them? Which one is to be preferred?*

Initially, we could say that the timetable needing the smaller cycle is better; it uses the existing resources more efficiently. Now, assume that both timetables need about the same amount of time to be executed. Then, one could argue that the timetable providing more reserves to recover from delays is better, because it is easier to operate. It is important to note that the train scheduling problem is an off-line problem, *i. e.* there

		IC+	IC−	S1+	S1−	S2+	S2−
Abel-Heim	arr	-	57	-	**22**	⋮	⋮
Abel-Heim	dep	0	-	30	-	⋮	⋮
Babbage-Town	arr	⋮	⋮	⋮	⋮	-	58
Babbage-Town	dep	⋮	⋮	⋮	⋮	3	-
Euler-Hausen	arr	*(20)*	*(37)*	14	**36**	19	40
Euler-Hausen	dep	*(20)*	*(37)*	16	**38**	21	42
Dantzig-Stadt	arr	25	28	22	-	27	-
Dantzig-Stadt	dep	29	32	-	**30**	-	34
Cauchy-Ville	arr	57	-	⋮	⋮	⋮	⋮
Cauchy-Ville	dep	-	0	⋮	⋮	⋮	⋮

Table 1.2: A feasible schedule for the MATHWORLD example satisfying all restrictions.

is no uncertainty about any of the framework parameters. This implies that all trains are assumed to run exactly on time. However, a good timetable does not neglect the inherent variability of railway services.

In chapter two the complex interrelationship between «resource utilization» and «stability of the timetable» will be discussed.

The topic of interference is discussed in chapter three where it will be shown how conflicts can be detected. In this chapter the fundamental model and a basic algorithm to check the feasibility for a draft timetable are developed.

Chapters four and five then address the optimization problem, *i. e.* can a feasible timetable be stabilized? It is not perfectly obvious how to measure the stability of a timetable. Therefore, several measures that account for the different aspects on stability are introduced. Based on these measures, the optimization task is investigated and local search algorithms (heuristics) designed to find stable schedules are presented.

All steps above are executed during the planning phase of a new timetable. Yet, are these elaborate plans stable once they are placed in operation? In chapter six the point of view is changed and the schedule is studied on-line. The on-line problem is quite different, as interruptions occur during the execution of the schedule and eventually trains have to be rescheduled. The first question is whether for a given deviation from the timetable, the schedule has to be changed at all. Interruptions may be manyfold—they range from track failures to engine breakdowns to late departures. The result is often that one or more trains experience some delay, where the amount strongly depends on the type of the interruption. The operational reactions are manyfold too: Changing platforms, waiting for connecting trains up to rescheduling all trains within

a region. This study is restricted to the delays of trains and to showing whether a schedule needs adjustments or not.

1.3 Layout of the Case Study

This chapter concludes with the description of two realistic test cases for the validation of the presented models and algorithms. Both have the same track layout of the Bern station region, but with different train service intentions. The first one (Bern 2003) is drawn from a recent timetable (2003), and the other is a larger, imaginary situation (Bern 2020) that could be used in 15 to 20 years.

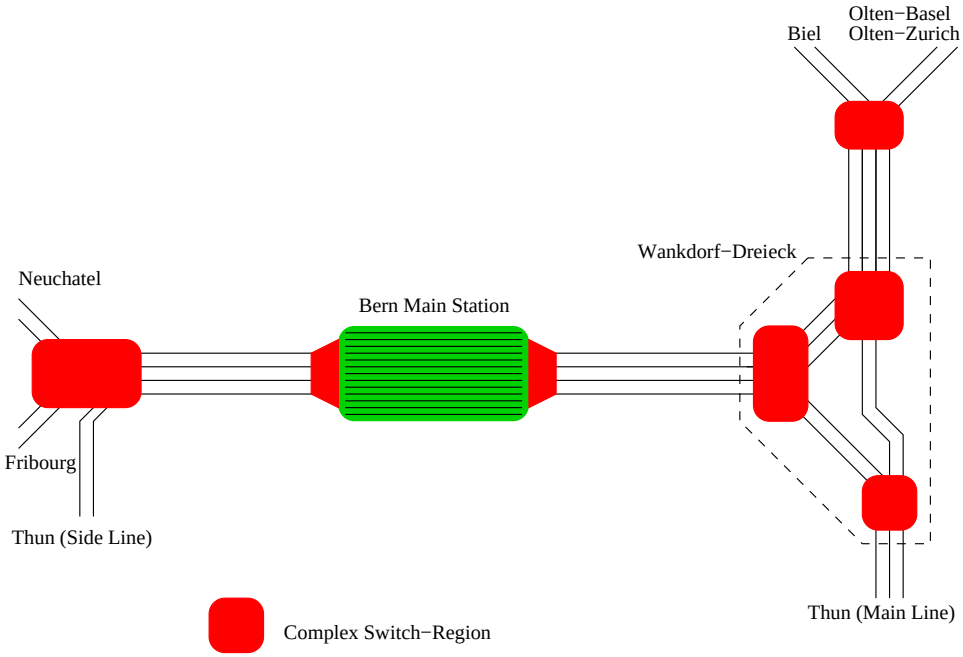

Figure 1.4: Track layout of the Bern station.

The Bern station is an important node in the Swiss railroad network. The Intercity line St. Gallen – Winterthur – Zurich Airport – Zurich – Bern – Fribourg – Lausanne – Geneva connects six of the seven largest cities in Switzerland. The segment Zurich – Bern is of great importance as it has the highest passenger volume and hence is one of the most profitable lines. Furthermore, one of the two main north-south routes (Basel – Milan) passes Bern too. This line is crucial for freight trains but also has a considerable passenger volume. Most of the freight trains do not have to enter the main station, as there is a tangential shortcut in the *Wankdorf-Dreieck*. Besides those two main lines, there are minor lines to Neuchatel, Biel and Thun (see Figure 1.4). Twelve platforms are available for trains to stop within the station. An additional thirteenth

track leads through the station and is mainly used for allowing platform twelve to be occupied with two short regional trains simultaneously. The most complex switching areas are located to the east and west of the main station where the four parallel inbound tracks broaden to the thirteen in-station tracks. The third critical area is the *Wankdorf-Dreieck*.

The train service intention for the scenario Bern 2003 has the lines shown in Table 1.3 (interchange possibilities are not shown). Despite the large number of already existing trains, the timetable for the year 2003 (as well as the recent 2005 timetable) still has some reserves for additional trains. The Swiss Federal Railways is planning to increase the frequency of passenger lines considerably, and a possible future train service intention is thus shown in Table 1.4. This extension emphasizes additional intercity and regional trains. However, this scenario—herein after referred to as Bern 2020 scenario—is only one of many possibilities to operate the existing track network of Bern at full capacity.

Local trains:	Thun – Bern – Thun	half-hourly
	Schwarzenburg – Langnau	half-hourly
	Belp – Biel	half-hourly
	Laupen – Thun	half-hourly
	Bümpliz Nord – Burgdorf	half-hourly
	Langnau – Schwarzenburg	hourly
	Thun – Fribourg	half-hourly
	Biel – Belp	half-hourly
	Burgdorf – Bümpliz Nord	half-hourly
	Schwarzenburg – Konolfingen	hourly
	Konolfingen – Schwarzenburg	hourly
	Fribourg – Bern – Fribourg	hourly
	Neuchatel – Bern – Neuchatel	hourly
	Murten – Bern – Murten	half-hourly
	Burgdorf – Bern – Burgdorf	hourly
Regional Express trains:	Geneva – Lucerne	hourly
	Lucerne – Geneva	hourly
	Zurich – Bern – Zurich	half-hourly
	Biel – Bern – Biel	half-hourly
	Neuchatel – Bern – Neuchatel	half-hourly
	Olten – Bern – Olten	half-hourly
Intercity trains:	Geneva – Zurich	half-hourly
	Zurich – Geneva	half-hourly
	Basel – Brig	hourly
	Brig – Basel	hourly
	Zurich – Interlaken	hourly
	Interlaken – Zurich	hourly

Table 1.3: Train service intention for the Bern 2003 scenarios.

Local trains:	Thun – Bern – Thun	X:00		X:30	
	Schwarzenburg – Langnau	X:00		X:30	
	Langnau – Schwarzenburg	X:00		X:30	
	Belp – Biel	X:00		X:30	
	Biel – Belp	X:00		X:30	
	Fribourg – Thun	X:00		X:30	
	Thun – Fribourg	X:00		X:30	
	Brünnen – Burgdorf	X:00		X:30	
	Burgdorf – Brünnen	X:00		X:30	
	Neuchatel – Bern – Neuchatel		X:15		X:45
	Payerne – Bern – Payerne			X:30	
Regional	Geneva – Lucerne	X:00			
Express trains:	Lucerne – Geneva	X:00			
	Biel – Bern – Biel	X:00		X:30	
	Emmental – Bern – Emmental	X:00			
	Schwarzenburg – Bern – Schwarzenburg		X:15		X:45
	Belp – Münchenbuchsee		X:15		X:45
	Münchenbuchsee – Belp		X:15		X:45
	Laupen – Münsingen		X:15		X:45
	Münsingen – Laupen		X:15		X:45
	Neuchatel – Bern – Neuchatel			X:30	
	Burgdorf – Bern – Burgdorf		X:15		X:45
	Olten – Bern – Olten			X:30	
Intercity trains:	Geneva – Zurich	X:00		X:30	
	Zurich – Geneva	X:00		X:30	
	Basel – Interlaken	X:00			
	Interlaken – Basel	X:00			
	Olten – Brig	X:00			
	Brig – Olten	X:00			
	Zurich – Bern – Zurich		X:15		X:45
	Basel – Bern – Basel			X:30	
	Aarau – Spiez			X:30	
	Spiez – Aarau			X:30	

Table 1.4: Train service intention for the Bern 2020 scenarios. Interconnections are defined by the affiliation of the trains to the four categories. Emphasized trains are additionally introduced compared to the 2003 timetable.

Note that in Table 1.4 the interchange possibilities are explicitly given by the affiliation of the trains to the four categories X:00, X:15, X:30, and X:45. Overall, some lines have changed, yet there are six additional lines within half an hour compared to the scenario Bern 2003. A draft schedule for this expanded train service was created by the SBB. However, it was concluded that this condensed timetable was too tight to execute. This future timetable could be made operational only by changing some constraints, such as the network or the train service intention (see results section of Chapter 5 and 6 for more information).

Chapter 2

Stability, Capacity, and Quality of Service

> The real voyage of discovery consists not in seeking new landscapes but in having new eyes.
> *Marcel Proust (1871–1922)*

Decisions on network layout and line plans are strategic long-term decisions, whereas the definite construction of schedules for engines, coaches and employees is a short-term task that has a horizon of some weeks—maybe a few months. Timetable construction lies somewhere in between; in Switzerland it is constructed for two year periods with small adjustments after the first year. Together with the strict resource restrictions, a timetable must also satisfy several soft restrictions. Soft restrictions are conditions that may be disregarded if a violation is unavoidable. Stability of a timetable is such a restriction: If possible, trains should not interfere each other in order to stabilize the schedule. However, trains never run exactly on time and therefore, a timetable only serves as a guide during operation, which should be fulfilled as precisely as possible. Therefore a timetable needs some reserves in order to cope with deviations.

2.1 Two Level Approach

In a joint project with Burkolter, the timetable saturation problem is addressed for station regions (see his thesis on *Capacity of Railways in Station Areas using Petri Nets* [Burkolter, 2005]) aiming at a maximum utilization of the existing track network

while maintaining today's level of stability. As input an exact track topology which is aggregated subsequently, and a train service intention including train acceleration characteristics is used. In our framework, timetables are designed in two steps: First, a draft timetable for the aggregated topology is constructed, omitting possible conflict regions such as switch areas but focussing on capacity restrictions in parallel track areas. In this timetable, passing times at important locations of the aggregated topology—such as arrival and departure times at the platforms or switching regions—are fixed. However, the draft timetable may not be feasible, *i. e.* it is not guaranteed that every train run is possible without interfering with other train runs within the omitted regions. Nevertheless, outside the possible conflict regions, the timetable respects safety restrictions and is dense in the sense that the train sequence is optimized in order to achieve a high utilization of the track network.

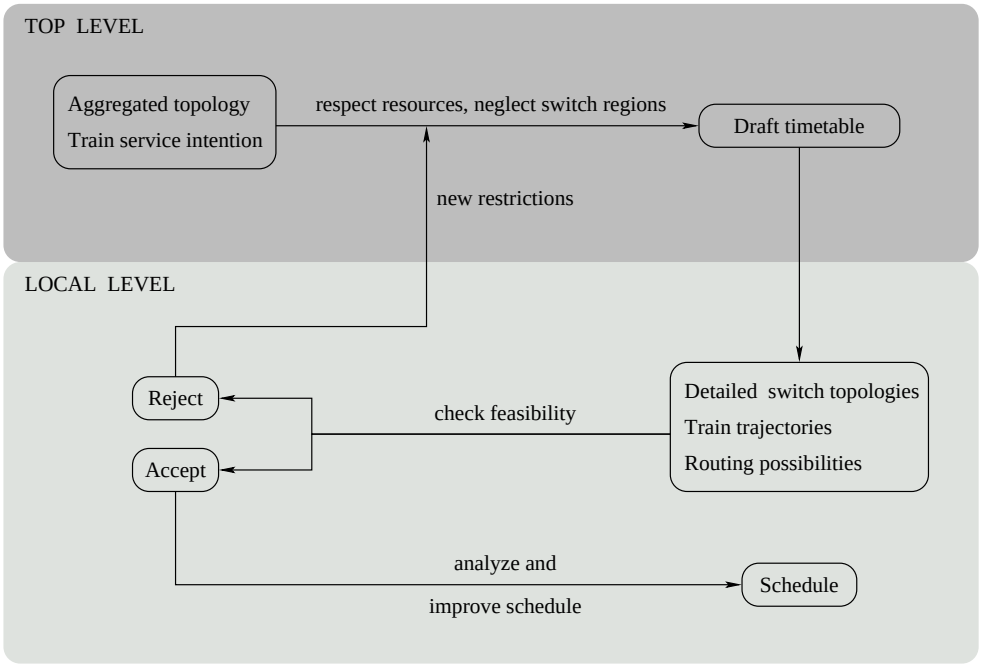

Figure 2.1: The two level approach for schedule construction.

In a second step, the verification phase, the timetable is checked for feasibility within the possible conflict areas. Since the lines are fixed, origin and destination of the train runs are known, but not the exact routes through the network. In the thesis at hand, the verification algorithms generate and check routes in detailed topologies for all intended itineraries, such that all safety restrictions are met—if possible. If such routes are successfully found, the draft timetable is feasible and is further investigated in a third step. In case of a failure, the draft timetable is not operational and has to be adapted. The idea is that the draft timetable can be made feasible by small changes

without discarding the timetable and restart the process by calculating a new draft timetable. The details on the handling of non-feasible timetables are described in Burkolter's thesis [Burkolter, 2005]. Figure 2.1 shows a schematic overview of this two level approach to the timetabling problem.

Within this structure, it is possible to separate the problem of respecting the capacity restrictions from the feasibility problem, which deals with exact routings in detailed topologies. Moreover, this split allows strict and soft restrictions to be considered separately. In the first step a draft timetable is constructed such that the available capacity is optimally used—given a train service intention. The second phase analyzes the feasibility and stability of such a draft timetable. Here, the times at certain locations (*e. g.* initial journey points) are fixed and an exact routing has to be determined. Once a timetable has been proven to be feasible, it is examined for certain aspects characterizing stability issues.

If the intended train service becomes denser (*e. g.* four trains within an hour instead of one or two trains) the timetabling problem becomes increasingly difficult to solve. Feasibility has seldomly been an issue so far in practical instances, but with increased train service intentions, the servicing of the intention becomes a challenging task. The ultimate goal for future schedules is the construction of a timetable which highly utilizes the available capacity and is at least as stable as the situation today. Moreover, a good timetable is not only feasible but satisfies given stability requirements in order to cope with small timetable deviations.

2.2 Stability of Timetables

The timetabling problem has both off-line and on-line aspects. «Off-line» means that the schedule is constructed so that some requirements such as the capacity and stability requirements are satisfied. In the on-line problem, dispatchers are confronted with certain timetable deviations. They have to maintain the schedule, *i. e.* given a certain deviation in the plan, their task is to restore the plan. Furthermore, solutions must be found quickly, since time is bounded and thus problems have to be solved in real-time or near real-time. Therefore, the planned schedule should not be too complex to manage; or in other words, the schedule should permit certain local disturbances that do not necessitate a global re-scheduling. In the thesis at hand, the focus is mainly on the off-line problem, however some on-line aspects are included. To develop algorithms suiting the real-time problems is not the main concern.

While capacity and safety requirements are fixed restrictions, which must be satisfied, the construction of timetables also comprises compliance with soft constraints, which are usually very complicated and difficult to formalize. A draft timetable provides passing times of the trains at the main points of a train service, but is inadequate when seeking to quantify the stability. Whereas capacity estimates are unus-

able without the certificate of a (feasible) timetable for a given train service intention (see [Burkolter, 2005]), statements on the stability are worthless unless the exact routes of all trains are known. Thus, the draft timetable has to be provided with routings, *i. e.* paths for each train through the track network.

Obviously, the stability of a timetable depends heavily on the degree of utilization of the network. In a fictitious timetable, in which all trains are scheduled to run at maximum speed and with smallest allowed distance, it is almost impossible for the trains to recover from a delay. The crucial element in the quantification of stability is the *interaction* among trains. *Therefore, any measurement of the stability of a timetable explicitly or implicitly measures the «level of interaction» among trains.*

In Chapter 4 different aspects of stability are formally defined. Regardless of the explicit measure, a timetable is called *stable* as long as the perturbations do not force changes to the schedule. Usually deviations are not isolated and do not occur for one train only, but are highly correlated. In essence, a *maximum perturbation tolerance* can be defined in order to quantify stability. It indicates the limits of the self-regulation of a schedule and accordingly, operators have to intervene actively in the running system more or less often.

Furthermore, for a customer it is important that he/she can rely on the timetable and arrive on time at his/her destination. The customer's main concern is the reliability of the railway system—reliability of planning and operation; and both have to be harmonized. It does not make sense that the timetable planners design a new timetable that is very sensitive to outside influences or difficult to implement and maintain. Therefore, it is important to investigate the following question: *Can the quality of service be increased without changing the underlying service intentions?*

How is the quality of service measured? Is it the travel time from A to B that has to be minimized? Perhaps not; we all know that due to environmental influences, trains may be delayed. A customer then prefers a longer travel time with on-time arrivals to shorter travel times with often delayed arrivals. This is even more relevant if the customer has to change trains at intermediate stations (see [Goverde, 1998b]).

The most common method to increase timetable stability is to add buffer times to the timetable. Usually different types of reserves are included in the timetable, *e. g.* a travel time reserve or a headway reserve. The travel time reserve is a buffer time that is added individually for each train to its technical travel time. With this buffer the trains are able to reduce delays by driving faster.

Another possible means of including travel time reserves is to use a waiting time buffer in stations. The advantage is obvious: An incoming delayed train may catch up some of its delay, if the transfer of passengers is faster than expected. With the help of this buffer time a train—although late at arrival—may leave the station on time.

A third type of buffer is the headway reserve, *i. e.* the trains do not follow each other with the minimum allowed headway, but some additional distance is added between

two trains (see [Kaminsky, 2001]). The goal of this buffer is that small perturbations in the schedule do not propagate, but are absorbed by the headway buffers (prevention of the «domino effect»).

The German Railways studied the influence of additional travel and waiting times on punctuality (see for example [Kannengiesser and Wiche, 1987]) and implemented guidelines for train dispatchers. These guidelines contain lists that specify waiting rules for each train service; *e. g.* Intercity *A* waits no longer than three minutes for Intercity *B* and no longer than two minutes for the regional trains *C* and *D*. Additionally it contains a long list of exceptions for particular connections, especially for international trains. Similar guidelines are followed in the Netherlands and in Switzerland.

All these lists and guidelines on the so-called synchronization control margins are often based on rules of thumb and experience. The effect of synchronization control times on passenger waiting times has been rarely discussed in theory, yet understanding those effects is important to build efficient timetables. Goverde proposed to optimize the synchronization control time in order to minimize passenger unpleasantness (see [Goverde, 1998b]).

The two level approach allows another point of view in order to characterize timetables: The assignment of trains to routes is crucial in order to reduce the interaction among the trains, and having variety in choosing routes can increase both the stability and flexibility of a timetable. If the variation of possible routings is small, a timetable could become infeasible with the slightest deviation. However, having many alternatives results in problems of finding confirmations of non-feasibility of a draft timetable due to the large solution space. Moreover, having many switches (and thus many routing alternatives) may cause the timetable to become instable, as the dependencies among the routes of different trains are increased. Hence a late train could interfere with many other trains and propagation of initial delays could quickly start the «domino effect».

2.3 Node System

Additional waiting times in stations require capacity and inhibit the extension of the train service intention. Therefore waiting time buffers should be avoided—if possible. Allowing trains to travel at their technical and approved speed limit within station areas, regains spare capacity that can be used to introduce additional trains. This is the main idea of the *node system* which will be outlined now.

In order to cope with the increasing demand of track utilization the project «Bahn 2000» was started in the 1980s in Switzerland. The goal was not to make single investments in new tracks or new equipment, but to find a balance between all the different investment possibilities in order to provide a sound rail service. It was concluded that it is not so important that the trains travel as fast as possible, but that

all train transport is well coordinated. Today the task is to organize and manage the timetable in order to offer a comprehensive rail service. Crucial attributes of good timetables are short transfer times and an attractive line map for passengers. A good timetable should also be able to deal with delays and lastly—very important for the rail company itself—it should be easy to implement.

One of the main passenger requirements is the accurate connection at transfer stations. Additional waiting time should be short and many connections should be provided. Within the «Bahn 2000» concept this task is tackled by the node system: Among all the train stations in Switzerland there are some that act as designated transfer stations. In these stations called «node stations», all the trains from different directions arrive on the hour (or shortly before) and leave the station again some minutes after the clock hour to their destinations. This idea is not bound to the clock hour and has been extended to the half and quarter hour too (see Figure 2.2).

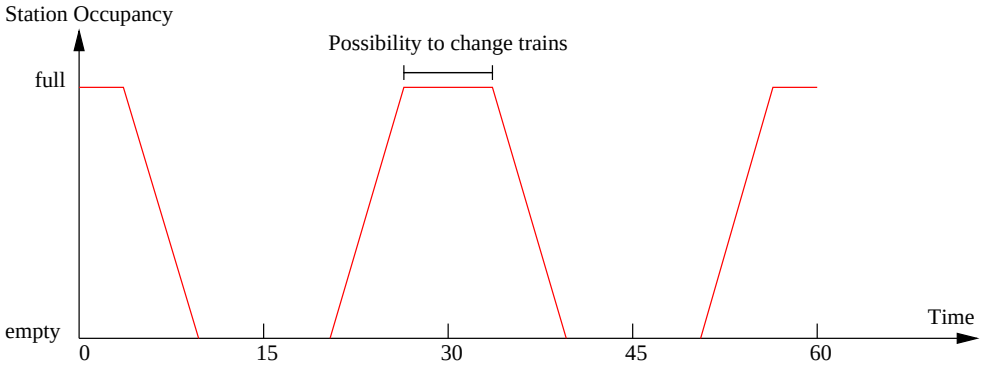

Figure 2.2: Station occupancy and the node system. On the hour and the half hour the station is filled with trains from every direction and afterwards it is emptied again. Of course the station is not empty at the time 15 or 45; at these times the station is used for commuter trains within the greater conurbation area for example.

An important requirement in order for this system to work is accurate train travel times between the node stations. The driving time should be less but close to 30 or 60 minutes. Figure 1.1 shows that the travel time between Bern and Zurich decreased from 69 to 58 minutes in 2004. This was possible because new tracks were built, where trains could travel faster. Besides the travel time reduction of 15%, this extension enabled the node system to be implemented in two main stations of Switzerland (see [SBB, 2004]). The CEO of the Swiss Federal Railways summarized the system in the following statement: *The trains in Switzerland do not travel as fast as possible, but as fast as necessary!* (source: [SBB, 2004])

The condensation of timetables in station areas obviously causes operational problems. The traffic becomes heavier and small disturbances of arrival or departure times

may lead to serious train managing problems. Therefore, an exact and efficient processing of train operations is of utmost importance in node stations. However, the concept of canceling time reserves from the station areas is a big change with respect to the system of today (see [Kaminsky, 2001]).

In this new concept there is a clear separation between highly utilized areas having no buffer times through which the trains should travel as quickly as possible, and lesser utilized areas in which trains are able to recover from delays due to the presence of buffer times. Burkolter shows that even condensed timetables in station areas *inherently* provide some small buffer times (see [Burkolter, 2005]). However, increasing the utilization necessitates more elaborate management and planning of available buffer times.

Trains cannot wait indefinitely for inbound connecting trains and connections will be canceled earlier under this new concept. The denser train service will compensate for this inconvenience. However, it is important that connecting trains wait, otherwise the travel time for certain passengers can be increased enormously, as they have to wait for the next train that may only leave one hour later (see also [Goverde, 1998a] or [Heusch et al., 1997]). Yet, waiting for a too long time causes other trains to wait at other stations, and soon there is not only one or two delayed trains but substantial parts of the train system are late («domino effect»).

The aim is to provide a system with the same stability as today and hence either the buffers have to be managed more efficiently and/or additional buffers have to be introduced elsewhere. Since reserve times are canceled from the station areas, the only possibility for including other buffers is in zones between stations.

2.4 Measurement of Capacity and Efficiency of Timetables

From the customer's point of view there are several aspects that are included in the notion «Quality of Service» (see [IRSE, 2004]). Railroad traffic will in any case increase, but stations cannot be extended in the short term, and the quality of service should not suffer. Therefore, it is important to discuss the relationship between track utilization and time reserves.

While generous buffer times in the timetable prevent delay propagation, they also prevent a high utilization of the track network. Recall that in the two level approach the headway buffer times are neglected while constructing a draft timetable. Since the stability of a timetable depends on the interaction between the train services; schedules for regions with many tracks and switches are crucial in order to achieve a high utilization of the track network. Usually these regions are found near (main) stations, whereas on the connections between the stations, the so-called link or connection zones, only few switches exist. Moreover, near stations trains usually travel slowly

whereas in the connection zones they run fast.

In order to fulfill future train service intentions, the utilization of stations must be increased, and hence the buffer times must be decreased. The introduction of ETCS (see Appendix A for a short overview) allows the timetable to be condensed without investment in new tracks, since the time gap between successive trains can be reduced. Another important instrument to enable high utilization of the network and simultaneously a simple train station management is the introduction of the node system. Yet, *what is the utilization of a railway system? What is the capacity of a railway network?*

The capacity of a network could be defined as the maximal throughput of trains within a certain amount of time given a network topology. Burkolter shows that this definition is not suitable as it neglects some substantial aspects of capacity measures (see [Burkolter, 2005]):

(i) Changing the train types that serve a certain line may increase or decrease the network's utilization considerably.

(ii) Altering the line plan, *e. g.* by changing designated platforms, may cause interference between trains and thus decrease the utilization since trains may have to cross each other instead of traveling side by side.

(iii) Introducing or canceling connections at main stations may introduce more or fewer dependencies and thus the throughput might be decreased or increased respectively.

(iv) Adding or removing switches within the topology may cause routing problems, since the dependencies between the trains have changed.

(v) Increasing or decreasing the minimum allowed distance between two trains influences the utilization of the network as well.

All the above mentioned points directly affect the available capacity. Therefore, a definition of capacity not only includes the network and train speed, but also the line plan, connection alternatives, safety rules and train types. In his thesis, Burkolter points out that train service intentions play an important role when defining capacity (see [Burkolter, 2005]):

Definition 2.1 (Capacity of Track Network) *The capacity of a track network is the minimal time needed to fulfill a given train service intention.*

Instead of maximizing the number or the throughput of trains within a fixed amount of time, the time needed to schedule all the trains in the train service intention is minimized. Hence together with a statement on the capacity, a confirmation (*i. e.* a

timetable) always has to be provided to validate the capacity statement. This definition does not provide an absolute definition of capacity, *i. e.* it is not independent of a particular train service intention.

Suppose that a feasible schedule is given. Due to the timetable, the period is thus known. Furthermore, suppose this period of the timetable is thought to be the shortest possible for the given train service intention. Although hardly operational, this timetable defines the capacity of the network. Any other timetable with equal or longer periodicity uses the available track resources less efficiently but probably has a higher stability. An efficient schedule is then defined as follows (see Figure 2.3):

Definition 2.2 (Efficient Schedule) *A schedule is called efficient, if the timetable period has to be increased in order to increase the maximum perturbation tolerance and vice versa, a decrease of the period causes a reduction of the maximum perturbation tolerance. The optimal schedule can be characterized by the schedule that achieves the maximum perturbation tolerance among all schedules that meet a certain given target periodicity.*

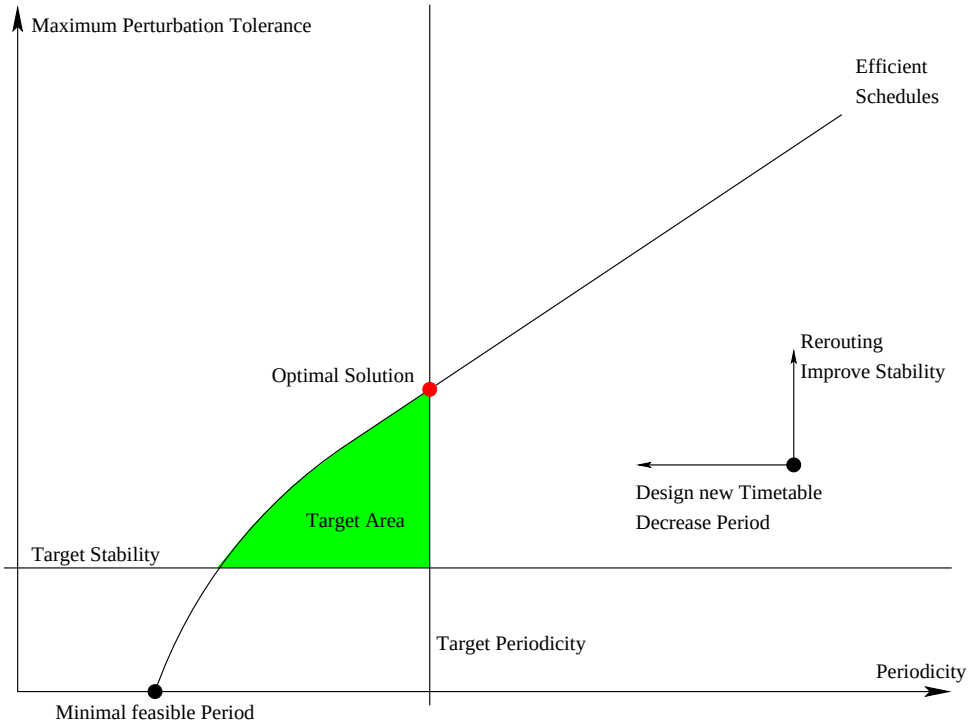

Figure 2.3: Relationship between capacity and stability.

Chapter 3

The Train Routing Problem

> Whenever a theory appears to you as the only possible one, take this as a sign that you have neither understood the theory nor the problem which it was intended to solve. *Karl Popper (1902–1994)*

The previous chapter showed that it is difficult to answer the question of whether or not a certain timetable is a «good» timetable. However, *efficient* schedules have been defined as those schedules for which an increase in the utilization of the track network causes a decrease in the stability of the timetable and vice versa. Moreover, schedules have been characterized by their periodicity, their tightness, their ability to recover from perturbations, etc. In this chapter, the fundamental problem of the feasibility of a given draft timetable is addressed, since this is at the core of building timetables. Henceforth assume that the following items are given:

Network topology. In the network topology, switches, stretches and platforms are specified. Other track characteristics (*e. g.* slopes or signals) are disregarded here. In Section 3.1 *double vertex graphs* are introduced to precisely describe track network topologies.

Train service intention. The train service intention defines lines that are provided and additionally the types of trains which serve the designated lines (see Definition 3.4 for a precise description).

Draft timetable. As described in the two level model (see Figure 2.1) switch regions are critical areas. Burkolter describes methods to find an aggregated network topology and condensed draft timetables given a train service intention

(see [Burkolter, 2005]). In Section 3.2 it is shown how a draft timetable is used to build track reservations using different train safety systems.

In order to verify whether a draft timetable is feasible or not, detail local topologies (such as switch regions) are used in the two level model (see Chapter 2). Yet, the given network topology is not restricted to such a critical region; entire station areas can be used as given network topologies. Although the presented models and algorithms are not bound to station or switch area topologies, they have been developed in order to cope with problems that particularly arise in station regions. The first problem—the *timetable feasibility problem*—whether an intended schedule is feasible or not can be stated as follows.

In order to decide a timetable to be feasible or infeasible, a network topology, a train service intention and a corresponding timetable have to be given. These three items together form the instance to the timetable feasibility problem:

Definition 3.1 (Instance to the Timetable Feasibility Problem) *The triple (network topology, train service intention and draft timetable) is called an instance to the timetable feasibility problem.*

In order to build and analyze an operational schedule, routes have to be assigned to each train in the train service intention. A draft timetable is said to be feasible, if there is an assignment of routes to all trains, which satisfies the draft timetable as well as the safety rules (trains are not allowed to travel too close to each other). If such an assignment does not exist, then the draft timetable is said to be infeasible, as it cannot be operationalized. In case of infeasibility at least one of the three inputs—timetable, topology or train service intention—has to be changed in order to generate a feasible schedule. Hence the timetable feasibility problem is:

Problem 3.2 (Timetable Feasibility Problem) *Given an instance to the timetable feasibility problem (network topology, train service intention and draft timetable); find for each train in the network a route, such that all safety restrictions are met and such that the draft timetable is adhered to.*

The first section of this chapter describes the model of the network topology and a search algorithm for finding all possible paths between two designated points of the topology. The obtained lists of paths form the decision space, since the feasibility check consists of assigning trains to compatible routes, which are described by a path and a draft timetable. The content of the second section is the modeling of the safety restrictions. As not every combination of routes is allowed, the infeasible configurations have to be detected; this is the content of the third section. The fourth and fifth sections deal with models and algorithms to solve the train routing problem. Some preliminary results about speed and reliability of the algorithm applied will conclude the chapter.

3.1 Paths Through Track Networks

A railway track network is very complex, especially in station areas. Henceforth, only the topological properties of the network are required to solve Problem 3.2. Montigel proposed to describe a track topology with *double vertex graphs* (see [Montigel, 1992] and [Montigel, 1994]). A railway network (*i. e.* the track layout) can be represented by a conventional graph with the exception that each vertex is doubled; *i. e.* each vertex has a unique «partner», formally:

Definition 3.3 (Double Vertex Graph) *Let V be a finite set of vertices, $E \subseteq V \times V$ a finite set of edges between the vertices, i. e. $E = \{(u,v) \mid u,v \in V\}$. Moreover, let $\circ : V \to V$ be a mapping, the so called* joining mapping, *which satisfies $\circ(v) \neq v$ and $\circ(\circ(v)) = v$ for all vertices $v \in V$. Then a triple $D = (V,E,\circ)$ is called a* double vertex graph. *Write v° for $\circ(v)$.*

The motivation for the introduction of double vertex graphs is the notion of a path in the network (see Figure 3.1). If a switch is represented by only four nodes, it is possible to enter the switch through one leg and leave it via the other leg. Such paths are impossible in reality and cannot be handled by conventional graphs and notions of paths. While the choice of the next path edge is independent of the elapsed path in such graphs, there is a simple rule in the double vertex graphs preventing the exploration of impossible paths. This rule is: *Never use more than one outgoing edge of a single vertex—always use both partner vertices to describe a path.* Hence a path in a double vertex graph always follows *node-node-edge-node-node-edge*.... Formally a path p is described as follows:

$$p := \{v_1, v_1^\circ, (v_1^\circ, v_2), v_2, v_2^\circ, (v_2^\circ, v_3), \ldots\} \tag{3.1}$$

The representation of a railway network using a double vertex graph is crucial and thus it is assumed that the network topology is always represented as a double vertex graph D. In order to precisely describe the train service intention, some vertices in the double vertex graph receive a special tag. The set of vertices corresponding to platforms is called *platform vertices*—denoted by V^s; and *portal vertices* V^p representing the border nodes of the network. For example in the Bern test case, there are twelve platforms in the main station, four of which can be used simultaneously by two short trains. Moreover, there are six portal nodes representing the six major directions leaving Bern (see Section 1.3 for the details about this test case).

In order to use the track network a set of lines has to be defined. Lines are described by a set of portal and platform vertices, outlining the route of a train. Moreover, the rolling stock serving a line has to be specified. This information is summarized in the train service intention:

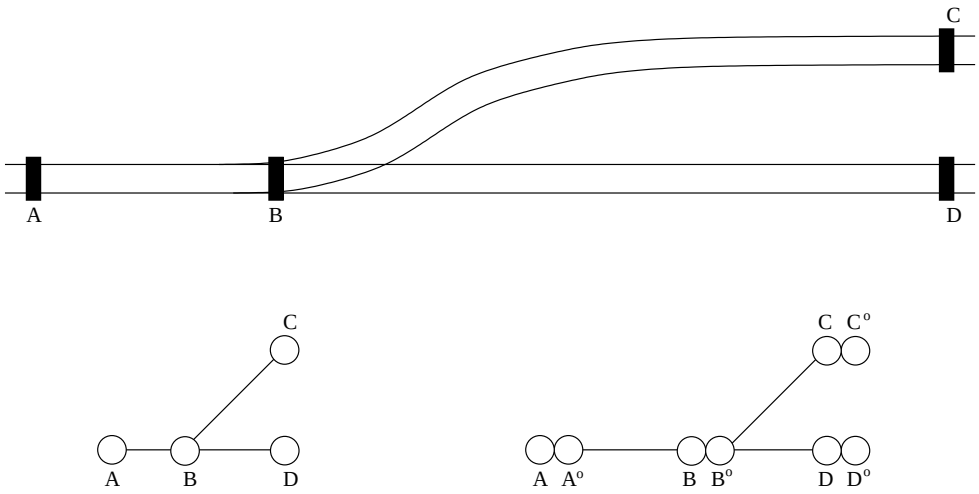

Figure 3.1: Modeling train itineraries in networks. In a railway network the itinerary $C \rightarrow B \rightarrow D$ is impossible (top), but would be a legal path in a conventional graph (lower left). However, with the double vertex graph and its rule always to pass both partner vertices, $C \rightarrow B \rightarrow D$ becomes illegal (lower right).

Definition 3.4 (Train Service Intention) *A given train service intention consists of a set of lines. A single line l_i is described by the following five-tuple:*

- *the type of the train w_i including the rolling stock*
- *eventually some pass-through points at specific locations within the station area (e. g. stops at minor stations located in the station area of the main station)*
- *an incoming portal node $v_i^{p_a} \in V^p$*
- *a platform node $v_i^s \in V^s$*
- *an outgoing portal node $v_i^{p_d} \in V^p$*

The first item characterizes the train serving line l_i: w_i contains all information—like the behavior while accelerating or braking, length, weight—which is needed for further calculations. This information is assumed to be given. The second item stores information whether a train should stop at additional stations within the station area. The last three nodes roughly determine the route of the train through the station area. The set of the triples $\{(v_i^{p_a} \in V^p, v_i^s \in V^s, v_i^{p_d} \in V^p)\}$ is called *line plan*. Note that here train interconnections are neglected since it is assumed that the provided timetable fulfills the train interconnection conditions.

However, exact routes connecting $v_i^{p_a}$, v_i^s and $v_i^{p_d}$ do not exist yet. With the help of the double vertex graph D, the exploration of the network topology to find all possible

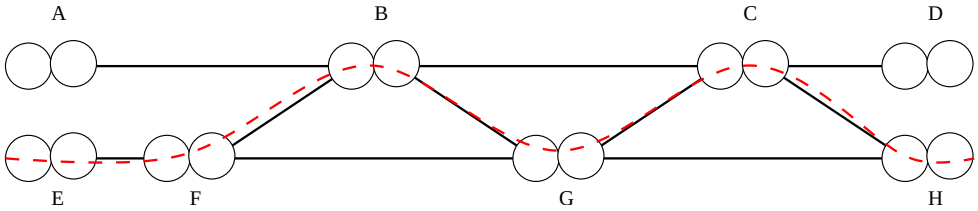

Figure 3.2: Dominated paths in track networks. Paths looking like sinuous lines (*e. g.* $E \to F \to B \to G \to C \to H$) are often dominated and can be canceled from the list of possible itineraries.

paths for the intended trips $v_i^{p_a} \to v_i^s \to v_i^{p_d}$ is made possible. Algorithm 3.1 is a variant of Montigel's FINDALLPATHS(v_i) algorithm (see [Montigel, 1992]). The algorithm is extended by a check that discards routes not fulfilling certain operational aspects.

Not all paths originally found by Montigel's FINDALLPATHS(v_i) algorithm correspond to routes a train may take. There are paths that cannot be executed or that are dominated by other, straighter paths. For example in Figure 3.2 the path $E \to F \to B \to G \to H$ is dominated by the straight path $E \to F \to G \to H$ under the assumption that the length of the tracks (length of the edges) is not too long. On the other hand the path $E \to F \to B \to C \to H$ may not be a poor itinerary, since a train on that path can pass a slower one that travels $F \to G \to H$—assuming that the edges are long enough. Hence, the underlying track network is important in order to specify a list of unwanted path fragments. Henceforth, a set S of path fragments considered to be «illegal» is assumed to be given additionally. With this set S of illegal path fragments, the list of all possible paths is perused and those having an illegal part are deleted in Algorithm 3.1.

Remark 3.5 (Time Complexity) *Obviously the time complexity of the path finding algorithms for double vertex graphs with n pairs of vertices is $O(2^n)$ as the simple example in Figure 3.3 shows. However, such configurations seldom exist in practice and the algorithms usually run fast.*

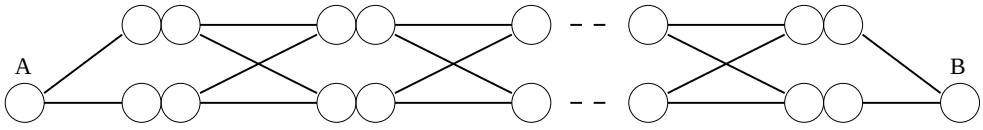

Figure 3.3: Finding all paths between two vertices A and B may cost exponential time as this example of n double switches with $2n+4$ pairs of vertices shows. Every inserted switch causes a doubling of the number of paths from A to B.

Algorithm 3.1 Find All Paths (u, v, P, F, S)

Input: A double vertex graph $D = (V, E)$ and two nodes $u, v \in V$ and a list of path fragments $S = \{\{v_{11}, v_{11}^\circ, (v_{11}^\circ, v_{12}), \ldots, v_{1k_1}\}, \ldots, \{v_{r1}, v_{r1}^\circ, (v_{r1}^\circ, v_{r2}), \ldots, v_{rk_r}\}\}$ that describe parts of unwanted routes; moreover a list P that stores all paths and a list F of already visited vertices (both initially empty).
Output: The list P of m_i paths contains all accepted paths connecting u and v.

$p := \emptyset$ // single path
set flag of *all* vertices v to UNDETERMINEDVERTEX

if $u = v$ **then**
    add v to p and p to P
    set flag of u to GOODVERTEX
    return
endif

u° := partner vertex of u
for all outgoing edges e of u° **do**
    w := **target**(e)
    if flag of w is not BADVERTEX **then**
        if $w \notin F$ **then**
            insert u in F
            FINDALLPATHS (w, v, P, F, S)
        endif
    endif
endfor
if $P = \emptyset$ **then**
    set flag of u to BADVERTEX
else
    set flag of u to GOODVERTEX
endif
for all paths $p \in P$ **do**
    insert u at the beginning of p
    if p contains a path fragment that is in S **then**
        delete p from P
    endif
endfor

Remark 3.6 (Itinerary Splitting) *Since the number of paths can be doubled by each switch, the line plan of the train service intention is changed for complexity reasons as follows. The planned itinerary $v_i^{p_a} \to v_i^s \to v_i^{p_d}$ of line l_i is divided into two itineraries $v_i^{p_a} \to v_i^s$ and $v_i^s \to v_i^{p_d}$. Henceforth, we assume to have n independent itineraries in the train service intention, each having one starting vertex v_i^a and one ending vertex v_i^d, whereas $v_i^a \in V^p$ and $v_i^d \in V^s$ or vice versa. A train with a portal starting point (and hence a station ending point) is called an incoming or inbound train and a train with a station starting and portal ending point is an outgoing or departing train.*

Remark 3.7 (Simplified Path Description) *In double vertex graphs a path is described by two vertices followed by an edge and followed again by two vertices, e. g. $u, u^\circ, (u^\circ, v), v, v^\circ, \ldots$. However, the additional rule using partner vertices is only necessary to enumerate all possible paths correctly. Always indicating both partner vertices is cumbersome and therefore, for the sake of simplicity, paths will be described by single vertices and edges only, i. e. by $u \to e \to v \to \ldots$. If not ambiguous, even edges might be left out, i. e. $u \to v \to w \to \ldots$*

3.2 Train Routes and Track Reservation

With Algorithm 3.1 all possible paths through the network can be enumerated for each train service, *i. e.* for all (v_i^a, v_i^d)-pairs, individually. By splitting the paths, the train itineraries are separated into incoming trains and outgoing trains (see Remark 3.6). Assume that n train itineraries and a corresponding set P_i of possible paths for each itinerary i is given. P_i contains m_i paths $p_{i1}, \ldots, p_{im_i}$ for $i = 1, \ldots, n$. A single possible path p_{ij} is described as $p_{ij} := v_1^{ij} \to v_2^{ij} \to \ldots \to v_*^{ij}$. Note that the length of the path—*i. e.* the number of nodes—depends on the path and therefore v_*^{ij} is used to represent the last vertex of the path p_{ij}. Moreover, assume that a draft timetable is given, which is characterized as follows:

Definition 3.8 (Draft Timetable) *A draft timetable for a path p_{ij} provides the exact position of train i on its path j for any given time τ. In particular, passing times in each node $v_1^{ij}, \ldots, v_*^{ij}$ are provided and denoted by $\tau(v_1^{ij}), \ldots, \tau(v_*^{ij})$.*

Timetables usually are assumed to have a periodicity T, implying that each event repeats after T time units. Hence, if a train service passes a vertex v at time $\tau(v)$ then the «same service» passes the vertex v also k time periods later at time $\tau(v) + kT$, for any $k \in \mathbb{Z}$. The times τ and $\tau + kT$ can thus be identified.

Analogously to time points, time intervals are bounded on domain $[0, T)$ as well.

Time intervals modulo T are written as $[a,b]_T$ and defined as follows:

$$\begin{aligned}
&\text{if } b-a \geq T && [a,b]_T = [0,T)\\
&\text{else}\\
&\quad \text{if } a \bmod T \leq b \bmod T && [a,b]_T = [a \bmod T, b \bmod T]\\
&\quad \text{if } b \bmod T < a \bmod T && [a,b]_T = [a \bmod T, T) \cup [0, b \bmod T]
\end{aligned}$$

Ideally a timetable not only provides passing times at vertices, but also a trajectory function $f_{ij} : [a,b] \longrightarrow \mathbb{R}$ that describes the exact position of a train i on its path j given the time $\tau \in [a,b] \subset \mathbb{R}$. The trajectory function $f_{ij}(\tau)$ then maps the time τ to the value $f_{ij}(\tau)$ representing the covered distance in the network area until time τ. By taking the inverse of the trajectory function f_{ij}, the passing time at position u in the network can be determined. u does not have to correspond to a vertex in the graph; it could also «lay» on the edge between two vertices. As each edge has a length, the position on the edge corresponding to u can be exactly determined. Henceforth, the passing time of an itinerary i at any position u on its path j in the network will be denoted by $\tau_{ij}(u)$ or—if not ambiguous—the path index j is omitted and train i passes u at time $\tau^i(u)$.

However, taking the periodicity T into account, several requirements have to be met in order to avoid ambiguity of $f_{ij}(\tau)$ and $\tau_{ij}(u)$:

(i) Two trains serving the same line are not allowed to travel on the same edge at the same time.

(ii) A train is not allowed to cross the same edge more than once.

Neither of the two assumptions is restrictive in reality: By splitting the itineraries into incoming and outgoing train itineraries (see Remark 3.6), a single itinerary uses edge e at most once, if the paths are circle-free, which can be assumed. The first requirement implies that the travel time of a train i over an edge e is smaller than the period time T. Since edge lengths are usually short in station regions (some hundred meters, maybe a few kilometers long) and the period time is usually 30 or 60 minutes, train itineraries do not cover an edge for a full period time. Thus, it can be assumed that the domain of $f_{ij}(\tau)$ for every edge e train i travels on is $[a,b]_T \subsetneqq [0,T)$ (see Figure 3.4).

Henceforth, a trajectory function is assumed to be monotonically increasing on $[a,b]_T$ and furthermore it is assumed that trajectory functions $f_{ij}(\tau)$ are given for every train itinerary i and every path p_{ij}. If only passing times at specific positions are available, then it is assumed that the trajectory functions are constructed by linear interpolation between any two consecutive values. Note that the trajectory function $f_{ij}(\tau)$ depends on the path p_{ij} and the train type w_i. By changing the train type or

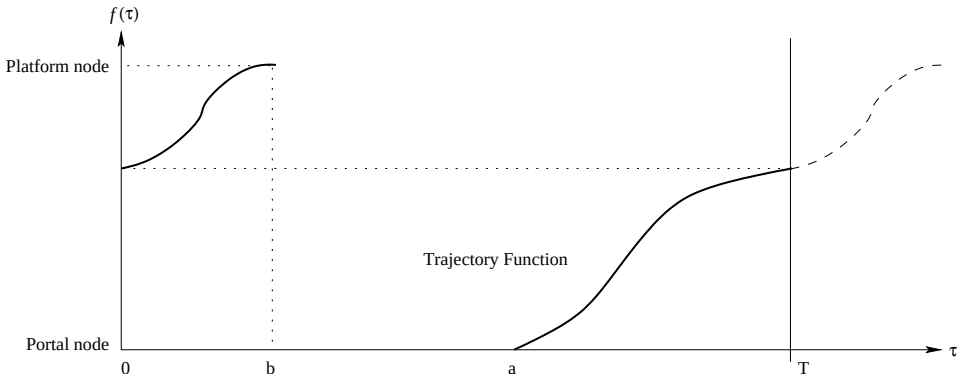

Figure 3.4: Trajectory function f with domain $[a,b]$ modulo T for an inbound train.

the path, the passing times at the vertices might also change. Paths and trajectory functions together form a route:

Definition 3.9 (Route of a Train) *A path p_{ij} and the corresponding trajectory function $f_{ij}(\tau)$ provide a route of train i. The sets P_i and $\{f_{ij}(\tau)\}$ provide the set of all possible routes for train i. Denote with r_{ij} a possible route and with R_i the set of all routing possibilities of train i. For the sake of simplicity a fixed route of train itinerary i will be denoted by $\hat{r}_i$.*

Remark 3.10 *Note that due to the periodicity of the timetable, $f_{ij}(\tau)$ occurs with period T in the timetable. Henceforth, routes are thus considered to be periodically repeating tuples $\{p_{ij}, f_{ij}(\tau)\}$ as well.*

Assume that train i is assigned to a route $\hat{r}_i$. Moreover, assume train i passes a track segment $e = (u,v)$. According to the applied safety rules, the segment e has to be reserved for a certain time by train i. There are two safety systems which differ conceptually in using either a fixed block rule allowing only one train per block (most of today's safety systems and *e. g.* ETCS Level 2, see Appendix A or [The Union of European Railway Industries UNIFE, 2005]), or a so-called moving block system, in which the resources are dynamically allocated according to speed, acceleration, etc. (*e. g.* ETCS Level 3, see Appendix A or [The Union of European Railway Industries UNIFE, 2005]). Although the specification of ETCS Level 3 is not completely fixed yet, the ideas are already available. Conceptually up to ETCS Level 2 nothing changes except the location and time of reserved blocks. However, the block lengths for ETCS Level 2 will be much smaller than today and especially more regular near stations. The two systems are outlined in Figure 3.5 showing path-time diagrams.

Whenever a resource element is used, it has to be reserved beforehand. $\underline{\tau}^i(v)$ denotes the beginning of the reservation of a vertex v for train i and $\overline{\tau}^i(v)$ the end of this

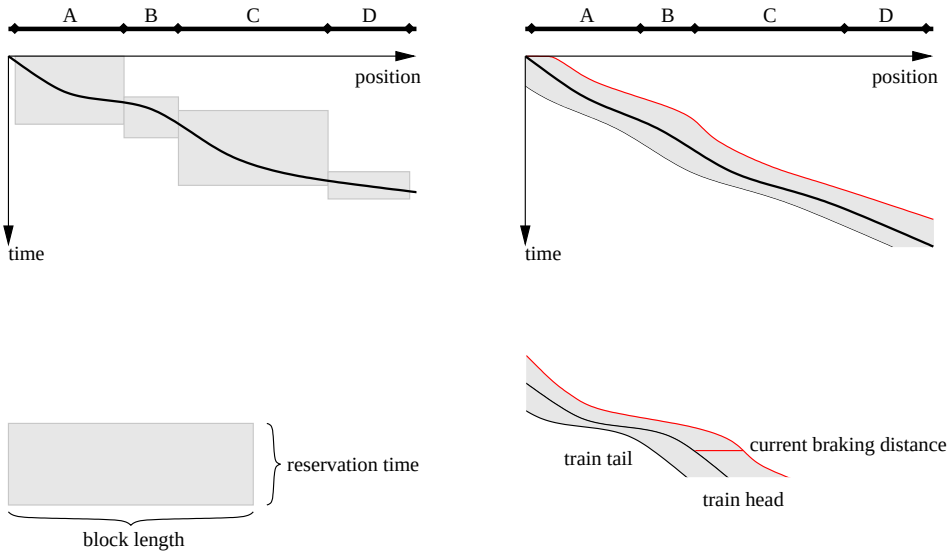

Figure 3.5: Different safety systems. The left figure shows a fixed block safety system. The right side of the figure shows the safety system as it could look like for ETCS Level 3, where the trains have to maintain a minimal distance that could be used for emergency stops (moving-block system).

reservation. These times can be seen as the time of allocation and de-allocation of the resource element v. Since the moving blocks safety system will not be introduced in the near future, fixed block safety systems are henceforth focused on. When using fixed block safety systems, edges—or blocks—are reserved as a whole. The point in time when the reservation begins and ends depends on the system. Whenever a train passes a track-side location device, block allocations and de-allocations take place. After the train leaves the block and a corresponding signal has been transmitted, the block is free for a reservation for any other train. In Figure 3.6 this reservation process is outlined.

The reservation procedure takes place before a train enters a track segment e. With the help of the trajectory functions $f_{ij}(\tau)$ reservation beginning and ending can be obtained. Henceforth assume the values $\underline{\tau}^i(v)$ and $\overline{\tau}^i(v)$ to be given. Note that the length of a block is conceptually not important. The shorter the blocks are, the more the system looks like a continuous system like ETCS Level 3. The main difference between a system used today and ETCS Level 2 is the length of the blocks. In ETCS Level 2 a block-length of about fifty meters is aimed for in stations, whereas today no typical block length exists. A block length can be up to several kilometers. Another difference between a conventional block system and ETCS Level 2 is the time of allocation and de-allocation of the track segment. With ETCS Level 2, the allocation

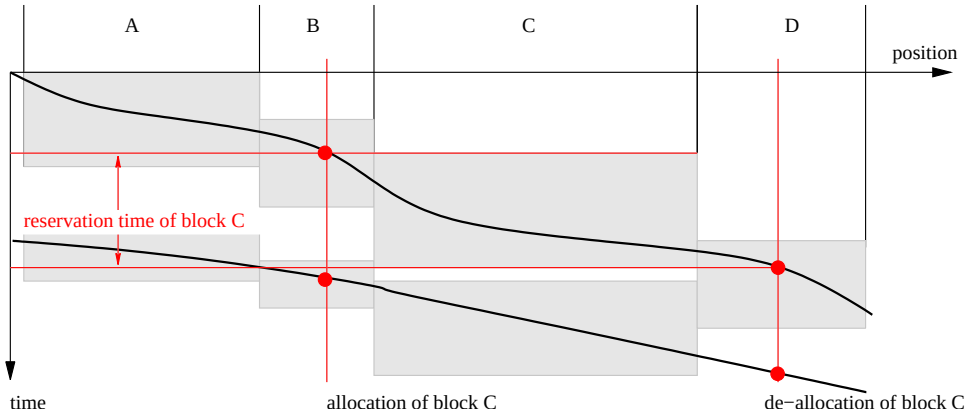

Figure 3.6: Allocation and de-allocation of track blocks. Installed track-side devices detect passing trains. While passing these devices either pervious block reservations are canceled or reservations for next blocks start.

takes place later than in a conventional system and the de-allocation time is earlier, resulting in a shorter reservation time. However, conceptually they are equal and only differ in the determination of the beginning and ending time of a block reservation.

In fact, the reservation time of edge e can be seen as a box around the trajectory of train i. No other train's trajectory is allowed to cross this box and hence the box can be seen as a protection of the train's itinerary. Subsequent edges of the route r_{ij} have their own boxes that have been determined in the same way. The box sequences on the left side in Figure 3.5 and Figure 3.6 are called a *protected time band*. The benefit of introducing ETCS Level 2 is a narrower *protected time band* due to the shorter block lengths and reservation times. These *protected time bands* will be used in the next section to detect conflicts among the train itineraries.

3.3 Detection of Conflicts

In order to decide whether a given timetable is feasible, we need to know which itineraries are mutually exclusive. Since the trains share a common rail network, not every combination of possible routes is feasible. There are configurations that are not allowed. To decide whether two routes r_{ij} and r_{kl} are compatible or not, a confirmation for insufficient minimal headway will be constructed. Henceforth assume that two routes $\hat{r}_i$ and $\hat{r}_k$ are passing a common track segment $e = (u,v)$ while a fixed block safety system is used.

Two routes are said to be conflicting or incompatible on an edge e, if the *protecting time bands* of the two routes are overlapping. Note that due to the periodicity T of a timetable there are more possible conflicting routes than without the modulo calculations, since the time point $\tau \in [0,T)$ is identified with time point $\tau + kT$, $k \in \mathbb{Z}$.

The definition of conflicting routes has to take the periodicity into account.

Definition 3.11 (Conflicting Routes) *Let $\hat{r}_i$ and $\hat{r}_k$ be two routes of two different train itineraries i and k sharing a common edge e. The two routes $\hat{r}_i$ and $\hat{r}_k$ are incompatible or conflicting on edge e if and only if there is a position w on the track segment corresponding to e, such that*

$$[\underline{\tau}^i(w), \overline{\tau}^i(w)]_T \cap [\underline{\tau}^k(w), \overline{\tau}^k(w)]_T \neq \emptyset$$

i. e. the two routes are incompatible or conflicting if and only if the reservation intervals of the corresponding itineraries overlap at least once while traveling on edge e. The routes $\hat{r}_i$ and $\hat{r}_k$ are said to be compatible if for each shared edge e the routes are not conflicting. Incompatible routes are denoted by $\hat{r}_i \nleftrightarrow \hat{r}_k$ and analogously, $\hat{r}_i \leftrightarrow \hat{r}_k$ indicates compatible routes.

3.3.1 The Fixed Block Safety System

Let train i travel on e from u to v. Using a fixed block safety system, edges or blocks are reserved as a whole, and hence the allocation time is the same for every position w on e. The same argument also holds for the de-allocation of the resources and thus for the two end vertices u and v the reservation interval is equal:

$$\underline{\tau}^i(u) = \underline{\tau}^i(v) =: \underline{\tau}^i(e) \qquad\qquad \overline{\tau}^i(u) = \overline{\tau}^i(v) =: \overline{\tau}^i(e)$$

$\underline{\tau}^i(e)$ and $\overline{\tau}^i(e)$ denote allocation and de-allocation time of train i for edge e. As a consequence it does not matter whether two trains travel in the same or opposite directions in order to determine conflicting routes. The following Lemma shows how conflicting routes are detected:

Lemma 3.12 *With the above notation, the routes $\hat{r}_i$ and $\hat{r}_k$ of two different train itineraries i and k are conflicting with respect to a fixed block safety system, if and only if for a shared edge e the following equation holds:*

$$0 \in [\underline{\tau}^k(e) - \overline{\tau}^i(e), \overline{\tau}^k(e) - \underline{\tau}^i(e)]_T$$

Proof. Using Definition 3.11 and the monotonicity of the trajectory functions, the routes $\hat{r}_i$ and $\hat{r}_k$ are conflicting if and only if their reservation intervals overlap:

$$[\underline{\tau}^i(e), \overline{\tau}^i(e)]_T \cap [\underline{\tau}^k(e), \overline{\tau}^k(e)]_T \neq \emptyset$$

Since these two intervals are calculated modulo T the situation can be outlined on a circle (see Figure 3.7). The two intervals are not disjoint if and only if one starting point is inside the other interval:

$$\underline{\tau}^i(e) \mod T \in [\underline{\tau}^k(e), \overline{\tau}^k(e)]_T \quad \vee \quad \underline{\tau}^k(e) \mod T \in [\underline{\tau}^i(e), \overline{\tau}^i(e)]_T$$

By the subtraction of $\underline{\tau}^i(e)$ and $\underline{\tau}^k(e)$ respectively, this is equivalent to

$$0 \in [\underline{\tau}^k(e) - \underline{\tau}^i(e), \overline{\tau}^k(e) - \underline{\tau}^i(e)]_T \quad \vee \quad 0 \in [\underline{\tau}^i(e) - \underline{\tau}^k(e), \overline{\tau}^i(e) - \underline{\tau}^k(e)]_T$$

Multiplying the second equation by -1 implies

$$0 \in [\underline{\tau}^k(e) - \underline{\tau}^i(e), \overline{\tau}^k(e) - \underline{\tau}^i(e)]_T \quad \vee \quad 0 \in [\underline{\tau}^k(e) - \overline{\tau}^i(e), \underline{\tau}^k(e) - \underline{\tau}^i(e)]_T$$

And hence, the routes $\hat{r}_i$ and $\hat{r}_k$ are incompatible on e if and only if

$$0 \in [\underline{\tau}^k(e) - \overline{\tau}^i(e), \overline{\tau}^k(e) - \underline{\tau}^i(e)]_T$$

□

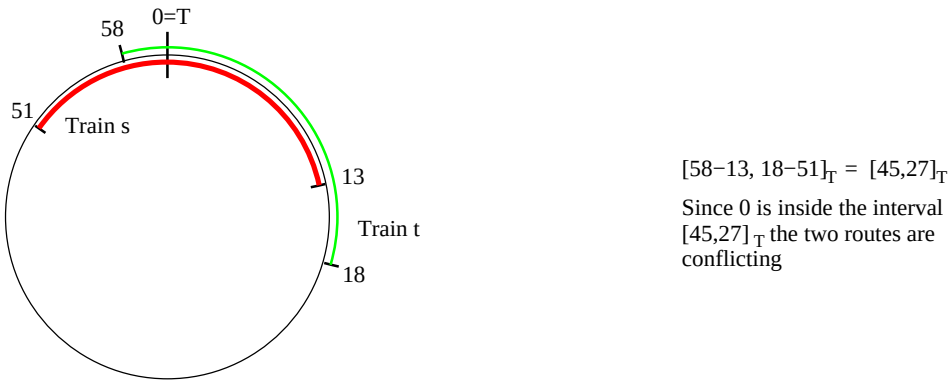

Figure 3.7: Incompatible routes. Two trains are using the same edge. Since the corresponding reservation intervals are not disjoint, the routes are incompatible.

In order to check whether two routes are conflicting in a fixed block system, it is only necessary to know when the reservations start and end for the corresponding trains. In contrast to ETCS Level 3, the calculations are almost independent of the acceleration behavior of the trains. In fixed block systems only passing times at vertices must be known in order to calculate the start and end times of reservations. Otherwise the trajectory function does not play any role.

3.3.2 Modeling ETCS Level 2

Since the exact locations of the ETCS Level 2 devices determining the blocks have not yet been defined, the blocks are also not yet available in the applied test topologies. However, in station areas the block length is assumed to be short. Thus, ETCS Level 2

can be approximated by the following rule: *The time distance between two trains passing any position w must not be smaller than a certain threshold* μ. This value μ depends on the two train types, the order of the trains (i travels in front of k or vice versa), and is assumed to be smaller than the period time T. The parameter μ is called *train succession time*. In order to approximate ETCS Level 2, 90 seconds is usually used for μ. In order to compare with a safety system used nowadays, the fixed blocks are also neglected, but the minimum time distance μ is set to 150 seconds for all train pairs.

However, the minimal time distance between a train k and a train i in front of k is a parameter that could also depend on position, speed and acceleration (and maybe some other environmental variables such as track slope, curvature, adhesion, ...). Denote by $\text{SAFE}(k,i)$ the train succession time between train i and train k, which follows train i. Henceforth assume that $\text{SAFE}(k,i)$ is independent of the position, speed and acceleration of the trains, and that $\text{SAFE}(k,i)$ is larger than the braking time to a full stop for train k. This allows the simplified modeling of a globally defined safety time between two train types.

Since in the model using train succession times, blocks are not allocated and de-allocated, the detection of conflicts differs from the procedure described above. Assume two train routes $\hat{r}_i$ and $\hat{r}_k$ use a common edge $e = (u,v)$. If the trains travel in opposite directions on e then there is no difference to the previous block reservation system. The second train has to wait until the first train has left the edge and hence de-allocated the first vertex on its itinerary.

Assume that the corresponding trajectories $\hat{f}_i(\tau)$ and $\hat{f}_k(\tau)$ are linear and monotonically increasing functions on every edge, except for u and v, where $\hat{f}_i$ and $\hat{f}_k$ may be constant. When the trains travel in the same direction (say from u to v), the conflict detection has to be adapted. The *protected time band* of $\hat{f}_i(\tau)$ is not a box anymore, but a parallelogram in which $\hat{f}_i(\tau)$ is embedded. The *protected time band* defining functions $\underline{f}$ and $\overline{f}$ are calculated as follows (see Figure 3.8). $\underline{f}$ and $\overline{f}$ indicate the earliest and latest possible allocation and de-allocation of any point w in the range of $\hat{f}_k(\tau)$ for train k, *i. e.*

$$\begin{aligned}\underline{f}(\tau) &= \hat{f}_k(\tau + \underline{z})\\ \overline{f}(\tau) &= \hat{f}_k(\tau + \overline{z})\end{aligned}$$

where the two vectors $\underline{z}$ and $\overline{z}$ are defined as the translation vector of $\hat{f}_k(\tau)$ to meet $\underline{f_i}(\tau)$ and $\overline{f_i}(\tau)$ respectively.

$$\begin{aligned}\underline{z} &= \max\{\underline{z}_u, \underline{z}_v\}\\ \underline{z}_u &= \overline{\tau}^k(u) - \underline{\tau}^i(u) + \text{SAFE}(i,k)\\ \underline{z}_v &= \overline{\tau}^k(v) - \underline{\tau}^i(v) + \text{SAFE}(i,k)\end{aligned}$$

similarly define

$$\begin{aligned}\overline{z} &= \min\{\overline{z}_u, \overline{z}_v\}\\ \overline{z}_u &= \underline{\tau}^k(u) - \overline{\tau}^i(u) - \textsc{Safe}\,(k,i)\\ \overline{z}_v &= \underline{\tau}^k(v) - \overline{\tau}^i(v) - \textsc{Safe}\,(k,i)\end{aligned}$$

If the trajectory $\hat{f}_k$ is between $\underline{f}$ and $\overline{f}$ with respect to the timetable periodicity T, then the $\hat{f}_k$ is within the *protected time band* of $\hat{f}_i$ and hence, the two routes are incompatible. There exists a simple check whether $\hat{f}_k$ is inside or outside the *protected time band*:

Lemma 3.13 *With the above notation, the routes $\hat{r}_i$ and $\hat{r}_k$ of two different trains i and k are incompatible with respect to a safety system using train succession time, if and only if for at least one shared edge $e = (u,v)$ the following equation holds:*

$$0 \in [\overline{z}, \underline{z}]_T$$

Proof. Since the two vectors $\underline{z}$ and $\overline{z}$ indicate translations, $\hat{f}_k$ is outside the *protected time band* if and only if the translation vectors are both pointing in the same direction. □

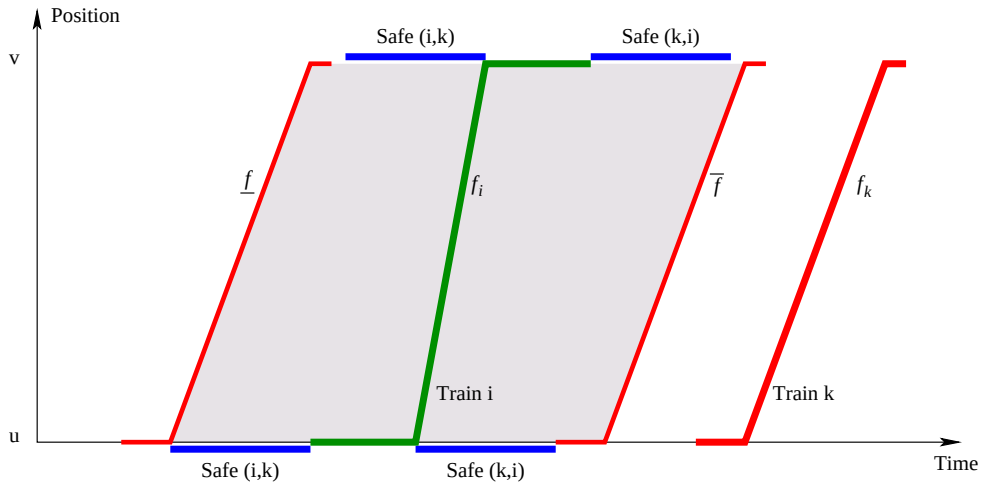

Figure 3.8: Situation when the trains trajectories are piecewise linear.

This modeling of the safety system is not exact but it is sufficient in order to investigate timetables. In reality, train schedules not only have to fulfill safety restrictions but are faced with numerous other «soft» constraints. The train succession time model respects only some crucial facets of the system. Nevertheless, all results obtained will

give hints of what could be done, which claims seem to be reasonable, and which expectations are impossible to satisfy.

With the help of Lemmas 3.12 and 3.13 it is possible to decide whether two routes r_{ij} and r_{kl} are compatible or not in the following way: Let $e_1, \ldots, e_m$ be the shared edges of their paths. If there is no such edge, then the routes are completely separated and cannot conflict. Otherwise, there is a potential conflict on each of the edges e_1 to e_m. For each edge the Lemmas then decide whether the two routes are compatible or not. If there is at least one edge on which the routes r_{ij} and r_{kl} are conflicting, then r_{ij} and r_{kl} are incompatible.

However, in order to check whether a draft timetable is feasible or not, each train has to be assigned a route. Finding a configuration of routes which are mutually compatible is difficult and the content of the next two sections.

3.4 Conflict Graph Model and the Train Routing Problem

A set of n trains is given, each having a set $R_i = \{r_{i1}, \ldots, r_{im_i}\}$, $i = 1, \ldots, n$ of m_i possible routes connecting their start and end points in the station area. Recall the assumption that itineraries have been split (see the Remark 3.6 about itinerary splitting on page 29). Hence the initial vertex is either a portal node or a station node and the end point either a station node or a portal node.

A basis for deciding whether an assignment of routes to trains is feasible or not, is the following conflict graph model. It is independent of the underlying safety rules, which are only used to determine whether two routes are conflicting or not. Construct a set C of incompatible routes as follows:

Definition 3.14 (Conflict Set) *The set C consists of all pairs of routes (r_{ij}, r_{kl}) that are incompatible or belong to the same train itinerary, i. e.*

$$C = \left\{ (r_{ij}, r_{kl}) \mid (i = k \wedge j \neq l) \vee r_{ij} \nleftrightarrow r_{kl} \right\}.$$

Remark 3.15 *Since only one route is needed for each train itinerary all pairs of routes belonging to the same set R_i are also contained in the conflict set. However, this definition of the conflict set yields two different forms of incompatibility (see Equations (3.3) and (3.4) below).*

Following the same modeling ideas as [Zwaneveld et al., 1996] an independent set problem to solve the feasibility problem is introduced. In a first stage only the question of whether a certain instance (topology, train service intention and timetable) is feasible or not, *i. e.* whether there is an assignment of trains to routes for *all* trains, is considered.

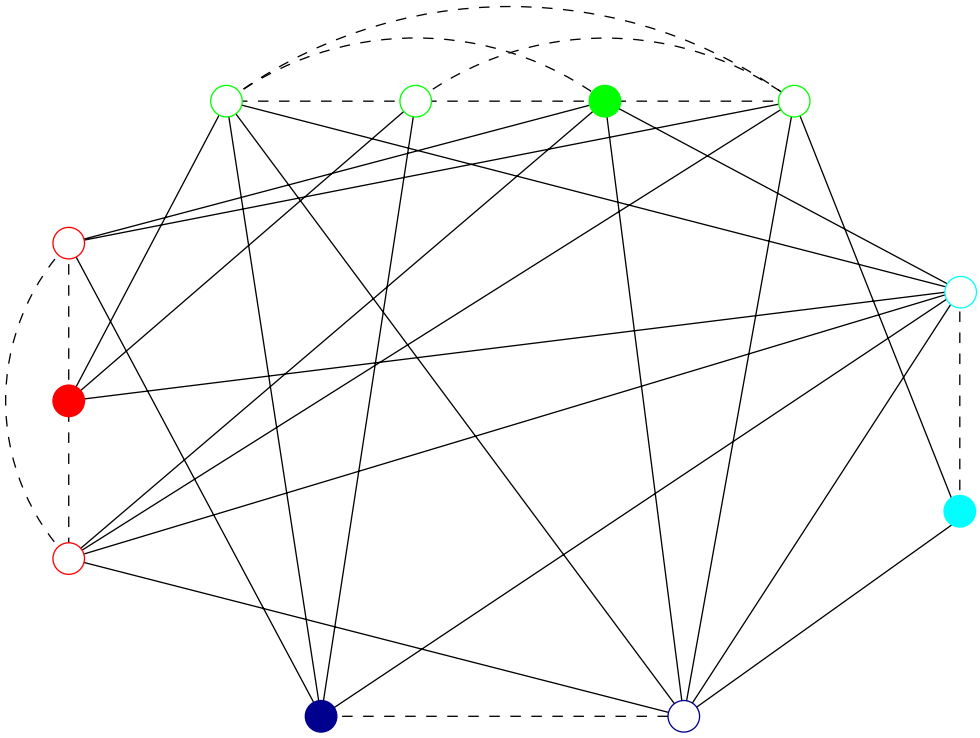

Figure 3.9: Conflict graph. Four trains traveling through a network. Each possible route is represented by a node, routes that conflict correspond to edges that connect the two respective nodes. The clique restrictions are outlined by dotted lines that only connect nodes belonging to the same train itinerary. A possible set of compatible routes is represented by the filled vertices.

The graph for the independent set problem is constructed as follows: The vertices are the elements of $R_1, \ldots, R_n$ and the edges correspond to the elements of C. Note that within the resulting graph all vertices corresponding to routes of the same train form a clique, *i. e.* a complete subgraph. Hence a feasible solution to the routing problem is given by a maximum independent set—with a size equal to the number of train itineraries (see Figure 3.9 for a graphical representation). The problem can be formulated as an Integer Linear Program (ILP) as follows. Let

$$x_{ij} = \begin{cases} 1 & \text{if route } r_{ij} \text{ is assigned to train } i \\ 0 & \text{otherwise} \end{cases}$$

Remember that m_i denotes the number of different routes train i may take and n the number of trains. With the definition of the x_{ij} the train routing problem is formulated as follows:

Problem 3.16 (Train Routing Problem)

$$\max \quad \sum_{i=1}^{n} \sum_{j=1}^{m_i} x_{ij} \tag{3.2}$$

$$s.t. \quad \sum_{j=1}^{m_i} x_{ij} = 1 \qquad \text{for all } i = 1, \ldots, n \tag{3.3}$$

$$x_{ij} + x_{kl} \leq 1 \qquad \text{for all } r_{ij} \nleftrightarrow r_{kl} \tag{3.4}$$

$$x_{ij} \in \{0,1\} \tag{3.5}$$

Equation (3.3) assures that only one vertex per clique induced by a train service is used. Equation (3.4) assures that no incompatible pair of routes is in the chosen set of vertices. The objective function (3.2) maximizes the number of vertices satisfying these constraints, *i. e.* the objective value indicates the cardinality of the independent set.

A tight upper bound on the maximum cardinality is known due to the clique structure of the graph. The maximum cardinality is equal to the number of train itineraries if and only if it is possible to find a conflict-free routing for all n itineraries. Graphs consisting of n cliques, which are interconnected by edges, are called *n-clique graphs*. An assignment of 0 or 1 to all variables x_{ij} respecting all constraints in Problem 3.16 proves the draft timetable to be feasible. Moreover, such an assignment corresponds to a *feasible schedule* L for which the timetable and designated routes for all trains are decided. Henceforth, $L = \{x_{11}, \ldots, x_{nm_n} \in \{0,1\} \mid (3.3) \text{ and } (3.4) \text{ are satisfied}\}$ will be used to denote a feasible schedule. Similarly, L will also be used to denote a node set of the conflict graph, meaning that L consists of those nodes whose corresponding x_{ij} is set to 1.

In [Zwaneveld et al., 1997] it is shown that Problem 3.16 is $\mathcal{NP}$-complete with respect to the infrastructure (topology) and the train service intention, *i. e.* the number of train itineraries through the station area. More precisely, it is shown that the train routing problem is $\mathcal{NP}$-complete by a reduction from SAT, as soon as each train has at least three different routing possibilities.

However, if the layout of the railway station area is fixed, then a dynamic programming approach solves the problem in polynomial time with respect to the number of trains. Fixing the layout of the station causes the number of possible routes to be bounded by a constant number and thus, the number of vertices in the conflict graph is no longer variable but bounded by a constant too. It should be noted that these results are interesting from a theoretical point of view. Their practical value is limited, since the presented algorithm needs $O(n^{I+1}\log n)$ time, where n denotes the number of trains and I the number of relevant track sections. For example, the Bern test cases contain about 20 trains per half hour and about 200 relevant track sections. Therefore, it is futile to solve real-world instances with the dynamic programming approach and thus, other algorithms have been used so far in order to solve the train routing problem in practice (*e. g.* see [Zwaneveld et al., 1996]). The next paragraph describes a new method to find independent sets of n-clique graphs.

3.5 The Fixed Point Iteration Method

In station areas the number of switches is usually large, thus the number of possible variants to reach a point v from u is large as well, resulting in large sets R_i. Because of this tremendous number of vertices and edges in the conflict graph, the Branch & Bound Algorithm described in [Zwaneveld et al., 1996] seems unapplicable; meaning that solutions to the Train Routing Problem (3.2)–(3.5)—where *all* trains have an assigned route—are hard to find in sufficiently short time. However, by accepting a probabilistic algorithm that finds solutions to feasible instances in the majority of the cases and fails for all infeasible instances, the train routing problem can be solved in reasonable time. This will now be shown.

In order to find a maximum independent set for Problem 3.16, an algorithm specially developed to solve *Constrained Semi-Assignment Problems* is adapted (see [Burkard, 2000]). The basic idea of this heuristic is to make a continuous relaxation of the boolean decision variables and then evolve, starting from an interior point, towards an extremal point, which corresponds to a feasible assignment. In his thesis Burkard shows by empirical comparisons with Tabu Search that the algorithm finds solutions to the *Constrained Semi-Assignment Problem* of similar target function value in less computation time [Burkard, 2000].

The main advantages of using this adapted heuristic for solving the train routing problem are that it allows the clique structure of the graph to be efficiently exploited

Algorithm 3.2 Fixed Point Iteration to find an Independent Set in n-clique graphs

Input: G with vertex set $V = \{v_{ij} \mid v_{i1}, \ldots, v_{im_i},$ build a clique, for $i = 1, \ldots, n\}$ and edge set $E \subseteq \{(v_{ij}, v_{kl}) \mid i \neq k\}$ (G a n-clique-graph)
Output: A set of vertices $I \subset V$ such that vertices v_{ij} and v_{kl}, $i \neq k$, belonging to I are not connected by an edge. I is an independent set of size n or $I = \emptyset$.

Initialization: Choose a maximum number of iterations T, a maximum number of randomizations T' and a maximum number of restarts T''. For every v_{ij} assign a value p_{ij}^0 such that

$$0 < p_{ij}^0 < 1 \quad \text{and} \quad \sum_{j=1}^{m_i} p_{ij}^0 = 1 \qquad \forall\, i \in \{1, \ldots, n\} \tag{3.6}$$

Iteration: While $t < T$ and $p_{ij}^{t+1} \neq p_{ij}^t$ for some p_{ij} do:

$$p_{ij}^{t+1} := \frac{p_{ij}^t \prod_{r_{kl} \nleftrightarrow r_{ij}} (1 - p_{kl}^t)}{\sum_{s=1}^{m_i} p_{is}^t \prod_{r_{kl} \nleftrightarrow r_{is}} (1 - p_{kl}^t)} \qquad i \in \{1, \ldots, n\} \quad j \in \{1, \ldots, m_i\} \tag{3.7}$$

Randomization:

(i) Randomly choose a clique i.

(ii) According to their distribution p_{ij}^T, $j = 1, \ldots, m_i$, randomly choose a vertex $\hat{v}_{ij}$ in clique i and add it to I.

(iii) Remove all vertices $v_{i\cdot}$ and all neighbors of $\hat{v}_{ij}$ from G, *i. e.* remove all v_{kl} with $(\hat{v}_{ij}, v_{kl}) \in E$.

(iv) Normalize the values p_{kl} of the remaining vertices and restart the randomization process until no clique is available.

(v) If I has size n then return I, else restart the randomization procedure (at most T' times) or restart the whole algorithm (at most T'' times).

and that many varying solutions can be found by including randomization (see below).

For each x_{ij} a new variable p_{ij} is introduced with $p_{ij} = 1$ if train itinerary i is assigned to its j-th route and $p_{ij} = 0$ otherwise. Allowing all values $p_{ij} \in (0,1)$ Algorithm 3.2 is conducted.

The intuition behind the iteration step (Equation (3.7) in Algorithm 3.2) is as follows. The interpretation of the p_{ij} as probabilities for choosing route r_{ij} for train i might be thought of as some Bayesian interference process. $(1 - p^t_{kl})$ can be interpreted as the probability of not choosing r_{kl} as the designated route of train k. The probability of selecting route r_{ij} for train i is penalized by all conflicting routes to r_{ij}. If such a conflicting route has a high probability of being selected, then its influence in the penalty is larger. Hence the nominator acts as a penalty function. In each iteration step t, the probability of selecting route r_{ij} is adjusted by $(1 - p^t_{kl})$ for all conflicting routes $r_{kl} \nleftrightarrow r_{ij}$. The probability of selecting a route is also adjusted by the probability of not choosing the alternative routes of the same train itinerary due to the clique structure of the graph. Thus, a feedback exists that increases the probability of choosing a likely route, which considerably accelerates convergence.

The denominator preserves $\sum_{s=1}^{m_i} p^t_{is} = 1$ for all cliques i and acts therefore as a normalizer. Without this normalization, the probabilities would rapidly tend to zero. In theory, attractive fixed points ($p^{t+1}_{ij} = p^t_{ij}$ for all i and j) of the iteration correspond to solutions of the routing problem, assuming that a solution exists. Yet, non-attractive fixed points which do not meet all restrictions may exist. This has been shown for a more general setting by Cochand (see [Cochand, 1993]).

Example 3.17 The example demonstrates the fixed point iteration procedure. The input graph consists of 3 cliques A, B and C, each one having 3 nodes. Figure 3.10 shows the graph without the clique edges, *i. e.* edges between nodes belonging to the same clique. The initial probabilities are randomly chosen, such that Equation (3.6) holds. The table shows the probabilities for some example iterations.

Node	Begin	Iter.1	Iter.2	Iter.5	End
A1	0.2	0.34	0.27	0.12	0
A2	0.5	0.10	0.01	0	0
A3	0.3	0.56	0.72	0.88	1
B1	0.2	0.38	0.34	0.21	0.1
B2	0.7	0.58	0.65	0.79	0.9
B3	0.1	0.04	0.01	0	0
C1	0.3	0.10	0.02	0	0
C2	0.6	0.85	0.97	1	1
C3	0.1	0.05	0.01	0	0

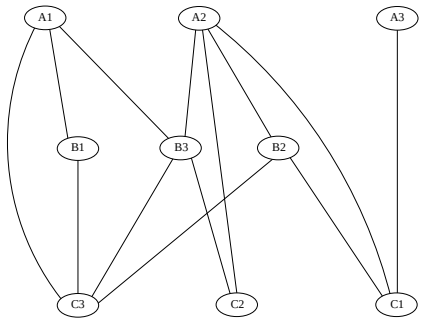

Figure 3.10: Example to illustrate the fixed point iteration procedure.

Already after a few number of iterations, quite good trends can be observed. For example the probability that A2 is chosen as the independent node for clique A is nearly zero after five iterations. This can be verified as follows: If A2 would be chosen, then B1 has to be chosen too, since B2 and B3 are directly connected to A2. However, A2 is connected to C1 and C2 and B1 to C3 and hence none of the nodes in clique C can be chosen either. Similar arguments hold for B3, C1 and C3. When the iteration stops, A3 and C2 are almost certainly chosen, and either of B1 and B2 too—both completions are feasible. However, a solution (A1, B2, C2) would also be feasible, yet it can only be reached from another starting point. ◇

Some remarks about the behavior of the algorithm conclude this section. Results for random instances of n-clique graphs as well as for four test cases from the Bern station area will be shown in Section 3.6.

Termination: If the input instance provides no independent set of size n then the algorithm obviously would not terminate without the upper bound T'' of maximal possible restarts. If the algorithm does not have any success with this number of restarts, then the instance is believed to be infeasible and $I = \emptyset$ is returned.

Choosing the initial distribution: The p_{ij} can be seen as probabilities that train i chooses route j (see above). As there is no obvious reason to prefer one route over another, the uniform distribution could be chosen for the initialization phase, *i. e.* $p_{ij} = \frac{1}{m_i}$. This choice has two drawbacks:

(i) Recall that the iteration itself is deterministic. At the end of the iteration phase, the result is a probability distribution over the routes for each train. By distributing the starting probabilities uniformly, the algorithm is deterministic up to the end of the iteration phase. However, having a wide variety of solutions is preferred to a sparse range of solutions (see also Chapter 5). By choosing the initial values of p_{ij} randomly, a large variation in the output is attained.

(ii) The second drawback is more subtle. Equation (3.3) of the program formulation favors solutions $p_{ij} = \frac{1}{m_i}$. If such a starting point is chosen, the probability to remain in this point is not zero. Moreover, there are different types of regions of attraction. Cochand and Burkard show that also interior points may be attractive (see [Cochand, 1993] and [Burkard, 2000]). The positive aspect is that the region of attraction for interior fixed points is not a ball. In every ε-ball around such an interior fixed point there is always a direction that is not attractive. By fixing the initial distribution to the uniform distribution, the starting point of the iteration procedure could be inside an attractive region for an interior fixed point. Usually those interior

fixed points are bad, in the sense that the randomized rounding procedure often fails. Hence, they should be avoided. Once more, by choosing a randomly generated starting point, the probability that the algorithm will fail because of converging to an interior fixed point is reduced. By restarting the algorithm the probability of failing with the randomized rounding procedure all the time is reduced.

Number of iterations: In practice, due to rounding off in computer arithmetic, the denominator in (3.7) might vanish. Due to the limited precision of computing, fixed points not corresponding to solutions may occur. Therefore, the iteration is stopped after a small number of iterations. Empirical evidence showed that after some hundreds of iterations—say 200 to 400 iterations—quite good «trends» in the distribution of the p_{ij}^T can be seen. Hence the randomized rounding procedure can be started early and the probability of success is still high—provided that a solution exists.

Number of randomizations: If no feasible solution is found after T' randomizations, a new fixed point is calculated from a new starting point. Experience showed that T' can be set to a small value—not larger than 10, as the distribution of the p_{ij}^T was in most of the cases either of high quality or of very poor quality. This means that either all randomizations succeeded or none at all. Thus it is better to spend computational time in evaluating many different starting points rather than in evaluating many randomizations.

Number of restarts: If the procedure fails to find a solution in the first round, the whole algorithm is restarted. The fixed point iteration procedure is deterministic, except for the random starting point and the randomization procedure to round to an integral solution. By varying the starting points not only bad fixed points are avoided but many fixed points leading to entirely different solutions are explored.

The problems are the infeasible or the difficult instances. The algorithm may fail for a thousand restarts, but is it certain that the instance has no solution? No, as the distributions and randomizations might have lead to bad points by misfortune! There is no confirmation that an instance has no solution. And therefore an instance can only be *believed* to be infeasible. However, by increasing the number of restarts, iterations and/or randomizations, this belief can be strengthened (at the cost of speed).

Circling of the iteration procedure: There are theoretical instances where the iteration procedure circles, *i. e.* after a couple of iterations the calculated point $p = (p_{11}, \ldots, p_{nm_n})$ has already been previously calculated. However, this does not seem to occur in practical instances.

3.6 Computational Results

For the Bern test scenarios, it is possible to have over 4'000 different paths for a single train itinerary—when allowing all possible paths. By excluding sinuous tours (see Figure 3.2), it is still possible to have over 800 different routes per train. In a single train intention service scenario, 15 to 20 different train itineraries within 15 to 30 minutes are scheduled.

The fixed point iteration algorithm is not only used to verify the feasibility of the Bern scenarios; while generating new timetables (see Two-Level Approach in Chapter 2) draft timetables have to be checked for their feasibility too, and hence Algorithm 3.2 is often applied. Therefore almost 300 randomly generated test instances have been solved, each with a different setting of number of nodes and cliques, density of the conflict graph and whether a feasible solution is provided or not. Providing a feasible solution is done constructively by randomly selecting one vertex per clique and removing the edges between the selected vertices beforehand. This procedure allows the generation of feasible instances. The number of iterations is set to 100, the number of randomizations to 5, and the number of restarts to 10. The main focus is the speed and reliability of the algorithm illustrated in the following figures. The test runs are performed on an AMD Athlon64 3400+ processor with 1.5GB RAM.

Since in the Bern test cases all train itineraries have to travel through Bern main station, it is possible to separate the east- from the the west-side itineraries. This partition speeds up the computation time considerably. In the scenarios of 2003, 19 itineraries are scheduled within half an hour on both sides east and west. However, the size of the conflict graph is quite different. Whereas on the east-side the sum of all routes is about 5'500, on the west-side there are only 1'400 nodes in the conflict graph. The number of edges in the Bern East 2003 scenario is about 740'000 and about 70'000 for the Bern West 2003 scenario. In both scenarios the fixed point iteration method was able to find an independent set within a minute.

Even the condensed Bern West 2020 scenario is solved within a minute. Note that in this scenario there are 11 itineraries in 15 minutes whereas in the corresponding 2003 scenario there are 19 itineraries in 30 minutes, resulting in about 400 additional nodes in the conflict graph. While the Bern East 2003 scenario provides a conflict graph with about 5'500 nodes and 740'000 edges, the conflict graph of the Bern East 2020 scenario contains 6'800 nodes and 7'100'000 edges. The number of itineraries changed from 19 in half an hour to 16 in a quarter of an hour. Whereas the Bern West 2020 scenario is still solvable within a minute, it lasts about half an hour to solve the Bern East 2020 scenario.

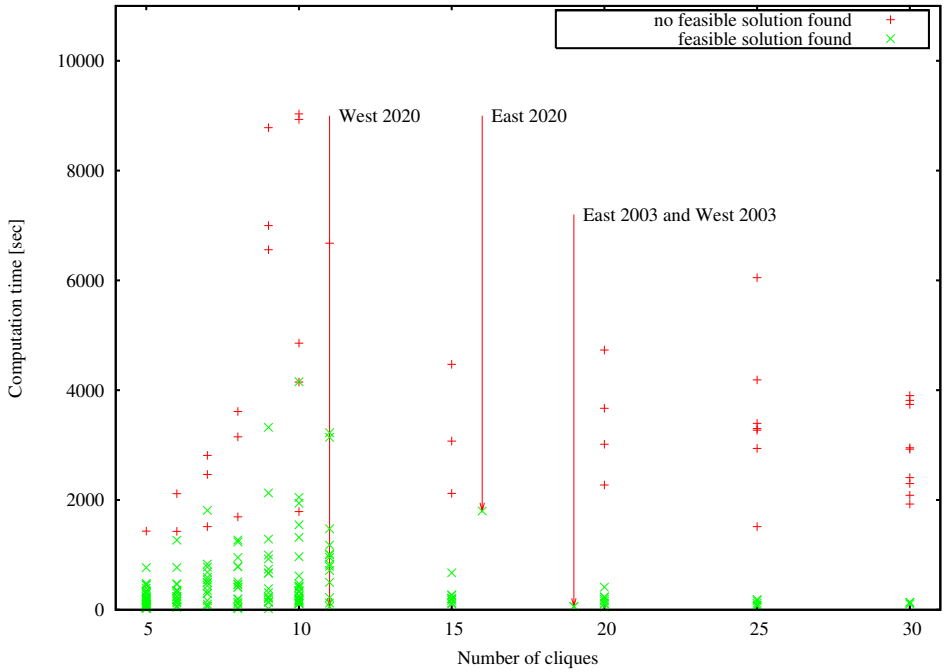

Figure 3.11: Number of cliques and speed of the fixed point iteration method. In most of the cases solutions are found in less than an hour. Hence, this plot suggests stopping the algorithm after about an hour.

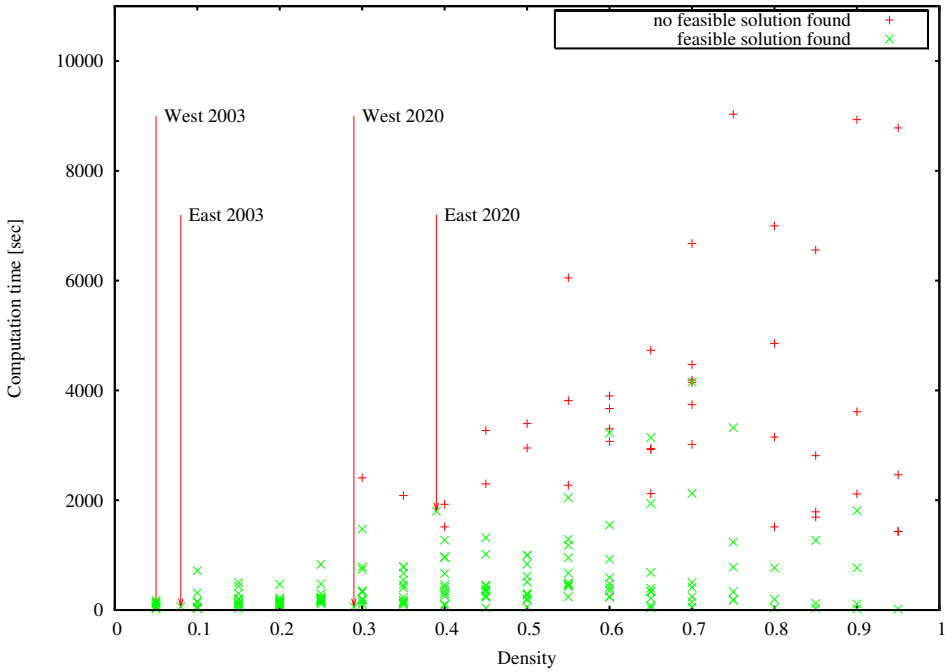

Figure 3.12: Computation time depending on the density of the conflict graph. Clique edges are not counted.

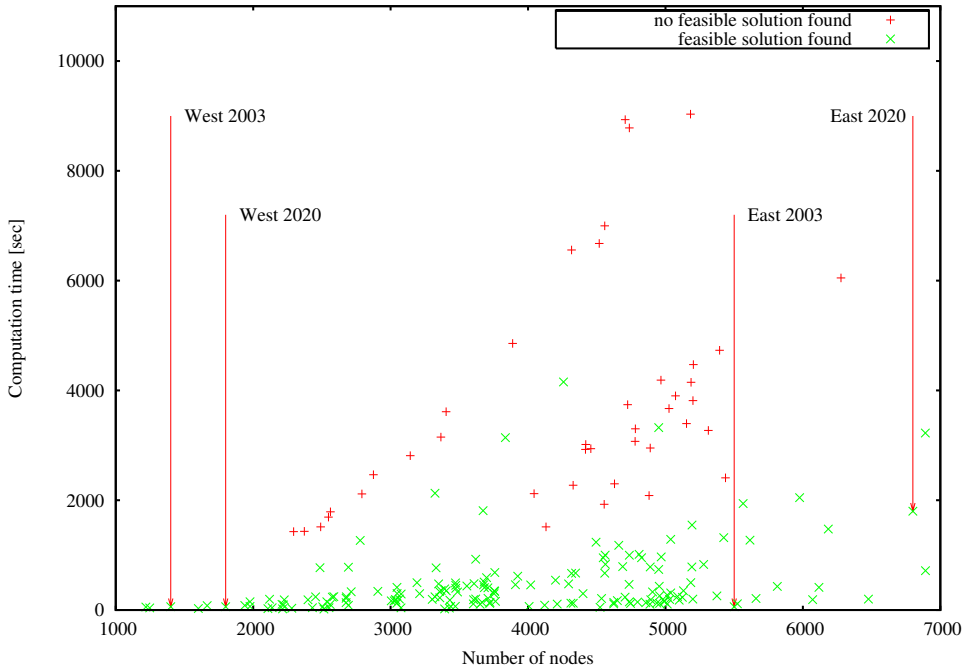

Figure 3.13: Computation time depending on the number of nodes in the conflict graph.

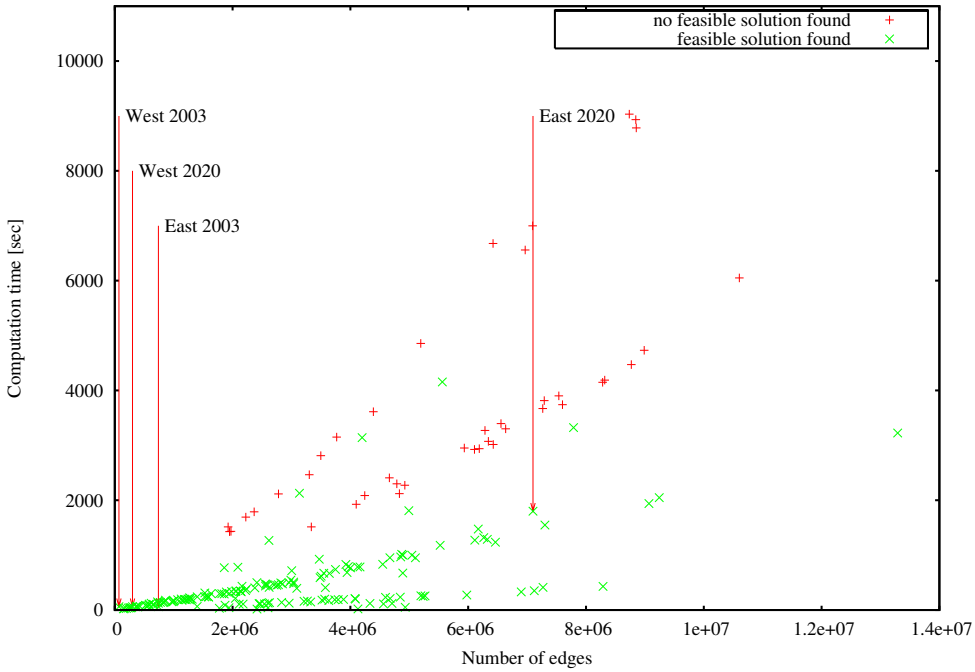

Figure 3.14: Computation time depending on the number of edges in the conflict graph. Clique edges are not counted.

The previous figures show that in most of the cases, solutions are found within an hour of computation time, independent of density or number of cliques, nodes and edges. Since the algorithm is used to check the feasibility of a draft timetable, it has to be reliable. The following figure shows that in about four out of five cases the algorithm is able to find a solution—with a rather small value for the number of iterations (100) and modest number of restarts (5) . In about 9 percent of all cases however, the algorithm fails, although a solution exists. Most of these cases are scenarios with high graph density or a high number of cliques. By adjusting the parameters describing the number of restarts, iterations, and randomizations, the algorithm becomes more reliable—at cost of speed. Allowing 400 iterations, most of the cases could be resolved, although not all. Yet, the time to resolve these problems increases by more than a factor of 4, since restarts are allowed. Although the probability that the algorithm misses a feasible instance can be lowered by increasing the number of iterations and restarts, there is still a small chance for the algorithm to fail.

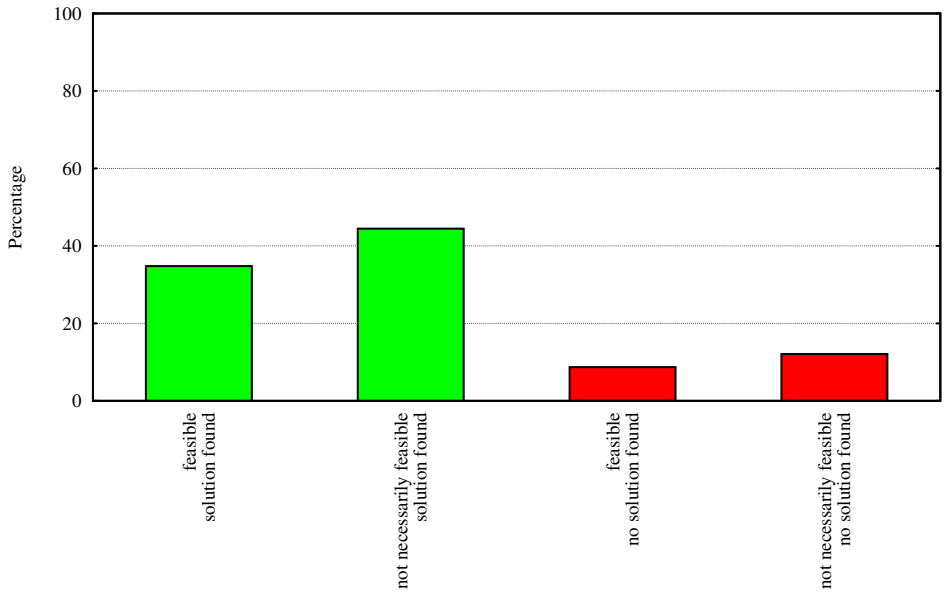

Figure 3.15: Reliability of fixed point iteration method. The plot shows that the algorithm is able to find an independent set in about 80 percents of the cases. In 9% of the cases it fails, and in about 11% it has not found a solution, yet in these cases it is not clear whether a solution exists at all.

In general, maximum independent set problems are difficult problems. They are

known to have no constant-factor approximation algorithm unless $\mathscr{P} = \mathscr{NP}$. However, the train routing problem yields a special structured graph—a n-clique graph. The proposed fixed point iteration algorithm solves most instances within an hour. By using this probabilistic algorithm the train routing problem can be solved for relevant real-world instances in sufficiently short time. In contrast to other methods based on Linear Programming approaches (*e. g.* see [Zwaneveld et al., 1996]), the fixed point iteration method generates a wide variety of routing choices for feasible instances. This diversity in routing possibilities will be important, when looking at optimization problems in Chapter 5 where the objective function (3.2) will be replaced by other objective functions aiming at stable schedules.

Chapter 4

Train Routing, Uncertainty and Stability Measures

> The thirty spokes unite in the one nave; but it is on the empty space for the axle, that the use of the wheel depends. Clay is fashioned into vessels but it is on their empty hollowness that their use depends. The door and windows are cut out from the walls to form an apartment, but it is on the empty space within that its use depends.
>
> *Lao Tzu (around 600 B.C.)*

Usually trains do not run exactly on time during operation. In a satisfactory scheduling solution, a planned route for a train should be free even if trains arrive or depart with some (small) delay. In order to increase stability of railway operations the routes for all trains should be chosen in a way that they interfere with each other as little as possible.

The crucial element in the planning of routes is the uncertainty of the behavior in reality. During a day, the system of running trains evolves and the environment variables change. For example, due to heavy traffic (more freight trains) or expanded passenger volume on the network (peak hours), trains may experience delays and the elaborate plans might fail. These delays have manyfold reasons. Ullius analyzed sources of delays of the timetable introduced on December 12th 2004 during the first six weeks of 2005. The main cause of delay was inherited delay (*e. g.* waiting for connecting trains), followed by delays due to safety installation failures and defective

rolling stock (see [Ullius, 2005] and [Ullius, 2004b]).

No system is able to eliminate all sources of delays, but the *propagation of the delays must be prevented* as much as possible. Therefore, the influence of uncertainty in the system of running trains has to be discussed. If trains are delayed, interventions are possibly needed in order to provide safe rail traffic. In the previous chapter a schedule has been determined—if possible—by computing an arbitrary routing for a given draft timetable. In this chapter functions to evaluate and analyze schedules are introduced. These functions are then used in the next chapter to compute *stable* schedules. As the schedule should not break down under a small perturbation of the timetable, these functions will act as objective functions in an optimization problem. In Chapter 6, the question to answer is whether the elaborate schedules are suitable and applicable, *i. e.* whether or not a schedule is executable.

For further consideration, the following problem setting is used. A schedule L is given respecting a given train service intention and a corresponding timetable. Each train i is assigned to a route $\hat{r}_i$ consisting of the nodes $v_1^i, v_2^i, \ldots v_{n_i}^i$. A timetable perturbation is defined as a collection of timetable perturbations for each train i in the system:

Definition 4.1 (Timetable Perturbation of a Train) *A timetable perturbation of a train i belonging to the train service intention consists of a triple* (X_i, v^i, ϑ_i) *where*

(i) X_i *is the amount of delay for train i*

(ii) $v^i \in \{v_1^i, v_2^i, \ldots v_{n_i}^i\}$ *indicates the source node of the delay (after passing node* v^i *all passing times of the subsequent nodes of the route are delayed by* X_i*)*

(iii) ϑ_i *indicates the relative reveal time of the delay*

In a timetable perturbation $(\mathbb{X}, V, \Theta)$, the permutation triple (X_i, v^i, ϑ_i) for each train i is specified in order to define $\mathbb{X} = (X_1, \ldots, X_n)$, $V = (v^1, \ldots, v^n)$, and $\Theta = (\vartheta_1, \ldots, \vartheta_n)$. X_i and v^i give the place and amount of the occurring delay, whereas the relative reveal time ϑ_i specifies when the train dispatchers receive the information about the delay of train i. Note that ϑ_i should be negative, meaning that a delay in v^i is known ϑ_i time units *in advance*. With the help of ETCS, ϑ_i will be smaller in future than today. Furthermore, ϑ_i is only assumed to be less or equal to 0.

Although it is possible that several timetable perturbations per train may occur, in further consideration it is assumed that delays only occur at the first node of a designated route. Moreover, in the model—if not stated otherwise—a timetable perturbation $(\mathbb{X}, V, \Theta)$ only consists of initial nodes, *i. e.* the set V contains only the vertices $v_1^1, v_1^2, \ldots, v_1^n$. Since itineraries are assumed to be split between incoming and outgoing itineraries (see Remark 3.6), perturbations are assumed to occur at the portal nodes for incoming trains and at the platform nodes for outgoing trains. Moreover,

it is assumed that these delays are the only sources of disturbance; there are no track blockages, engine failures or similar unusual events. Note that under these assumptions the trajectories of delayed trains are only translated in time, yet the shape is not affected, since only the time of the initial node is affected.

The «shortcoming» in the considerations of Chapter 3 is the $\{0,1\}$-system of conflicting routes—either there is a conflict between two routes or not. Now imagine that two routes are not in conflict, but the time difference between using the same track element e is very small. Obviously nobody would select those two routes to build the schedule, if there are other alternatives. Hence concepts are needed that take «almost-conflicts» into account. The crucial questions are:

(i) How large is the minimal time distance between two routes?

(ii) Is the schedule expected to fail often and what is the probability that a schedule will fail?

Hence, the main subject of interest is the stability of a schedule. In order to measure the stability of a schedule, stability indicators have to be defined. The schedule is then called stable if the stability indicator does not exceed a certain, given threshold.

The chapter is organized as follows: After introducing the *Time Slot Concept*, which is independent of any delay distribution (Section 4.1), the concepts of the *Expected Number of Conflicts*, *Schedule Failure Probabilities*, and *Critical Trains*, which are based on some delay distributions, are defined (Section 4.2). All four concepts—which will be referred to as stability indicators—are used to evaluate and characterize a given schedule from different viewpoints. Examples of discrete and continuous delay distributions are shown thereafter (Section 4.4), and will be used in Chapter 5 to calculate stable schedules and in Chapter 6 to generate sample delays. Recall that the problem is still to find an assignment of n routes to the n itineraries that are provided via the train service intention and a corresponding timetable. This assignment should respect the timetable and all safety rules. However, stability aspects should now be taken into account too.

4.1 Deterministic Stability: Time Slots

In a reliable schedule, there should be a minimal time interval between any two trains using the same resource in order to have a minimal amount of headway buffer time between any two trains. This is important in order to cope with small operational deviations.

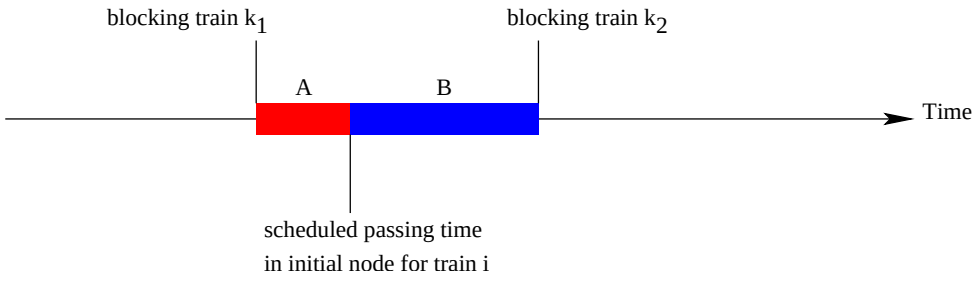

Figure 4.1: Calculation of Time Slots.

4.1.1 Simple Time Slot

Consider an assignment L of conflict-free routes $\hat{r}_i \in R_i$—one route for each train i. Assume that L is a feasible solution to the routing problem. The *simple time slot* for train i is then defined as follows:

Definition 4.2 (Simple Time Slot) *The simple time slot $F_i(L)$ is defined as the time interval in which train i may depart from its initial node for its planned route $\hat{r}_i$ without conflicting with any other train's route $\hat{r}_k$, $k \neq i$, under the assumption that all other trains run on time.*

The *simple time slot* characterizes the minimal time distance of a train i to any of the other trains. Train i may have any delay within its time slot and still it will not interfere with any other train. The length of a time slot $|F_i(L)|$ is calculated for a given assignment L (assuming that all trains except i are running exactly on time) in the following way (see Figure 4.1):

$$|F_i(L)| := \underbrace{\min_{k \neq i} f(\hat{r}_i, \hat{r}_k)}_{B} + \underbrace{\min_{k \neq i} f(\hat{r}_k, \hat{r}_i)}_{A}$$

where $f(\hat{r}_i, \hat{r}_k)$ is the amount of time train i may be delayed and still not be in conflict with the designated route $\hat{r}_k$ of train k. Note that the value is symmetric in the sense that $f(\hat{r}_i, \hat{r}_k)$ also stands for the amount of time that train k may arrive prior to schedule if train i is on time. There are several points on the time slot concept to be discussed:

(i) Given a routing and thus a solution L, the time slot concept is not complex; yet it is not trivial to determine the time slots of the trains. The calculation of $f(\hat{r}_i, \hat{r}_k)$ is sketched in Figure 4.2 and presented in Algorithm 4.1. The algorithm for determining the simple time slot of train k projects the reservation of the trains using the same track resources as train k onto a time axis representing potential passing times of train k at its initial vertex. Train k is then not allowed to pass its initial starting point, if the passing time is covered by any of the projections.

Algorithm 4.1 Time Slot Calculation

Input: A feasible schedule L with designated routes $\hat{r}_i$ for all trains i and the timetable periodicity T.

Output: A set of functions $g_i(\tau)$ indicating whether τ could serve as departure time for train i from the initial node of the itinerary. Moreover, the time slots $F_i(L)$ and their length $|F_i(L)|$ are returned.

$g_i(\tau)$:=0 for all trains i and all $\tau \in [0,T]$ *// free*
τ_i:= departure time of train i at its initial node
for all edges e in the topology **do**
  for all routes $\hat{r}_i$ passing e **do**
    $\underline{\tau}^i(e)$:= blocking begin of train i on e and $\overline{\tau}^i(e)$:= blocking end of train i on e
    for all routes $\hat{r}_k \neq \hat{r}_i$ passing e **do**
      $d = \overline{\tau}^k(e) - \underline{\tau}^k(e)$ *//reservation time of k*
      $s = \underline{\tau}^k(e) - \tau_k$ *//travel time of k until e is reserved*
      $a_i = \underline{\tau}^i(e) - s - d$ and $b_i = \overline{\tau}^i(e) - s$
      $g_k(\tau) = 1$ for all $\tau \in [a_i, b_i]_T$ *// blocked (*)*
    endfor
  endfor
endfor
//Note: $g_k(\tau_k) = 0$ for all k; otherwise the schedule is not feasible!
for all trains k **do**
  $\min_{i \neq k} f(\hat{r}_i, \hat{r}_k) = \min\{t | t \geq 0 \wedge g_k(\tau_k - t) = 1\}$ *(**)*
  $\min_{i \neq k} f(\hat{r}_k, \hat{r}_i) = \min\{t | t \geq 0 \wedge g_k(\tau_k + t) = 1\}$ *(**)*
  $|F_k|_L = \min_{k \neq i} f(\hat{r}_k, \hat{r}_i) + \min_{i \neq k} f(\hat{r}_i, \hat{r}_k)$
endfor

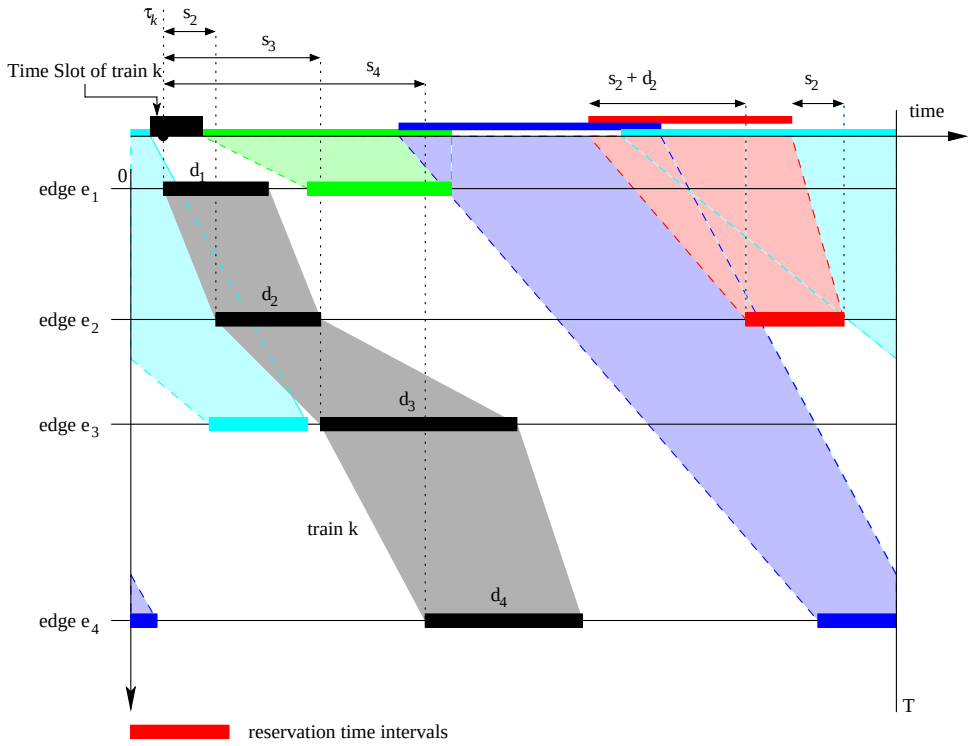

Figure 4.2: Calculation of $f(\hat{r}_i, \hat{r}_k)$. Train k passes edges e_1, e_2, e_3, and e_4. Four other trains use the same edges too and therefore block the departure time τ_k of train k in its initial node. The calculation in the innermost *for*-loop is a projection of the reservation intervals to a time axis. The time slot consists of all points around the scheduled passing time of the initial vertex on the time axis that are not used for the projection of the different reservation intervals. An example projection is illustrated for the reservation interval on edge e_2.

The simple time slot is a time interval around the scheduled passing time of the initial vertex.

(ii) The time slots are dependent on the underlying routing. This makes it difficult to gradually extend a subrouting without falling below a minimal time slot length: In a subrouting k trains are assigned to their routes, whereas $n-k$ trains do not have designated routes yet. However, the interaction between the routes is so high that it seems impossible to develop an incremental algorithm that assigns step-by-step trains to routes in such a way that a minimal time slot length can be guaranteed at the end of the procedure.

(iii) As previously mentioned, the concept of simple time slots is independent of any delay distribution—it is deterministic but depends on the routing. By using the additional information on delays—available for example from large statistics—better routing plans can be derived. A small time slot is not necessarily a bad time slot. If a train runs with very high level of punctuality then its time slot can be smaller than the time slot of a train which is known to have a broad delay spectrum.

Remark 4.3 (Time Complexity) *Let n denote the number of trains in the train service intention and E the number of edges in the topology. With the help of appropriate data structures, Algorithm 4.1 runs in $O(n^2E)$ time.*

An interesting extension to the time slot concept is the following: Let $g_i(\tau)$ not only indicate whether the designated route $\hat{r}_i$ is free (0) or not (1), but how many other routes are blocking the designated route $\hat{r}_i$ for a given point in time. «$g_k(\tau) = g_k(\tau)+1$ for all $\tau \in [a_i, b_i]_T$ for which route $\hat{r}_i$ has not yet blocked route $\hat{r}_k$» then replaces line (*) in Algorithm 4.1. Accordingly, in the terms «$g_k(\tau_k \pm t) = 1$» in lines (**) are replaced by «$g_k(\tau_k \pm t) \geq 1$».

Counting these numbers is helpful, as it can be seen, which trains are affected by a delay of train k. A small number seems to be preferable, since then there are only a few dependencies between $\hat{r}_k$ and the routes of all other trains. In Figure 4.3 an example simple time slot is shown (upper picture). On the x-axis the time is depicted and on the y-axis $g_k(\tau)$ is shown. The train's passing time at its initial (portal) node is 470, the time slot ranges from 307 until 524, *i. e.* the length of the time slot is 218. Note that there is no other possibility for the train to travel on its designated route between 0 and 1200.

Among all time slots corresponding to the n trains, the shortest is of particular interest since it indicates the minimal buffer time each train has. If a train has a delay X such that the departure time in its initial node is still inside the time slot, then no interventions are needed as long as all other trains run exactly on time. Hence the stability of a schedule can be seen as the self-regulation of the timetable with respect

to some (small) delays and thus the deterministic stability of a schedule can then be defined as follows:

Definition 4.4 (Deterministic Stability of a Schedule) *A timetable is called deterministically stable, if a desired minimal time slot for each train running in the system is provided, assuming that all trains run exactly on time.*

4.1.2 Extended Time Slot

If a train i has a delay larger than its time slot length $|F_i(L)|$, interventions are needed in order to guarantee safe running of the system, assuming all other trains are on time. A possible intervention could be a rerouting of some trains inclusive or exclusive train i. Taking the possibility of intervention into account, the time slot concept can be extended as follows. For the calculation of the simple time slot of train i it is required that train i is assigned to a route $\hat{r}_i$. For the extended time slot no designated route $\hat{r}_i$ for train i is required anymore, yet all other trains are assumed to have designated routes. More precisely:

Definition 4.5 (Extended Time Slot) *The extended time slot $\tilde{F}_i(L)$ is defined as the time interval in which train i may depart from its initial node without conflicting with any other train's route $\hat{r}_k$, $k \neq i$ (assuming that all trains except i are running exactly on time).*

Note that train i may then take another route than its designated route. This is a simple intervention that might enlarge the time slot length considerably. Moreover, if the difference between the simple and the extended time slot is significant, then this is an indication for a more «flexible» timetable. Besides using the available network to full capacity and having a stable timetable it is important to have flexibility in the schedule. Whereas in a stable timetable the operator's intervention is prevented as far as possible, a flexible timetable allows to deal with disturbances for which intervention is needed, without having significant and harmful consequences for the customers. Hence the extended time slot is a measure of flexibility in the schedule.

In Figure 4.3 an example of an extended time slot is given in the lower picture for the same example train. The x-axis represents time and on the y-axis the number of free routes $h(\tau)$ is shown, *i. e.* if the train would leave its initial node at time τ then $h(\tau)$ routes are not conflicting with the designated routes $\hat{r}_i$, $i \neq k$. Note the difference of 70 seconds between the extended and the simple time slot. The extended time slot now ranges from 307 until 595, Moreover, there are small time slots from 144 to 167, from 886 to 982, and from 1055 to 1089. Among all possible routes there are some routes unblocked during these time intervals, yet the designated route is not contained in the list of these routes.

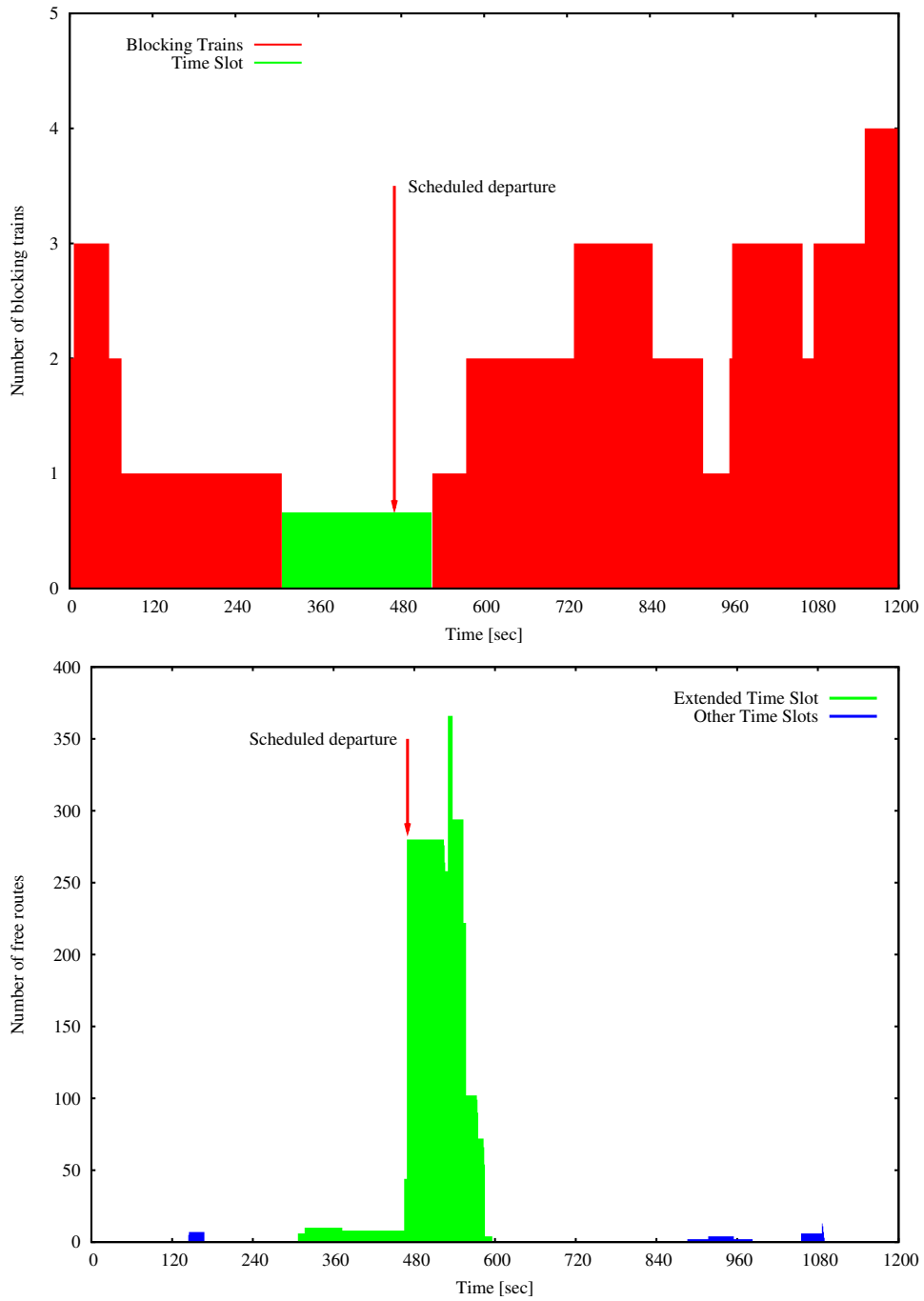

Figure 4.3: Time slot of an intercity train arriving in Bern coming from Zurich using the timetable from 2003.

Remark 4.6 (Time Complexity) *As in Remark 4.3, let n denote the number of trains in the train service intention and E the number of edges in the network topology. Moreover, let N denote the sum of all routing possibilities of all trains, i. e. the number of vertices in the conflict graph. The time needed to determine the extended time slots is calculated as follows: For each vertex in the conflict graph the time slot has to be determined. A time slot computation for a single train takes $O(nE)$ time (see Remark 4.3). Hence, the time to calculate the extended time slots is $O(nE \cdot N)$. Recall that the number of vertices per train in the conflict graph could be as large as $O(2^E)$ (see Remark 3.5). Hence, in the worst case $O(n^2 E 2^E)$ time is needed in order to compute all extended time slots. Even if the number of vertices per train can be delimited, the time needed for determining extended time slots is much longer than for computing simple time slots—even in practice.*

4.2 Probabilistic Stability Measures

The time slot concept is independent of any probability distribution. Now, probabilistic measures will be introduced. Consider a timetable perturbation $(\mathbb{X}, V, \Theta)$, in which V consists of initial nodes only, *i. e.* $(\mathbb{X}, V, \Theta)$ denotes a perturbation on the departure time of the trains at their initial nodes. Moreover, assume that for all trains i an individual and independent probability distribution $P(X_i \leq d)$ is given. For each train there is a physical barrier that defines a lower bound on the negative delays, since a train has a technical speed limit that cannot be exceeded. For an upper bound there is no such physical barrier, yet the further focus will be on small delays.

Now, consider two routes r_{ij} and r_{kl} of two different trains i and k as well as the corresponding delays X_i and X_k. Depending on these delays, the routes do not need to be compatible. Therefore, introduce the following random variable $Y_{ij,kl}$:

$$Y_{ij,kl} = \begin{cases} 1 & \text{if } r_{ij} \nleftrightarrow r_{kl} \text{ according to } X_i, X_k \\ 0 & \text{if } r_{ij} \leftrightarrow r_{kl} \text{ according to } X_i, X_k \end{cases} \tag{4.1}$$

$Y_{ij,kl}$ indicates, whether or not two selected routes are compatible according to the delays X_i and X_k. Consider any draft schedule L and the corresponding assignment of trains to routes. This assignment is not assumed to be feasible, *i. e.* the assigned routes might be incompatible, even if each train runs on time. Denote with $\hat{r}_i$ and $\hat{r}_k$ the assigned routes of trains i and k respectively and with $\hat{Y}_{i,k}$ the corresponding indicator variable that shows, whether or not the two assigned routes $\hat{r}_i$ and $\hat{r}_k$ are compatible.

In order to evaluate schedules, several probabilistic stability measures will be developed and explained in the following paragraphs. All are based on the definition of $Y_{ij,kl}$ and $\hat{Y}_{i,k}$ respectively. The corresponding problems can be visualized in the

following extended graph model (see Figure 4.4): As in Section 3.4 let each route r_{ij} correspond to a vertex v_{ij}. In contrast to Section 3.4, each vertex v_{ij} is now connected to all other vertices v_{kl}. Introduce the following weights on the edges:

$$w_{ij,kl} = \begin{cases} \infty & \text{if } i = k, \textit{ i. e.} \text{ if the trains are equal} \\ 1 & \text{if } r_{ij} \nleftrightarrow r_{kl} \text{ for } X_i = X_k = 0, \textit{ i. e.} \text{ for incompatible routes} \\ P\left(r_{ij} \nleftrightarrow r_{kl}\right) & \text{if the routes are feasible for } X_i = X_k = 0 \end{cases} \tag{4.2}$$

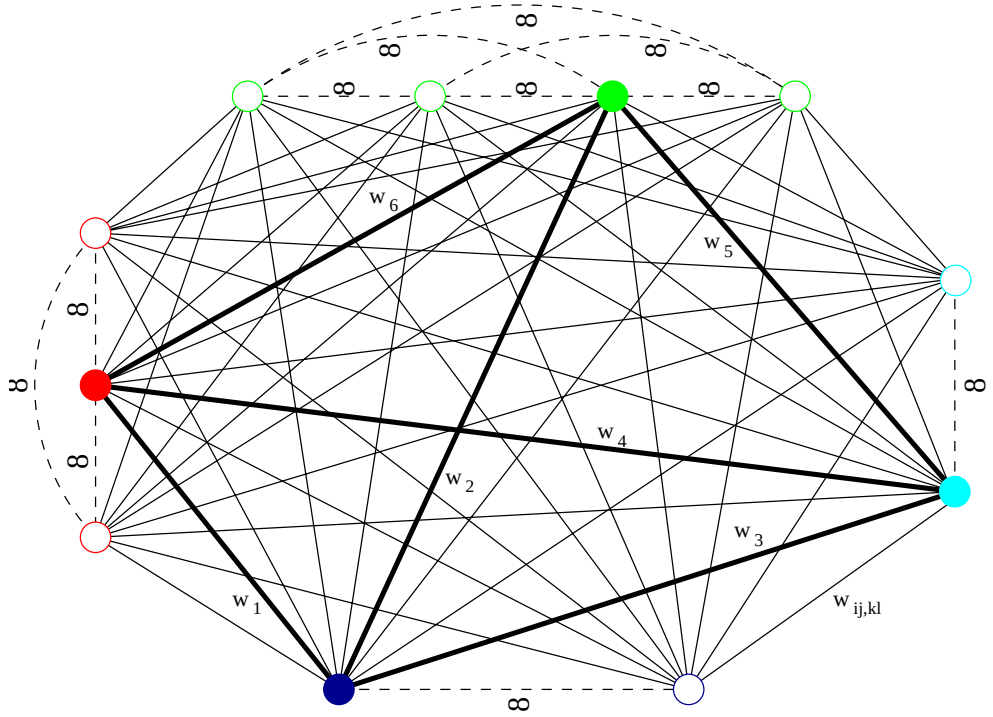

Figure 4.4: Extended conflict graph. All vertices are connected and every edge has a weight. A set of routes induces a clique, shown as thick edges.

A set of selected vertices no longer builds an independent set (since all vertices are connected), yet it induces a *clique*, *i. e.* a subgraph in which all vertices are connected to each other. Note that any selected set of nodes is feasible with respect to the train routing problem (Problem 3.16, Equations (3.2)–(3.5)) if and only if there is no edge in the induced clique having a weight of ∞ or 1. A weight $w_{ij,kl}$ between 0 and 1 indicates the probability that the routes are conflicting, *i. e.* if train i is assigned to its route j and train k to its routes l then $w_{ij,kl}$ is the probability that this assignment is in conflict. It is very difficult to calculate the probability that two routes are in

conflict with each other unless the delay probabilities are assumed to be independently distributed (see Sections 4.3 and 4.5).

Note that the weights $w_{ij,kl}$ are *not* independent. Look at a 3-clique with the trains g, i, and k and the corresponding routes h, j, and l. Then $Y_{gh,kl}$, $Y_{gh,ij}$, and $Y_{ij,kl}$ are not independent since they depend on the delays X_g, X_i, and X_k. Moreover, it might be impossible to find delays X_g, X_i, and X_k such that any given configuration $Y_{gh,kl}$, $Y_{gh,ij}$, $Y_{ij,kl} \in \{0,1\}$ can be satisfied.

According to the probability distribution of X_i and X_k the probability $P\left(r_{ij} \nleftrightarrow r_{kl}\right)$ has to be calculated differently. In Sections 4.3 – 4.5 it will be shown how the probabilities are calculated in order to compute the described graph. As every schedule induces a weighted clique, new probabilistic stability measures will be established through graph characteristics first.

4.2.1 Expected Number of Conflicts

It is of particular interest to know the expected number of conflicts that have eventually to be settled for a given schedule L. If the expected number of conflicting routes is small, the schedule is expected to be insensitive to deviations and hence is assumed to be stable. A large expected value indicates a high sensitivity to perturbations. Moreover, if this value is too high, then the schedule is assumed to be non-operational as there are often deviations causing conflicts that have to be solved. The goal is to find a feasible assignment that minimizes this expected value.

Let L be any given draft schedule—not necessarily feasible—and $\hat{Y}_{i,k}$ the random variables that indicate whether the routes of train i and k are compatible or not. Let Y_C be a random variable defined as:

$$Y_C = \sum_{i,k} \hat{Y}_{i,k} \tag{4.3}$$

Y_C indicates the number of conflicting routes for a timetable perturbation $(\mathbb{X}, V, \Theta)$ in an assignment L. Of course in a feasible assignment and assuming all delays to be 0, Y_C is equal to zero. Yet, in the infeasible case Y_C is larger than zero. Hence, Y_C counts the conflicts in a schedule for given delays $X_1, \ldots, X_n$. The expected number of conflicts is the value $\mathbb{E}[Y_C]$ which gives an indication on the stability of the schedule. It is calculated as follows:

$$\mathbb{E}[Y_C] = \mathbb{E}\left[\sum_{i,k} \hat{Y}_{i,k}\right] = \sum_{i,k} \mathbb{E}\left[\hat{Y}_{i,k}\right] = \sum_{i,k} P\left(\hat{Y}_{i,k} = 1\right) = \sum_{i,k} P\left(\hat{r}_i \nleftrightarrow \hat{r}_k\right) \tag{4.4}$$

The expectation thus corresponds to the sum of all edge weights of the clique, which is induced by the assignment L. The goal is to select vertices as «independent» as possible. «Independence» in this context means that the weight of the induced clique should be as small as possible.

Definition 4.7 (Conflict Stability of a Schedule) *A timetable is called conflict-stable, if the expected number of conflicts does not exceed a desired maximal number of conflicts, assuming that the train delay distributions are given.*

4.2.2 Schedule Failure Probability

The expected number of conflicting routes $\mathbb{E}[Y_C]$ is an indicator of whether the schedule might be stable or not. Yet, this number alone states nothing about the variation of the involved probabilities. The schedule becomes infeasible as soon as the trains have delays such that on some track segment e the scheduled routes of at least two trains become incompatible. If for a schedule L and a timetable perturbation $(\mathbb{X}, V, \Theta)$ the random variable Y_C is zero, then the schedule L is still feasible, whereas $Y_C \geq 1$ means that the schedule fails. Hence, the value $P(Y_C \geq 1)$ is of crucial interest.

Definition 4.8 (Schedule Failure Probability) *The probability $P(Y_C \geq 1)$ is called the Schedule Failure Probability.*

Due to the dependence of the random variables $\hat{Y}_{i,k}$, it is difficult to calculate $P(Y_C \geq 1)$ directly. However, lower and upper bounds can be given that indicate whether or not a schedule is likely to fail. In the following Lemma 4.9 an upper bound and a lower bound are given.

Lemma 4.9 *Using the previous notation, the following upper and lower bounds for the schedule failure probability $P(Y_C \geq 1)$ hold:*

(i) $P(Y_C \geq 1) \leq \sum_{i,k} P\left(\hat{Y}_{i,k} = 1\right) = \mathbb{E}[Y_C]$

(ii) $P(Y_C \geq 1) \geq \max_{i,k} P\left(\hat{Y}_{i,k} = 1\right)$

Proof. Note that the probabilities $P(Y_C = 0)$ and $P(Y_C \geq 1)$ can be written in the following way:

$$P(Y_C = 0) = P\left(\hat{Y}_{1,2} = 0 \wedge \hat{Y}_{1,3} = 0 \wedge \ldots \wedge \hat{Y}_{n-1,n} = 0\right)$$
$$P(Y_C \geq 1) = P\left(\hat{Y}_{1,2} = 1 \vee \hat{Y}_{1,3} = 1 \vee \ldots \vee \hat{Y}_{n-1,n} = 1\right)$$

(i) The first inequality $P(Y_C \geq 1) \leq \sum_{i,k} P\left(\hat{Y}_{i,k} = 1\right)$ follows directly from the Markov Inequality $P(X \geq t) \leq \frac{\mathbb{E}[X]}{t}$:

$$P(Y_C \geq 1) \leq \mathbb{E}[Y_C] = \sum_{i,k} P\left(\hat{Y}_{i,k} = 1\right)$$

(ii) The second inequality follows from the fact that $P(A \wedge B) \leq \min\{P(A), P(B)\}$, and thus:

$$\begin{aligned} P(Y_C \geq 1) &= 1 - P\left(\hat{Y}_{1,2} = 0 \wedge \hat{Y}_{1,3} = 0 \wedge \ldots \wedge \hat{Y}_{n-1,n} = 0\right) \\ &\geq 1 - \min_{i,k} P\left(\hat{Y}_{i,k} = 0\right) \\ &= \max_{i,k} 1 - P\left(\hat{Y}_{i,k} = 0\right) \\ &= \max_{i,k} P\left(\hat{Y}_{i,k} = 1\right) \end{aligned}$$

□

The first inequality gives a good upper bound if the expected number of conflicts is small. However, if the expected number of conflicts is high, *i. e.* larger than 1, then the inequality is replaced by the trivial upper bound 1. Yet, it turns out that other meaningful upper bounds are impossible to find, if nothing is known about the delay probability distributions. Moreover, since the random variables $\hat{Y}_{i,k}$ are not independent, exact bounds are very difficult to calculate. However, the second inequality always derives a non-trivial lower bound.

In order to calculate tighter bounds, the probabilities $P\left(\hat{Y}_{i,k} = 1 \wedge \hat{Y}_{r,s} = 1\right)$ of dependent variables $\hat{Y}_{i,k}$ and $\hat{Y}_{r,s}$ have to be determined. If a good estimation for $P\left(\hat{Y}_{i,k} = 1 \wedge \hat{Y}_{r,s} = 1\right)$ can be found, then with the help of the *Inclusion-Exclusion Principle* tighter bounds could be achieved:

$$\begin{aligned} P(A_1 \vee A_2 \vee \ldots A_n) = &\sum_{i=1}^{n} P(A_i) - \sum_{1=i<j}^{n} P\left(A_i \wedge A_j\right) + \sum_{1=i<j<k}^{n} P\left(A_i \wedge A_j \wedge A_k\right) \\ &- \ldots + (-1)^{n-1} P(A_1 \wedge A_2 \wedge \ldots \wedge A_n) \end{aligned}$$

Using the fact that the absolute value of a certain term is smaller than the absolute value of any of its precedent terms, *e. g.*

$$\left| \sum_{1=i<j<k}^{n} P\left(A_i \wedge A_j \wedge A_k\right) \right| \leq \left| \sum_{1=i<j}^{n} P\left(A_i \wedge A_j\right) \right|,$$

the summation can be truncated and a valid inequality results. Hence for the proba-

bility of a schedule failure the bounds could be tightened as follows:

$$
\begin{aligned}
P(Y_C \geq 1) &= P\left(\hat{Y}_{1,2} = 1 \vee \hat{Y}_{1,3} = 1 \vee \ldots \vee \hat{Y}_{n-1,n} = 1\right) \\
P(Y_C \geq 1) &\geq \sum_{i,k} P\left(\hat{Y}_{i,k} = 1\right) - \sum_{i,k<r,s} P\left(\hat{Y}_{i,k} = 1 \wedge \hat{Y}_{r,s} = 1\right) \\
P(Y_C \geq 1) &\leq \sum_{i,k} P\left(\hat{Y}_{i,k} = 1\right) - \sum_{i,k<r,s} P\left(\hat{Y}_{i,k} = 1 \wedge \hat{Y}_{r,s} = 1\right) \\
&\quad + \sum_{i,k<r,s<a,b} P\left(\hat{Y}_{i,k} = 1 \wedge \hat{Y}_{r,s} = 1 \wedge \hat{Y}_{a,b} = 1\right)
\end{aligned}
$$

Since for the schedule failure probability only bounds can be derived, the schedule failure probability has to be determined by Monte-Carlo Simulations. The bounds act as an indicator of stability, yet not as a measure. The second inequality $P(Y_C \geq 1) \geq \max_{i,k} P\left(\hat{Y}_{i,k} = 1\right)$ shows that the failure probability is bounded from below by the maximum probability that two routes will become incompatible. This can be visualized in the extended conflict graph model (see Figure 4.4): The probability that a schedule has more than one incompatible route is larger than the maximum edge weight of the induced clique. The edge obtaining the maximum weight indicates the two most critical train routes. Hence, these routes demand attention and should be monitored, as they are «the weakest link» in the schedule. Thus, the following stability measure is given:

Definition 4.10 (Structural Stability of a Schedule) *A timetable is called structurally stable, if the maximum probability that two routes will become incompatible does not exceed a desired maximal probability, assuming that the train delay distributions are given.*

The larger the weights, the higher the probability that the schedule plan will fail, as Lemma 4.9 shows. On the other hand, a small maximum weighted edge obviously does not guarantee a small schedule failure probability. Nevertheless, the lower bound for the failure probability is a useful third stability criterion. The larger this value, the higher the probability that the schedule will break down and intervention will be needed.

4.2.3 Critical Train

Usually there are train itineraries that seldom cause problems and other train itineraries that are responsible for most of the interventions. Whereas the expected number of conflicts characterizes a schedule from a global point of view, the weakest link gives a hint of the two trains having the most critical relationship. A further interesting concept considers the most critical train. Consider an assignment L and define

by Y_i the random variable indicating the number of conflicts of train i, *i. e.*

$$Y_i = \sum_k \hat{Y}_{i,k} \tag{4.5}$$

If a train has a delay then the probability that many other trains will be affected by this initial delay is larger for some trains than for others. Hence, the expectation $\mathbb{E}[Y_i]$ is of crucial interest. It can be calculated as follows:

$$\mathbb{E}[Y_i] = \mathbb{E}\left[\sum_k \hat{Y}_{i,k}\right] = \sum_k \mathbb{E}\left[\hat{Y}_{i,k}\right] = \sum_k P\left(\hat{Y}_{i,k} = 1\right) \tag{4.6}$$

Definition 4.11 (Critical Train) *The train having the largest $\mathbb{E}[Y_i]$ among all trains is called the critical train.*

In the extended conflict graph model, this expectation corresponds to the sum of the weights of the edges that are incident to node v_{ij} of the clique induced by L. Note that the overall expectation $\mathbb{E}[Y_C]$, *i. e.* the expected number of conflicts can be calculated by

$$\mathbb{E}[Y_C] = \frac{1}{2}\sum_i \mathbb{E}[Y_i]$$

The distribution of the Y_i is interesting. If all the Y_i are similar in their value, then the trains contribute about the same amount to the overall expected number of conflicts. However, if the values Y_i are distributed with a skew, then there are train itineraries which are more exposed than others. A schedule will fail, if at least one train causes a conflict with another train. In a stable schedule the overall expected number of conflicts should not be concentrated in a small number of trains. Nevertheless, if a schedule will fail too often, then the critical train may be an indicator for the problem.

Definition 4.12 (Cluster Stability of a Schedule) *A timetable is called cluster-stable, if the expected number of conflicts of each train does not exceed a desired maximal number of conflicts per train, assuming that the train delay distributions are given.*

4.3 Probability of Incompatible Routes

The aim is to anticipate deviations and include the insight from historic data into the generation of the plans in order to develop schedules rarely creating needs for interventions.

In the previous section, some probabilistic measures have been developed all making use of the probability of incompatible routes. Now, look at the calculation of

conflicts (see Section 3.3). Let $\hat{r}_i$ and $\hat{r}_k$ be the planned routes for trains i and k respectively. Let e be an edge of the track topology used by both routes. As the calculation of conflicts has shown, the occupation time of both trains is needed to decide whether or not two routes are in conflict. Let $\underline{\tau}^i(e)$, $\overline{\tau}^i(e)$, $\underline{\tau}^k(e)$ and $\overline{\tau}^k(e)$ be the starting and the ending time of the occupation of e of train i and k respectively. Recall that since the area of concern is a station region, the initial node of every train run is either a platform node in the main station or an entry node at one of the portals. Furthermore, since it is assumed that the trains are only delayed in their initial nodes but do not gather any further delay on the way, the following random variables are introduced:

$$X_i = \text{Delay of train } i \text{ at initial node}$$
$$X_k = \text{Delay of train } k \text{ at initial node}$$

It can be decided whether or not the two routes are in conflict by introducing the initial delays X_i and X_k. Of crucial interest hereby is the difference between the delays, as the following Lemma 4.13 shows.

Lemma 4.13 *Using the same notation as above, the designated routes $\hat{r}_i$ and $\hat{r}_k$ are conflicting on e (using a fixed block safety system) if and only if*

$$(X_i - X_k) \ \ mod\ T \in [\underline{\tau}^k(e) - \overline{\tau}^i(e), \overline{\tau}^k(e) - \underline{\tau}^i(e)]_T \tag{4.7}$$

Proof. The difference from the proof of Lemma 3.12 is that the reservation times might be delayed. Since only the passing time in the initial node might be delayed, the two routes are in conflict if and only if (by Lemma 3.12):

$$0 \in [\underline{\tau}^k(e) - \overline{\tau}^i(e) + X_k - X_i, \overline{\tau}^k(e) - \underline{\tau}^i(e) + X_k - X_i]_T$$

Adding $X_i - X_k$ proves the claim. □

In the safety system based on train succession time, the headway between any two trains is at least some threshold μ. Now, a similar argument yields:

Lemma 4.14 *If a train succession time safety system is used, then the routes $\hat{r}_i$ and $\hat{r}_k$ are incompatible on edge $e = (u,v)$ if and only if (compare to Lemma 3.13):*

$$(X_i - X_k) \ \ mod\ T \ \in [\overline{z}, \underline{z}]_T$$

where $\overline{z}$ and $\underline{z}$ are defined as follows:

$$\underline{z} = \max\{\underline{z}_u, \underline{z}_v\}$$
$$\underline{z}_u = \overline{\tau}^k(u) - \underline{\tau}^i(u) + \textsc{Safe}\,(i,k)$$
$$\underline{z}_v = \overline{\tau}^k(v) - \underline{\tau}^i(v) + \textsc{Safe}\,(i,k)$$

and similarly

$$\begin{aligned}\overline{z} &= \min\{\overline{z}_u, \overline{z}_v\}\\ \overline{z}_u &= \underline{\tau}^k(u) - \overline{\tau}^i(u) - \textsc{Safe}\,(k,i)\\ \overline{z}_v &= \underline{\tau}^k(v) - \overline{\tau}^i(v) - \textsc{Safe}\,(k,i)\end{aligned}$$

Lemma 4.13 and 4.14 show that the probability distribution of $X_i - X_k$ is needed in order to calculate the probability that the two routes $\hat{r}_i$ and $\hat{r}_k$ are incompatible:

$$P\,(\hat{r}_i \nleftrightarrow \hat{r}_k) = P\left((X_i - X_k) \mod T \in [\underline{\tau}^k(e) - \overline{\tau}^i(e), \overline{\tau}^k(e) - \underline{\tau}^i(e)]_T\right) \tag{4.8}$$

With the assumption that only small delays are considered, it can be assumed that $X_i - X_k \in [-\frac{T}{2}, +\frac{T}{2}]$. If T is 60 minutes, then «small delays» are assumed to be in the interval $[-5, +25]$ minutes. As an example, an empirical delay distribution shows (see Section 4.4) this delay interval to be sufficient to cover almost all delays. Equation (4.8) can then be simplified as follows. The interval $[\underline{\tau}^k(e) - \overline{\tau}^i(e), \overline{\tau}^k(e) - \underline{\tau}^i(e)]_T$ is transformed into an equivalent interval such that the beginning and the end are between $-\frac{T}{2}$ and $+\frac{T}{2}$. Let τ_a, $\tau_b \in [-\frac{T}{2}, +\frac{T}{2}]$ denote the modified beginning and end of the interval. Then Equation (4.8) can be written as:

$$P\,(\hat{r}_i \nleftrightarrow \hat{r}_k) = P\,(X_i - X_k \in [\tau_a, \tau_b]) \tag{4.9}$$

Remark 4.15 *In the case of compatible routes either both values τ_a and τ_b are positive, or both are negative. This follows from the fact that if 0 is inside the interval $[\tau_a, \tau_b]$ then the routes are incompatible if the trains run on time.*

The two routes $\hat{r}_i$ and $\hat{r}_k$ eventually not only share one edge e but several edges $e_1, \ldots, e_n$. The routes are compatible if and only if they are compatible on all shared edges $e_1, \ldots, e_n$. Hence the probability that the two routes are incompatible is:

$$\begin{aligned}P\,(\hat{r}_i \nleftrightarrow \hat{r}_k) &= P\,(\hat{r}_i \nleftrightarrow_{e_1} \hat{r}_k \vee \ldots \vee \hat{r}_i \nleftrightarrow_{e_n} \hat{r}_k)\\ &= P\left(\exists e_h :\ (X_i - X_k) \mod T \in [\underline{\tau}^k(e_h) - \overline{\tau}^i(e_h), \overline{\tau}^k(e_h) - \underline{\tau}^i(e_h)]_T\right)\\ &= P\left((X_i - X_k) \mod T \in \bigcup_{h=1}^{n} [\underline{\tau}^k(e_h) - \overline{\tau}^i(e_h), \overline{\tau}^k(e_h) - \underline{\tau}^i(e_h)]_T\right)\end{aligned} \tag{4.10}$$

or equivalently for the simplified Equation (4.9)

$$= P\left(X_i - X_k \in \bigcup_{h=1}^{n} [\tau_{a_h}, \tau_{b_h}]\right) \tag{4.11}$$

In fact the crucial part is to calculate the distribution of $X_i - X_k$, *i. e.* the convolution of X_i and $-X_k$ if the distributions of X_i and X_k are known. Then the probability of whether or not two routes are in conflict can be calculated. Note that by altering the distribution of X_i and X_k the model also changes and hence the stability measures of a schedule L change too. In the next sections some delay distributions are shown as well as the calculation of Equations (4.8) and (4.9).

4.4 Delay Distributions

Knowledge about train delays is needed in order to develop schedules in which the anticipation of delays is included. Usually delays of trains are small—say up to a few minutes. If a train has a longer delay, it receives a special treatment and dispatchers have to act depending on the current situation and usually no easy solution is available. For these rather unusual events, the interventions are manyfold and made under enormous time pressure. Often rigorous interventions, such as early termination of the train run or cancellations of trains follow in order to restore the plan as far as possible. Inconvenience for the passengers—which should be avoided of course—is often the result. However, in order to still maintain a good service, such consequences are often inevitable. Yet, what are *small* delays and which delays are considered to be *large*?

In the past, several research projects analyzed train delays and the structure of their distribution (see *e. g.* [Ullius, 2004b] and [Ullius, 2005] or [Weidner, 1997], [Lüthi, 2005]). It turns out that most delays are less than a quarter of an hour. Henceforth, small delays are assumed to be less than 15 minutes, although this amount of delay is unusual for node stations. However, *small delays* could also be defined differently; for the models and calculations this is irrelevant.

A stable timetable should absorb these cases of small delays, whereas it is aimless to require a timetable to absorb larger delays. Hence, it is not necessary for a timetable to be stable for *any* delay scenario but the «common» cases of delays should be resolved successfully most of the time. This is even more important if a timetable is tighter and has less flexibility to resolve delays.

Many railroad companies today have—relating to customers—service goals stating that a certain share of trains should not be more than x minutes delayed upon arrival. Since 1998, the Swiss Federal Railways have the goal that more than 75% of all trains have at most an arrival delay of 1 minute and 95% of all trains have 4 or fewer minutes of arrival delay. Figure 4.5 shows the percentage of trains that met this goal for the years 1998 – 2004. The actual numbers can be found at *http://mct.sbb.ch/mct/infrastruktur*.

Ullius describes how to gather such information on train delays and their probability distributions [Ullius, 2004b]. A general overview of the quality of service is provided

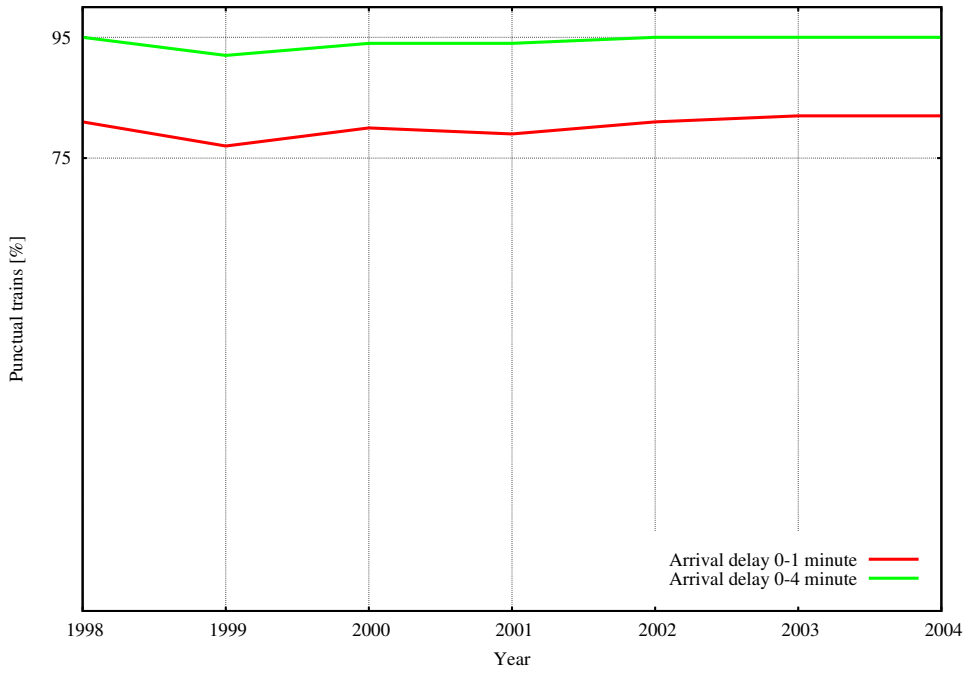

Figure 4.5: SBB's delay statistics since 1998. The 75% goal has always been achieved, yet only in the past few years the 95% goal could be satisfied.

in [IRSE, 2004]. Since trains stopping at platforms are not allowed to depart early, two different delay distributions have to be considered: *Arrival delay distributions* and *departure delay distributions*. Figure 4.6 shows an example of an arrival (top) and a departure (bottom) delay distribution.

Usually available data is discrete, imprecise, and lacks accuracy, as the methods of measuring are not standardized. The crucial questions are *where* and *when* delays should be measured. Is it the time when the engine passes a certain point; or when the last train coach passes? Is it the beginning or the end of the platform or a point in between that indicates the position to measure? Or is it the time when the train has just stopped that counts? At the moment policies to measure train delays are seldom consistent. Therefore, it is difficult to compare train delays for different stations or even between different companies.

However, schedules should not be so tight that every second counts. For example, the Swiss Federal Railways give a sharp margin for ETCS Level 2 at 10 seconds and a soft margin at 15 seconds. This means that the reservation blocks of two trains should have a time distance of 15 seconds or more. If it is only 10 seconds, the schedule is still operational, but a time gap shorter than 10 seconds is considered to be only practical in extreme situations (although it is feasible). Hence, it is not so important whether a train has a delay of 27 or 35 seconds; both delays are about 30 seconds.

In order to model train delay distributions, continuous or discrete distributions can be used. Discrete train delay distributions have the advantage that neither assumptions on the distribution have to be made nor parameters have to be estimated. The distributions are uniquely described by the cumulative distribution function. An important and interesting possibility opens up by using an ETCS Level 2 safety system: It is possible to manage the trains in such a way that delays do not just occur and are randomly distributed, but that the trains have more or less predefined discrete delays, *e. g.* $k\tau$, where τ is a time distance and $k \in N$ and $N \subset \mathbb{Z}$. In other words—due to the usage of ETCS Level 2—dispatchers have the possibility to manage a delayed train in such a way that it does not disturb the current schedule too much, since it is assumed that they can assign delayed trains to new open time slots more easily. This active managing of the available buffers is beyond the scope of this thesis, yet a virtual idealized delay distribution—the *pulse delay distribution*—will be introduced and analyzed in order to get a feeling of whether or not this approach is promising.

The drawback of discrete distributions is that either a tremendous number of measurements must be available or that the data has to be categorized into «bins». The distribution may look different according to the bin size. Moreover, a narrow bin size is only possible if huge amount of empirical data is available. A continuous delay distribution has the advantage that its parameters can be estimated with relatively few measurements. Therefore, using a continuous train delay distribution can be advantageous.

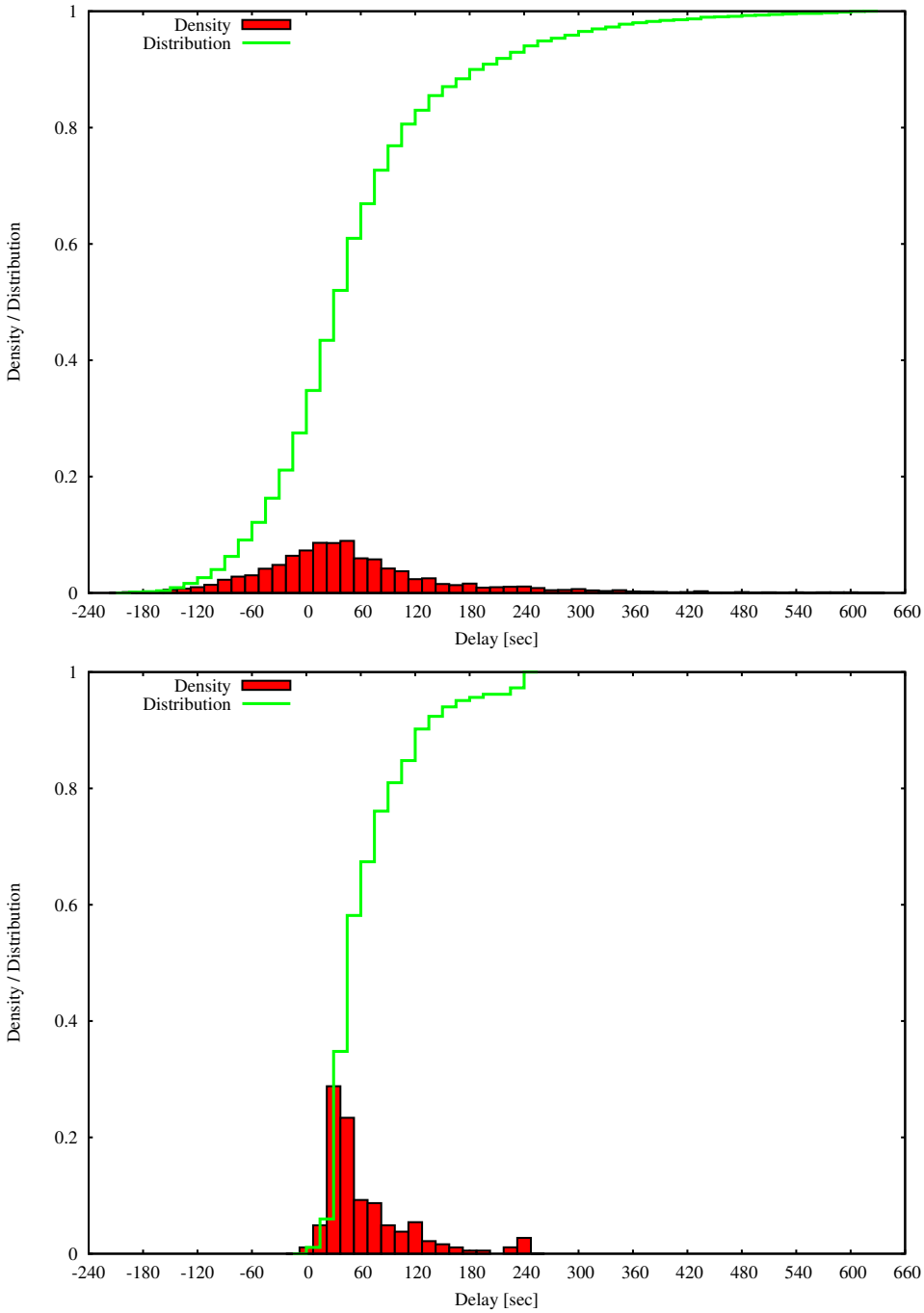

Figure 4.6: Example of arrival (top) and departure (bottom) delay distributions based on real data (source [Ullius, 2004a])

4.4.1 Weighted Exponential Distribution

Based on the work of Schwanhäusser, the weighted exponential distribution is used to model train arrival delays (see *e. g.* [Schwanhäusser, 1974], [Meng, 1991], [Kaminsky, 2001], and [Ullius, 2004b]). In contrast to the simple exponential distribution an additional parameter $0 \leq \mu \leq 1$ is introduced that indicates the share of delayed trains. This parameter ensures that exact on-time arrivals do not have probability 0, but a positive probability $1-\mu$, whenever $\mu < 1$. Hence the probability function of the exponential distribution is only changed in 0, where the mass $1-\mu$ is introduced.

Definition 4.16 (Weighted Exponential Distribution) *Let X be a random variable. Let $0 \leq \mu \leq 1$ and $\lambda > 0$. X is said to be weighted exponentially distributed, denoted as $X \sim wexp(\mu, \lambda)$, if X has the following density function (illustrated in Figure 4.7):*

$$f_X(t) = \begin{cases} 0 & t < 0 \\ (1-\mu)\delta(t) & t = 0 \\ \mu\lambda e^{-\lambda t} & t > 0 \end{cases} \tag{4.12}$$

where $\delta(t)$ is the Dirac-Delta-Function. A useful representation for $f_X(t)$—that will often be used—is the following:

$$f_X(t) = (1-\mu)\delta(t) + \mu\lambda e^{-\lambda t}\mathbf{1}_{t>0}$$

The discontinuity in 0 leads to some inconvenience when calculating convolutions of random variables that are weighted exponentially distributed. The properties of the Dirac-Delta-Function are summarized in the following lemma; proofs can be found in [Bracewell, 1999]:

Lemma 4.17 *The following properties hold for the Dirac-Delta-Function:*

$$(i)\ \ \delta(x-a) = 0 \quad \text{for all } x \neq a \tag{4.13}$$

$$(ii)\ \ \int_{-\infty}^{+\infty} \delta(x-a)dx = 1 \tag{4.14}$$

$$(iii)\ \ \int_{-\infty}^{+\infty} \delta(x-a)\,f(x)dx = f(a) \quad \text{for all continuous and bounded functions } f(x) \tag{4.15}$$

The first two properties are the defining properties, and the third is called the Sifting property.

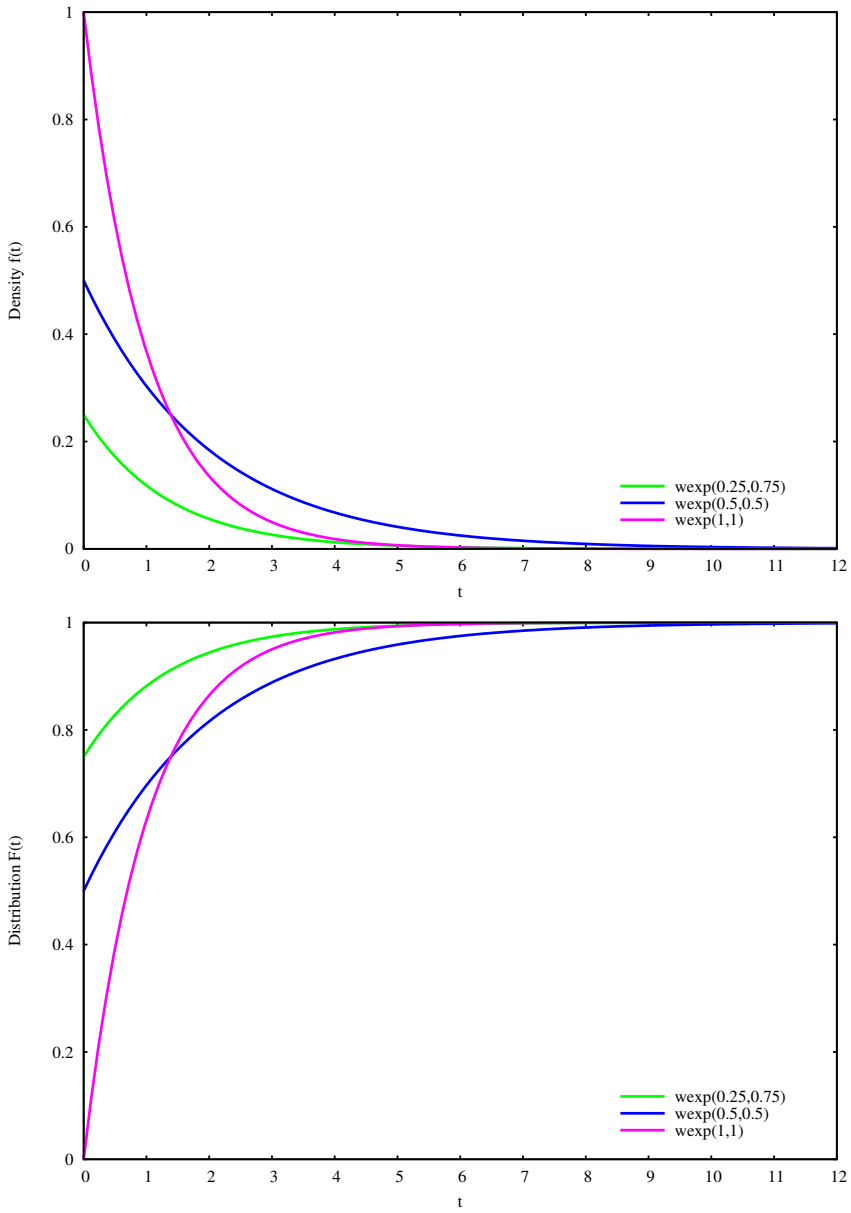

Figure 4.7: Examples of *wexp*-distributions. Three different density and their corresponding distribution functions for the weighted exponential distribution are shown. It is important to notice that the *wexp*$(1,1)$-distribution is equal to the *exp*(1)-distribution. Note that the parameter λ will be much smaller when using the *wexp*-distribution to model train delays, since such delays are more widely spread.

Lemma 4.18 *$f_X(t)$ as defined in Equation (4.12) is a density function whose corresponding distribution $F_X(a)$ satisfies (see Figure 4.7):*

$$F_X(a) = P(X \leq a) = \begin{cases} 0 & a < 0 \\ 1-\mu & a = 0 \\ 1-\mu e^{-\lambda a} & a > 0 \end{cases} \tag{4.16}$$

Proof. Five equations have to be shown:

(i) $f_X(t)$ is always greater or equal to 0, because $0 \leq \mu \leq 1$ and $\lambda > 0$

(ii) Let $a < 0$; then

$$\int_{-\infty}^{a} f_X(t)dt = 0$$

(iii) Let $a = 0$; then, since $t \leq 0$,

$$\begin{aligned} \int_{-\infty}^{a} f_X(t)dt &= \int_{-\infty}^{0} (1-\mu)\delta(t)dt \\ &= (1-\mu)\int_{-\infty}^{+\infty} \delta(t)dt \\ &= 1-\mu \end{aligned}$$

(iv) Let $a > 0$; then

$$\begin{aligned} \int_{-\infty}^{a} f_X(t)dt &= \int_{-\infty}^{a} (1-\mu)\delta(t) + \mu\lambda e^{-\lambda t}\mathbf{1}_{t>0}dt \\ &= \int_{-\infty}^{+\infty} (1-\mu)\delta(t)dt + \int_{0}^{a} \mu\lambda e^{-\lambda t}dt \\ &= (1-\mu) + \mu - \mu e^{-\lambda a} \\ &= 1-\mu e^{-\lambda a} \end{aligned}$$

(v) Let a tend to $+\infty$ in the previous equation and the last property is obtained:

$$\lim_{a\to+\infty} \int_{-\infty}^{a} f_X(t)dt = 1$$

□

The exponential part in the weighted exponential distribution satisfies the requirement for a memoryless random distribution. In fact, the exponential distribution is the only continuous memoryless random distribution. The density decreases exponentially and monotonically as t increases, and is convex (see Figure 4.7). The primary

application of the exponential distribution is the modeling of the behavior of items with a constant failure rate. Thus, reliability theory makes extensive use of the exponential distribution. Hereby the failure is the «event» and the goal is to forecast the rate or the probability of such events for a certain device. The constant failure rate of the exponential distribution requires the device to fail just as likely during the first period as during any other period. Here, failures of devices can be seen as delays of trains, hence using the exponential distribution (or the variant of a weighted exponential distribution) seems to be appropriate to model train delay distributions.

The expected value of a random variable, which obeys the *wexp*-distribution, can be calculated as follows:

Lemma 4.19 *Let $X \sim wexp(\lambda, \mu)$, where $\lambda > 0$ and $0 \leq \mu \leq 1$. Then $\mathbb{E}[X] = \frac{\mu}{\lambda}$.*

Proof. The proof is a simple calculation using Equation (4.15).

$$\begin{aligned}
\mathbb{E}[X] &= \int_{-\infty}^{+\infty} t\, f_X(t) dt \\
&= \int_{-\infty}^{+\infty} t(1-\mu)\delta(t) dt + \int_{0}^{+\infty} t\mu\lambda e^{-\lambda t} dt \\
&= (1-\mu)0 + \mu\lambda \left(0 + \frac{1}{\lambda^2}\right) \\
&= \frac{\mu}{\lambda}
\end{aligned}$$

□

4.4.2 Estimation of the Parameters

For the weighted exponential distribution, the parameters λ and μ have to be estimated. As mentioned at the beginning of this section, the Swiss Federal Railways have the quality target relating to the customers that 75% of all trains should arrive at most one minute later than scheduled and 95% of all trains should arrive with at most four minutes of delay (see also Figure 4.5). Calculating the parameters for the weighted exponential distribution implies the following equation system (with *seconds* as time unit):

$$\begin{aligned}
1 - \mu e^{-240\lambda} = 0.95 \quad &\longrightarrow \quad \mu e^{-240\lambda} = 0.05 \\
1 - \mu e^{-60\lambda} = 0.75 \quad &\longrightarrow \quad \mu e^{-60\lambda} = 0.25
\end{aligned}$$

Solving this system for μ and λ yields:

$$\begin{aligned}
\lambda &= \frac{\ln(5)}{180} = 0.00894 \\
\mu &= \frac{1}{4} 5^{\frac{1}{3}} = 0.42749
\end{aligned} \tag{4.17}$$

The values are calculated such that the *wexp*-distribution fits the quality requirements of the SBB. However, this distribution is a theoretical distribution visualizing SBB's quality goal by a distribution function.

Since real data is available (source [Ullius, 2004a]), μ and λ can be estimated from the empirical data and compared to the theoretical values. Denote with $\hat{\mu}$ and $\hat{\lambda}$ the estimated values. The estimates are calculated according to the observed delays applying the Maximum Likelihood Estimator: Denote with $x_1, \ldots, x_n$ n observations of delays. Let $x_1 = x_2 = \ldots = x_k = 0$ and $0 < x_{k+1} \leq x_{k+2} \leq \ldots \leq x_n$. The Likelihood function is then defined as:

$$L(x_1, \ldots, x_n | \mu, \lambda) = (1-\mu)^k (\mu\lambda)^{n-k} e^{-\lambda(x_{k+1}+\ldots+x_n)}$$

Looking at the logarithm of L yields

$$\begin{aligned} \ln(L) = {} & k\ln(1-\mu) + (n-k)\ln(\mu) \\ & + (n-k)\ln(\lambda) - \lambda(x_{k+1} + \ldots + x_n) \end{aligned}$$

The partial derivatives are needed in order to estimate μ and λ:

$$\frac{\partial \ln(L)}{\partial \mu} = \frac{k}{\mu - 1} + \frac{n-k}{\mu} = 0 \tag{4.18}$$

$$\frac{\partial \ln(L)}{\partial \lambda} = \frac{n-k}{\lambda} - \sum_{i=k+1}^{n} x_i = 0 \tag{4.19}$$

The two estimated values $\hat{\mu}$ and $\hat{\lambda}$ are calculated by solving the equations (4.18) and (4.19):

$$\begin{aligned} \hat{\mu} &= \frac{n-k}{n} \\ \hat{\lambda} &= \frac{n-k}{\sum_{i=k+1}^{n} x_i} \end{aligned}$$

Using empirical data of train delays (see Figure 4.6) at the Bern main station, the calculation of $\hat{\mu}$ and $\hat{\lambda}$ of the weighted exponential distribution results in the following estimated parameters for arrival and departure delay distributions:

$$\begin{aligned} \hat{\mu}_{\text{arr}} &= 0.51913 & \hat{\mu}_{\text{dep}} &= 1.0 \\ \hat{\lambda}_{\text{arr}} &= 0.00835 & \hat{\lambda}_{\text{dep}} &= 0.01329 \end{aligned} \tag{4.20}$$

$\hat{\mu}_{\text{dep}} = 1.0$ means that no train departs exactly on-time, *i. e.* precisely at the designated second. However, $\hat{\lambda}_{\text{dep}} \geq \hat{\lambda}_{\text{arr}}$ shows that the trains depart closer to the designated time than they arrive, yet the estimated mean value is still larger than one minute.

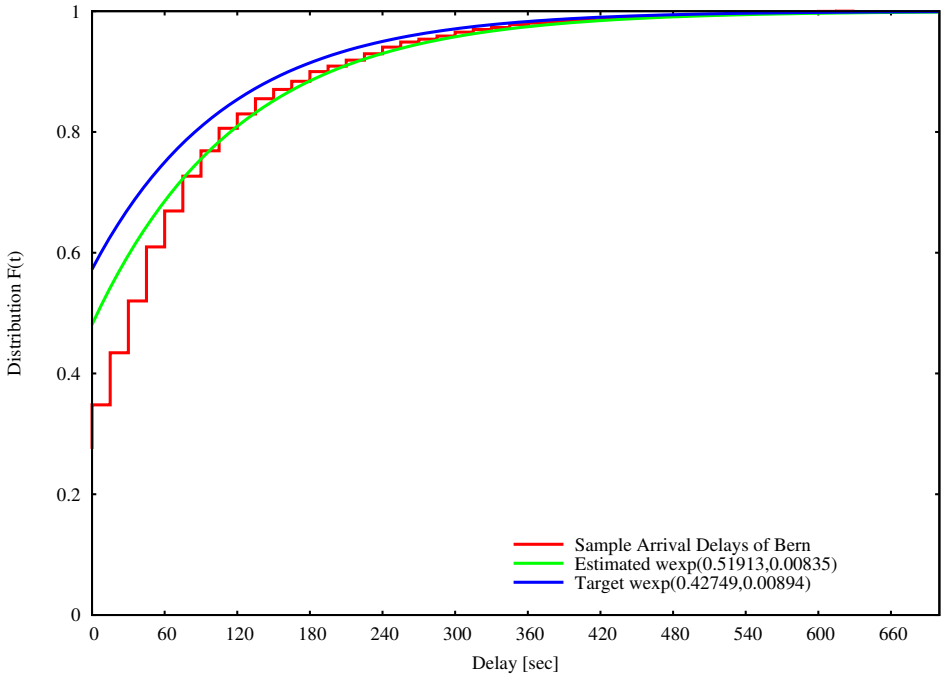

Figure 4.8: Sample, estimated and target distribution using the *wexp*-distribution with calculated and estimated parameters.

Figure 4.8 shows the plot of the arrival distribution of the empirical data, the estimated and target weighted exponential distributions. The parameters λ and μ for the target and the estimated *wexp*-distribution are taken from equations (4.17) and (4.20) respectively. The resulting distribution as well as the calculated target distribution of the SBB is shown. It seems that the weighted exponential distribution fits quite well; however, the *wexp*-distribution is too optimistic for delays within a minute. Moreover, there is a discrepancy in the number of delayed trains. In the sample data, about 35% of all trains had less than 15 seconds of delay, whereas $\hat{\mu}_{\text{arr}} = 52\%$, *i. e.* about 48% of all trains would travel on time. Using the weighted exponential distribution to model the target delay distribution of the SBB, the discrepancy is even larger as the amount of punctual trains $1 - \mu = 57\%$ shows.

4.4.3 Distribution Based on Empirical Data

The weighted exponential distribution has the drawback that negative arrival delays cannot be sampled. However, arriving trains sometimes are early and this may cause the same problems as late trains: The designated route is not necessarily open and the train has to wait until its way is free. During this waiting time some resource might be occupied blocking designated routes of other trains. Moreover, such a waiting process outside the station region causes the very time-consuming operation of starting the train up again in order to enter the station. This process might be so time-consuming that early trains might become late!

Due to the availability of empirical data (see [Ullius, 2004b]) a discrete distribution could be used to describe train delays. A bin size of 15 seconds seems to be appropriate and therefore, delays are grouped into different delay intervals of 15 seconds length. All trains having a delay between -7 and +7 seconds are considered to have zero delay. Figure 4.6 illustrates this grouping for the sample arrival data. It shows that about 10% of all train arrivals were more than one minute too early.

In contrast to the continuous distributions, random numbers of discrete distributions belong to one of the classes ..., -15, 0, +15, ... More real world data is needed in order to decrease the gap between the classes while still having the same accuracy of the estimated distribution.

4.4.4 Pulse Delay Distribution

The two previously presented delay distributions based on available data can be well used to analyze current timetables. Yet, with the possibility of communication with train drivers (see Chapter 1 and Appendix), it is possible to adjust velocities on-line in such a way that trains get «exact» delays. The idea is—like in the airline industry—to give a delayed train a new time window in which it can be handled. In that new time window it produces fewer interferences with other trains and hence the influence on

other trains is reduced. «Exact» delays are therefore assumed to be easier to handle for train dispatchers.

Since the time interval between two trains using the same resource is 90 seconds in our model for future dense timetables (see 3.3.2), it seems adequate to use multiples of 90 seconds for the table of *pulse* delays. Using again the same empirical data as previously, Table 4.1 can be deduced by grouping the data into the classes -90, 0, 90, ...

Note that by grouping the empirical arrival delay data into groups of 90 seconds, about 42% of all trains have a delay of 90 seconds! This comes from the fact that the empirical delays have been assigned to the groups by a «always round up» rule, *i. e.* delays between 1 and 90 seconds are assigned to the group «90», delays between 91 and 180 to group «180» and so on. Although this distribution seems to be inferior to the empirical arrival distribution, it will turn out that the *pulse* distribution should be preferred for future dense timetables (see Chapter 5). For departure delays the rule is different: A departure delay less than 45 seconds is assumed to be punctual (group 0), delays between 46 and 135 seconds are assigned to group 90, and so on. In Figure 4.9 the distributions are summarized for arrival and departure delays. Moreover the figure shows both distributions—the empirical and the pulse distribution.

If it is possible to decrease the probability of large delays in future, then other delay distributions can be derived as well. In order to see the impact of concentrated delay distributions on the stability of a timetable, fictitious delay distributions will be applied later (see Chapter 5.3).

Pulse Arrival

k	$P(X = k)$	$P(X \leq k)$
-90	0.063	0.063
0	0.285	0.348
90	0.421	0.769
180	0.131	0.900
270	0.041	0.941
360	0.04	0.981
450	0.007	0.988
540	0.009	0.997
630	0.004	1.000

Pulse Departure

k	$P(X = k)$	$P(X \leq k)$
0	0.582	0.582
90	0.320	0.902
180	0.060	0.962
270	0.038	1.000

Table 4.1: Estimated *pulse* delay distribution. The probabilities for the *pulse* delay distribution are calculated using the same empirical arrival and departure delay data.

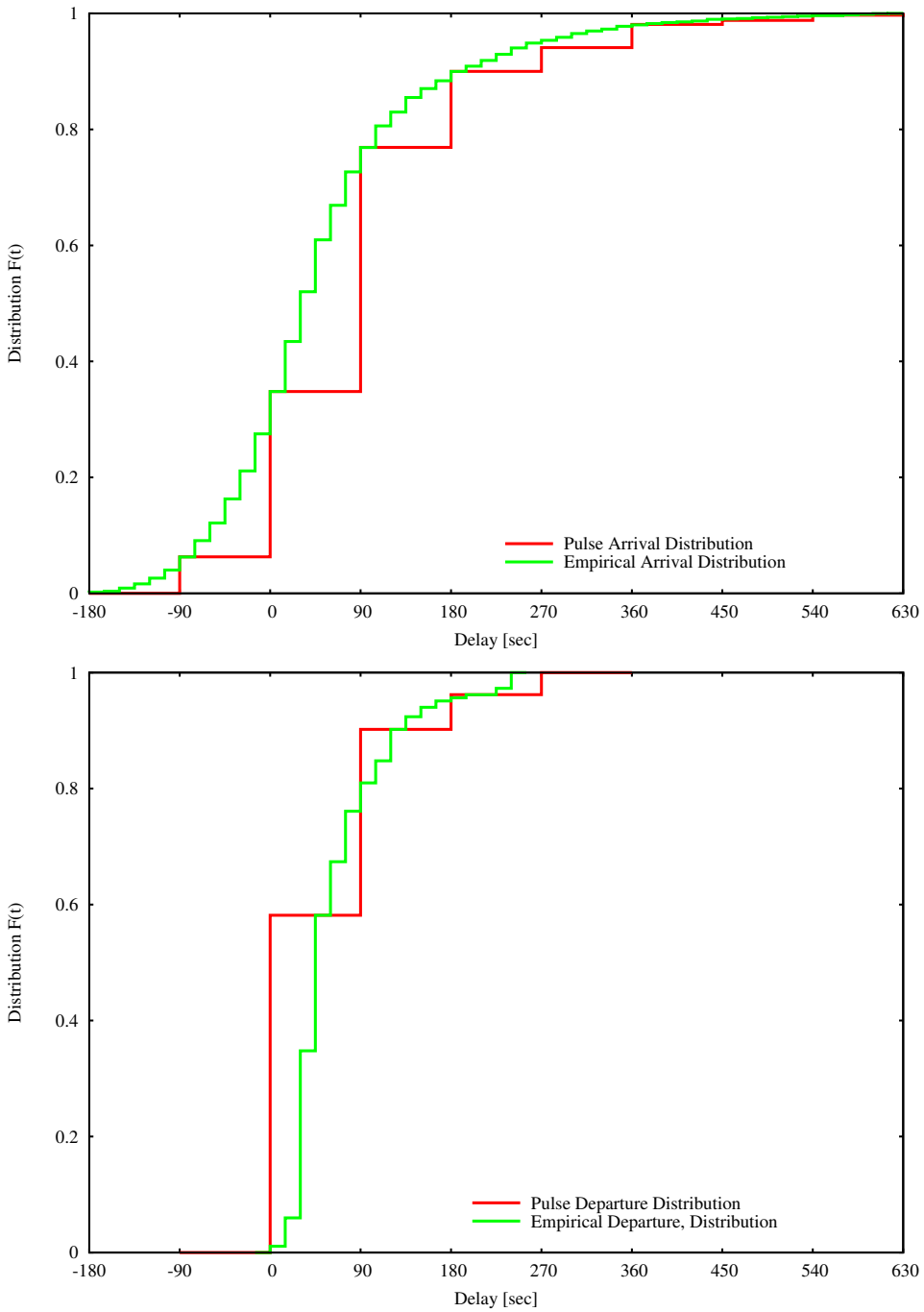

Figure 4.9: Probability distribution for the *pulse* delay distribution.

4.5 Densities of Delay Differences

In order to calculate the probabilities (4.8) and (4.9), the density of $X_i - X_k$ is needed. Depending on the distribution of X_i and X_k, this density will now be calculated by using the delay distributions introduced in Section 4.4. Define a new random variable Z as the difference between X_i and X_k:

$$Z := X_i - X_k$$

4.5.1 Weighted Exponential Distribution

Let X_i and X_k be both weighted exponentially distributed, *i. e.* let $X_i \sim wexp(\lambda_i, \mu_i)$ and $X_k \sim wexp(\lambda_k, \mu_k)$ with the following density functions

$$f_{X_i}(x) = (1-\mu_i)\delta(x) + \mu_i\lambda_i e^{-\lambda_i x}\mathbf{1}_{x>0} \qquad f_{X_k}(x) = (1-\mu_k)\delta(x) + \mu_k\lambda_k e^{-\lambda_k x}\mathbf{1}_{x>0}$$

Since the density of $X_i - X_k$ has to be calculated, the density of $-X_k$ is of interest:

$$f_{-X_k}(x) = (1-\mu_k)\delta(x) + \mu_k\lambda_k e^{\lambda_k x}\mathbf{1}_{x<0}$$

Lemma 4.20 *The density $f_Z(z)$ of the random variable $Z = X_i - X_k$ is as follows:*

$$f_Z(z) = \begin{cases} \frac{\mu_i\mu_k\lambda_i\lambda_k}{\lambda_i+\lambda_k} e^{\lambda_k z} + (1-\mu_i)\mu_k\lambda_k e^{\lambda_k z} & \text{for } z < 0 \\ (1-\mu_i)(1-\mu_k)\delta(z) & \text{for } z = 0 \\ \frac{\mu_i\mu_k\lambda_i\lambda_k}{\lambda_i+\lambda_k} e^{-\lambda_i z} + \mu_i(1-\mu_k)\lambda_i e^{-\lambda_i z} & \text{for } z > 0 \end{cases} \tag{4.21}$$

Proof. The density of $f_Z(z)$ is the convolution of f_{X_i} and f_{-X_k} and can be written as

$$f_Z(z) = (f_{X_i} * f_{-X_k})(z) = \int_{-\infty}^{+\infty} f_{X_i}(x)\, f_{-X_k}(z-x)dx$$

Using the compact formulation of f_{X_i} and f_{-X_k} yields the product:

$$\begin{aligned} f_{X_i}(x)\, f_{-X_k}(z-x) &= \Big(\delta(x)(1-\mu_i) + \mu_i\lambda_i e^{-\lambda_i x}\mathbf{1}_{x>0}\Big) \\ &\quad \Big(\delta(z-x)(1-\mu_k) + \mu_k\lambda_k e^{\lambda_k(z-x)}\mathbf{1}_{z-x<0}\Big) \\ &= \delta(x)\delta(z-x)(1-\mu_i)(1-\mu_k) && (4.22) \\ &\quad + \delta(x)(1-\mu_i)\mu_k\lambda_k e^{\lambda_k(z-x)}\mathbf{1}_{z-x<0} && (4.23) \\ &\quad + \delta(z-x)(1-\mu_k)\mu_i\lambda_i e^{-\lambda_i x}\mathbf{1}_{x>0} && (4.24) \\ &\quad + \mu_i\mu_k\lambda_i\lambda_k e^{-(\lambda_i+\lambda_k)x+\lambda_k z}\mathbf{1}_{x>0\wedge x>z} && (4.25) \end{aligned}$$

The four parts (4.22)-(4.25) are now integrated separately.

(i) The integral of (4.22) is calculated as follows:

$$\int_{-\infty}^{+\infty} \delta(x)\delta(z-x)(1-\mu_i)(1-\mu_k)dx = (1-\mu_i)(1-\mu_k)\int_{-\infty}^{+\infty} \delta(x)\delta(z-x)dx$$

Since $(1-\mu_i)(1-\mu_k)$ only contributes to f_Z if $z=0$, this immediately results in

$$= (1-\mu_i)(1-\mu_k)\delta(z)$$

(ii) The integral of (4.23) is

$$\int_{-\infty}^{+\infty} \delta(x)(1-\mu_i)\mu_k\lambda_k e^{\lambda_k(z-x)}\mathbf{1}_{z-x<0}dx$$
$$= (1-\mu_i)\mu_k\lambda_k\int_z^{+\infty} \delta(x)e^{\lambda_k(z-x)}dx$$

If $z>0$ then its integral is 0 since $\delta(x)=0$ for all $x\neq 0$. Hence only the case $z\leq 0$ is interesting. Using once more the Sifting property we get

$$= (1-\mu_i)\mu_k\lambda_k e^{\lambda_k z} \qquad \text{for all } z\leq 0$$

(iii) The third integral (4.24) is calculated in the same way:

$$\int_{-\infty}^{+\infty} \delta(z-x)(1-\mu_k)\mu_i\lambda_i e^{-\lambda_i x}\mathbf{1}_{x>0}dx$$
$$= (1-\mu_k)\mu_i\lambda_i\int_0^{+\infty} \delta(z-x)e^{-\lambda_i x}dx$$

If $z<0$ then $z-x<0$ since $x>0$. Hence the Dirac function is always 0 in this case. If $z\geq 0$ then this equals to

$$= \mu_i(1-\mu_k)\lambda_i e^{-\lambda_i z} \qquad \text{for all } z\geq 0$$

(iv) The last part, integration of equation (4.25), is:

$$\int_{-\infty}^{+\infty} \mu_i\mu_k\lambda_i\lambda_k e^{-(\lambda_i+\lambda_k)x+\lambda_k z}\mathbf{1}_{x>0\wedge x>z}dx$$

Two cases have to be distinguished, either $z>0$ or $z\leq 0$. For $z>0$ the integral is

$$= \mu_i\mu_k\lambda_i\lambda_k e^{\lambda_k z}\int_z^{+\infty} e^{-(\lambda_i+\lambda_k)x}dx$$
$$= \frac{\mu_i\mu_k\lambda_i\lambda_k}{\lambda_i+\lambda_k}e^{-\lambda_i z} \qquad \text{for } z>0$$

If $z \leq 0$, then the integral equals to

$$\begin{aligned} &= \mu_i \mu_k \lambda_i \lambda_k e^{\lambda_k z} \int_0^{+\infty} e^{-(\lambda_i+\lambda_k)x} dx \\ &= \frac{\mu_i \mu_k \lambda_i \lambda_k}{\lambda_i + \lambda_k} e^{\lambda_k z} \qquad \text{for } z \leq 0 \end{aligned}$$

The results of (i)–(iv) are combined resulting in the density function $f_Z(z)$.

$$f_Z(z) = \begin{cases} \frac{\mu_i \mu_k \lambda_i \lambda_k}{\lambda_i+\lambda_k} e^{\lambda_k z} + (1-\mu_i)\mu_k \lambda_k e^{\lambda_k z} & \text{for } z < 0 \\ (1-\mu_i)(1-\mu_k)\delta(z) & \text{for } z = 0 \\ \frac{\mu_i \mu_k \lambda_i \lambda_k}{\lambda_i+\lambda_k} e^{-\lambda_i z} + \mu_i(1-\mu_k)\lambda_i e^{-\lambda_i z} & \text{for } z > 0 \end{cases}$$

□

$f_Z(z)$ is a density function indeed. This can be seen by calculating the integral $\int_{-\infty}^{+\infty} f_Z(z)dz$:

$$\begin{aligned} \int_{-\infty}^{+\infty} f_Z(z)dz = &\int_{-\infty}^{0} \frac{\mu_i \mu_k \lambda_i \lambda_k}{\lambda_i + \lambda_k} e^{\lambda_k z} + (1-\mu_i)\mu_k \lambda_k e^{\lambda_k z} dz \\ &+ \int_0^{+\infty} \frac{\mu_i \mu_k \lambda_i \lambda_k}{\lambda_i + \lambda_k} e^{-\lambda_i z} + \mu_i(1-\mu_k)\lambda_i e^{-\lambda_i z} dz \\ &+ \int_{-\infty}^{+\infty} (1-\mu_i)(1-\mu_k)\delta(z)dz \\ = &\frac{\mu_i \mu_k \lambda_i + (\lambda_i + \lambda_k)(1-\mu_i)\mu_k}{\lambda_i + \lambda_k} \\ &+ \frac{(\lambda_i + \lambda_k)\mu_i(1-\mu_k) + \mu_i \mu_k \lambda_k}{\lambda_i + \lambda_k} \\ &+ (1-\mu_i)(1-\mu_k) \\ = &1 \end{aligned}$$

The density function $f_Z(z)$, as well as the corresponding probability distribution is illustrated in Figure 4.10.

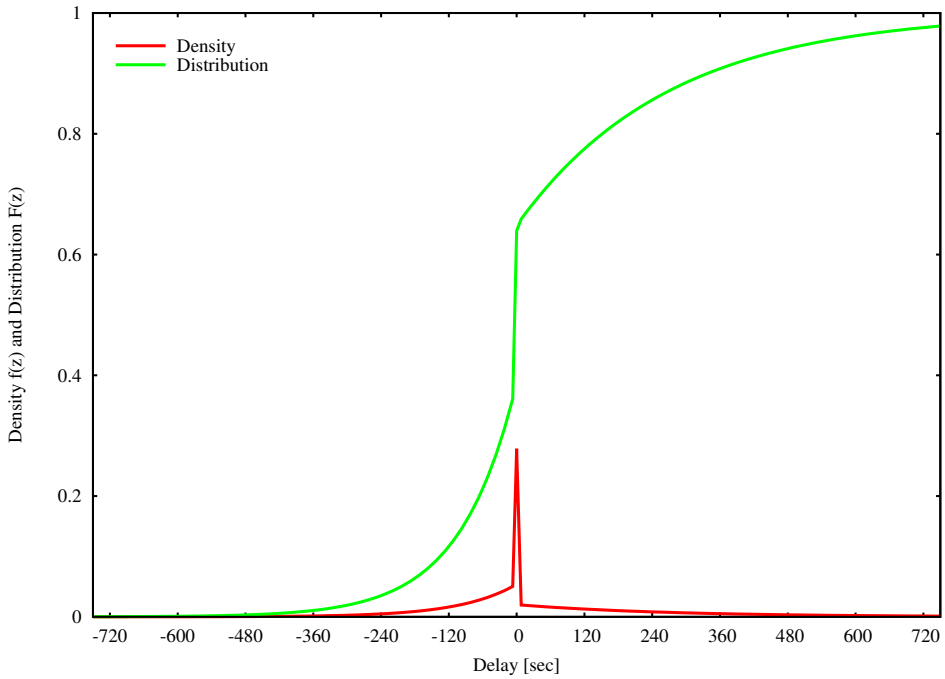

Figure 4.10: The density function $f_Z(z)$ and the corresponding probability distribution $F_Z(z)$ for $Z = X_i - X_k$, X_i and X_k both *wexp*-distributed. The parameters used are $\mu_i = 0.42$, $\mu_k = 0.52$, $\lambda_i = 0.00372$, and $\lambda_k = 0.01$. Most of the delay differences are between -5 and +5 minutes.

4.5.2 Discrete Distributions

The calculation of interest is the joint probability of the difference of two discrete delay distributions. Let X_i and X_k be random variables describing the delay of a train using any discrete distributions. The probabilities are given by $P(X_i = p) = x_p^i$ with $\sum_p x_p^i = 1$ and $P(X_k = q) = x_q^k$ with $\sum_q x_q^k = 1$. Consider the random variable $Z = X_i - X_k$. The distribution of Z is then defined as the discrete convolution:

$$P(Z = z) = \sum_q P(X_i = q) P(X_k = q - z)$$

The distribution of Z looks slightly different depending on the distribution of X_i and X_k. As an example, in Table 4.2 the convoluted distribution is shown, using the *pulse* delay distribution for arrivals and assuming that both random variables are identically distributed. In Figure 4.11 the corresponding probability distribution of Z is illustrated.

z	$P(Z = z)$	$P(Z \leq z)$
-720	0.0003	0.0003
-630	0.0017	0.002
-540	0.0047	0.0067
-450	0.0088	0.0155
-360	0.0183	0.0338
-270	0.0382	0.072
-180	0.087	0.159
-90	0.2005	0.3595
0	0.281	0.6405
90	0.2005	0.841
180	0.087	0.928
270	0.0382	0.9662
360	0.0183	0.9845
450	0.0088	0.9933
540	0.0047	0.998
630	0.0017	0.9997
720	0.0003	1

Table 4.2: Convoluted probability distribution using *pulse* arrival delay distributions. Probabilities for the distribution $Z = X_i - X_k$ in case of *pulse* arrival delay distributions for both random variables X_i and X_k.

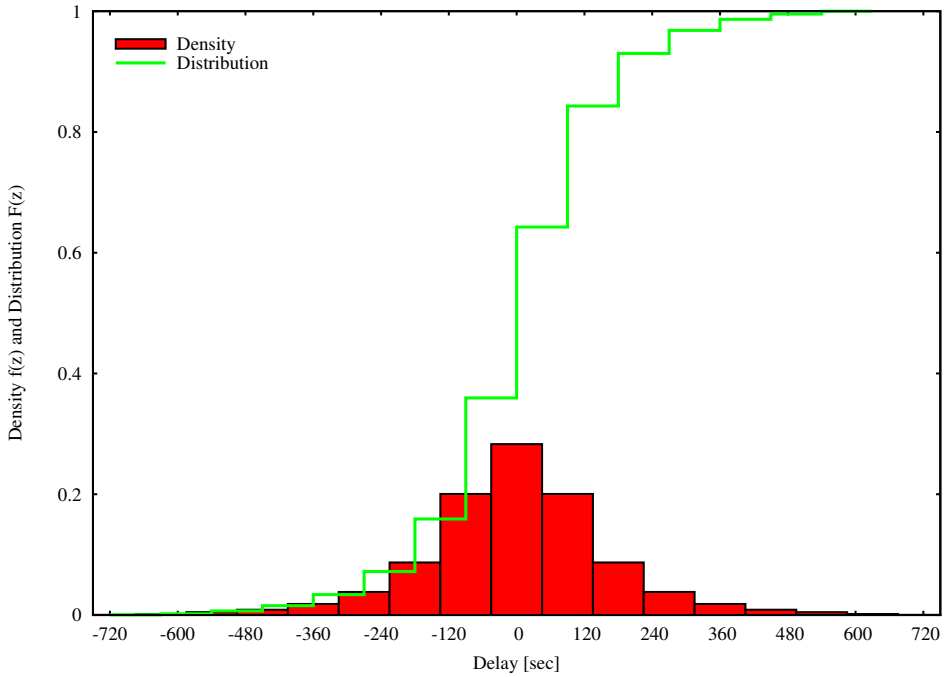

Figure 4.11: Probability distribution of $Z = X_i - X_k$ in the case of *pulse* arrival delay distributions.

Chapter 5

Stable Train Routings

You may never know what results come of your action, but if you do nothing there will be no result.
Mahatma Gandhi (1869–1948)

In the previous chapter, probabilistic and deterministic measures for the evaluation of train schedules were established in order for them to be evaluated and analyzed from different points of view. The aim in this chapter now is to find routings that are not only conflict-free but that will also remain conflict-free for perturbations as large as possible. As mentioned in Chapter 2, the resulting schedules should be as close to the efficient frontier as possible. Since the timetable and thus the period are fixed, the chosen routes must thus not interfere with each other and trains should be assigned to routes such that the routes are as «far apart» from each other as possible. The stability measures introduced in Chapter 4 examine relationships among the chosen routes and indicate the sensitivity of routings against (small) deviations of the timetable.

So far these stability measures have not been used to construct operable schedules from draft timetables. Therefore, given a draft timetable, the goal is to find routings that satisfy safety requirements and at the same time optimize a target function incorporating stability aspects. The stability measures of Chapter 4 are taken into account here in order to formulate four different optimization problems. Solving these problems for the same instance will lead to slightly different optimized schedules and each optimized schedule will have its own characteristics depending on the assumed train delay distribution.

Note that the optimal schedules are not automatically operable; they still have to satisfy certain minimum requirements (see Figure 5.1 and Chapter 2). The goal for

future (dense) schedules is stated relative to today's schedules: «Higher traffic volume maintaining at least same level of stability». Therefore current timetables act as reference schedules. Future schedules should have the same stability characteristics as current schedules. Hence, the threshold values introduced in Definitions 4.4, 4.7, 4.10, and 4.12 are determined by the stability values of today's schedules. Although requirements for operable schedules additionally include operational and managerial aspects such as reaction times of dispatchers or communication and information flows, they are not addressed here.

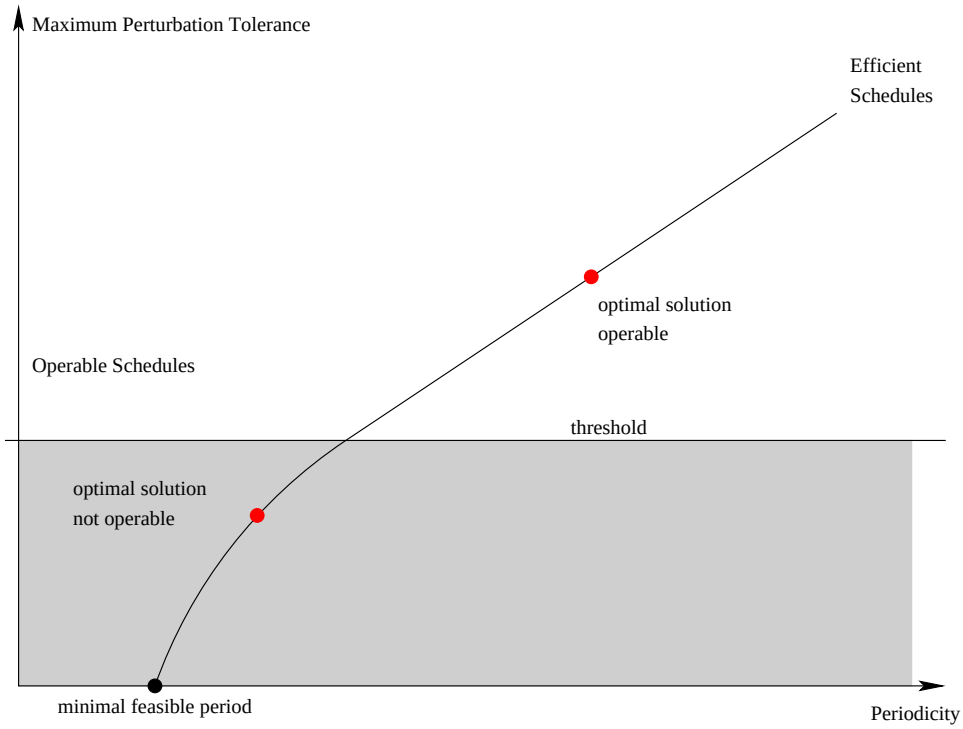

Figure 5.1: Operable schedules. Schedules satisfying some stability requirements are operable, others not.

The chapter is structured as follows. In the first section the optimization problems and a basic algorithm are presented. The second section shows variants of the basic algorithm where the different objective functions are taken into account. The chapter is completed by extensive comparisons of the results of different optimization methods for the Bern test case.

5.1 The Timetable Stabilization Problems

It is essential that the schedule is stable—from the railroad company's as well as from the customer's point of view. In Definitions 4.4, 4.7, 4.10, and 4.12 a timetable is called stable, if a desired lower or upper threshold is not exceeded by the corresponding stability indicator value. Each of the four stability measures examines the distances between the trains—each one having a different focus. Let x_{ij} denote whether route j of train i belongs to the solution ($x_{ij} = 1$) or not ($x_{ij} = 0$). Then the four objective functions can be formulated as follows:

Deterministic Stability—Minimal Time Slots: The time slot measures the time distance between two trains. The corresponding objective function maximizes the minimum time slot length, or in a more general setting, the k smallest time slots are taken into account. For the first objective function $\mathscr{F}_1$, $\omega_1 \geq \cdots \geq \omega_n \geq 0$ are weights and $\sigma(s)$ is a permutation such that the length of the time slots are brought into ascending order, *i. e.* $\left|F_{\sigma(1)}(x_{11},\ldots x_{nm_n})\right| \leq \cdots \leq \left|F_{\sigma(n)}(x_{11},\ldots x_{nm_n})\right|$. Therefore, the shortest time slot receives the largest weight in the target function. By using $\omega_1 = 1$ and $\omega_i = 0$ for all $i \neq 1$ one focuses solely on the train having the shortest time slot. The weights are introduced in order to obtain better results, *i. e.* solutions in which not only the minimal time slot is considered but the k shortest with emphasis on the smallest. In the test runs, the weights will be set to $\omega_1 = 4/10$, $\omega_2 = 3/10$, $\omega_3 = 2/10$, $\omega_4 = 1/10$, and $\omega_i = 0$ for $i \geq 5$. In order to formulate the timetable optimization problems uniformly, the function $\mathscr{F}_1$ is defined here as:

$$\mathscr{F}_1 := -\sum_{s=1}^{n} \omega_s \left|F_{\sigma(s)}(x_{11},\ldots,x_{nm_n})\right| \tag{5.1}$$

Conflict Stability—Maximum Clique Weight: The expected number of conflicts corresponds to the sum of all weights of the clique induced by the routing, *i. e.* by the variables x_{ij}. Hence, the second target function focussing on the conflict stability by considering the clique weight is defined as:

$$\mathscr{F}_2 := \sum_{i,j,k,l} x_{ij}x_{kl}P\left(r_{ij} \nleftrightarrow r_{kl}\right) \tag{5.2}$$

Structural Stability—Maximum Edge Weight: The structural stability focuses on the maximum edge weight in the clique induced by x_{ij}. Moreover, the structural stability is used to calculate a lower bound on the schedule failure probability and the target function is formulated as:

$$\mathscr{F}_3 := \max_{i,j,k,l} x_{ij}x_{kl}P\left(r_{ij} \nleftrightarrow r_{kl}\right) \tag{5.3}$$

Cluster Stability—Maximum Node Weight: The expected number of conflicts should not be concentrated in a small number of trains. The sum of the weights of the edges incident to a node build the expected number of conflicts for the train corresponding to this node. Thus, the objective function expressing the maximum node weight is:

$$\mathscr{F}_4 := \max_{i,j} x_{ij} \sum_{kl,k\neq i} x_{kl} P\left(r_{ij} \nleftrightarrow r_{kl}\right) \tag{5.4}$$

Any of the four stability measures could act as an objective function. In contrast to Problem 3.16 (Equations (3.2)–(3.5)), the objective function is replaced by either of the four objective functions $\mathscr{F}_1$, $\mathscr{F}_2$, $\mathscr{F}_3$ or $\mathscr{F}_4$. Yet, the constraints remain the same and the four timetable stabilization optimization problems are:

Problem 5.1 (Timetable Stabilization Problem)

$$\min \quad \mathscr{F}_i \tag{5.5}$$

$$s.t. \quad \sum_{j=1}^{m_i} x_{ij} = 1 \qquad \textit{for all } i = 1,\ldots,n \tag{5.6}$$

$$x_{ij} + x_{kl} \leq 1 \qquad \textit{for all } r_{i_j} \nleftrightarrow r_{k_l} \tag{5.7}$$

$$x_{ij} \in \{0,1\} \tag{5.8}$$

As in the model described in Chapter 3, the timetable stabilization problem still consists of finding a compatible set of vertices in a conflict graph. Since the underlying problem is the decisive problem of finding an independent set, all four optimization problems belong to the class $\mathscr{NPO}$ as long as the layout of the station is not fixed. Hence, heuristics depending on $\mathscr{F}_i$ will be used to solve the optimization problem.

Remark 5.2 *Whereas simple time slots are used as an objective function, extended time slots are not considered as an optimization function here, because computationally, it takes too long to evaluate a schedule many times with the length of extended time slots as the objective function (see Remark 4.6 about the time complexity to compute extended time slots). Yet, the proposed heuristics require many evaluations of the objective function in order to solve the optimization problem. Extended time slots therefore act as an analysis tool of optimized schedules only in order to complete the picture about timetable stability. See also remarks about time slots in Section 5.2.1.*

As it will be shown, the computational costs to evaluate a solution are important for an algorithm that solves Problem 5.1. While in the fixed point iteration method (see Section 3.5) the focus has been on the question of feasibility, the optimization task is

addressed here. Generally, the weakness of $\mathscr{F}_2$ through $\mathscr{F}_4$ is the construction of the extended conflict graph, especially its weights, whereas for $\mathscr{F}_1$ no such preprocessing has to be made. However, the strength of $\mathscr{F}_2$ through $\mathscr{F}_4$ is that a single evaluation of a solution is computationally cheap, whereas for evaluating $\mathscr{F}_1$ a non-trivial algorithm has to be executed (see Algorithm 4.1). Nevertheless, the question of which objective function yields the most stable schedules is interesting.

5.2 Random-Restart Local Search Method

The fixed point iteration algorithm (Algorithm 3.2) «solves» the feasibility problem in the sense that for any instance, it detects whether the instance is feasible or whether it is believed to be infeasible. Henceforth, it is thus assumed that the problem instance is feasible and an initial schedule L is available. Recall that $L = \{x_{11}, \ldots, x_{nm_n} \in \{0,1\}\}$ is an assignment of routes to trains respecting the Constraints (5.6)–(5.8) of Problem 5.1. Moreover, recall that L is also used to denote a set of nodes in the extended conflict graph.

A large spectrum of optimization algorithms is available, each one running best if some conditions are satisfied. The timetable stabilization problems consist of finding a clique of n nodes in a graph, in such a way that the clique does not contain an edge with weight 1. As the problem instance is assumed to be feasible, the maximum number of nodes in the independent set is known. A better solution (with respect to the chosen target function) can only be achieved by replacing a set of nodes by another, equivalently large set of nodes respecting the constraints (5.6)–(5.8). However, finding a large, feasible replacement set is difficult as a large independent set problem must be solved again.

Assume that two different node sets (*i. e.* schedules) L and L'—*i. e.* node sets—for the same instance are available. According to the functions $\mathscr{F}_1$ through $\mathscr{F}_4$, L or L' is to be preferred. However, the solution space, *i. e.* the number of different Ls can be tremendous (see Chapter 3) and hence, it is futile to enumerate all possible solutions and pick the best. Even for the densest test case instances, many different solutions could be found by the fixed point iteration method (see Algorithm 3.2).

Thus, in order to solve Problem 5.1, the following algorithm is introduced describing a local search heuristic to find a favorable solution L: It is a general algorithm for which several variants will be discussed later depending on the objective function. In Chapter 3 the fixed point iteration method has been applied to check the feasibility of the instances. As already discussed there, a simple random search technique to find feasible node sets would again fail here, if no additional information (such as well founded probabilities for the nodes) is used. For the same reason, greedy techniques like Greedy Randomized Adaptive Search Procedure (GRASP) (see [Blum and Roli, 2003] and [Feo and Resende, 1995]) used to solve the optimiza-

tion problem fail as well, as does an extension of the fixed point iteration scheme, where weights are used (see [Caimi, 2004]). Due to the availability of an initial solution and due to the fact that many initial start solutions can be generated with the fixed point iteration method, a local search heuristic is applied to solve the timetable stabilization problems (see Algorithm 5.1).

Algorithm 5.1 Local Search Heuristic to solve the Timetable Stabilization Problem

Input: A (extended) conflict graph, a neighborhood-size k, a feasible solution L, and an objective function $\mathscr{F}$
Output: A k-optimal solution L^* (according to $\mathscr{F}$)

```
BestSchedule=L
while restart_algorithm = TRUE do
    threshold := objective_value(L)      // Evaluate current solution
    try to substitute L by L' according to a neighborhood N_k and threshold
    if trying not successful then
        restart_algorithm = FALSE
    else
        if necessary, change BestSchedule to L'
    endif
endwhile
return L* := BestSchedule
```

Local search is a common method to solve computationally difficult problems. It is usually applied if the space of candidate solutions of a given problem instance is very large (see *e. g.* [Blum and Roli, 2003]). The basic principle is to start from an initial solution (*i. e.* an assignment L) and then to choose another candidate solution from its direct neighborhood, and hence a neighborhood relation on this space has to be defined. The evaluation of each move is based on the objective function $\mathscr{F}$ and the moving procedure stops when a termination condition is met. Commonly, two methods are used to make moves in the neighborhood: Only improvements are allowed (greedy approach) or worsenings are also accepted, however in fact only under certain conditions (simulated annealing or tabu search). In the latter case, the iterations are typically repeated until a given computational budget (*e. g.* time or maximum number of steps) has been exhausted. In the former case, the termination criterion is met if a k-optimal solution (see below) has been found. Note that if k is chosen equal to n (the number of trains) then Algorithm 5.1 finds the optimal solution. Yet, k usually is a small number depending on the problem size—*e. g.* $k \leq 4$.

Random-restart local search simply runs an outer loop over the local search method.

Each step of the outer loop chooses a random initial solution L to start the local search procedure. The best solution L^* is kept; if a new run produces a better L^*, it replaces the old L^*.

Random-restart local search is a surprisingly effective algorithm for these kind of instances. It is a state-of-the-art technique to solve combinatorial problems, yet in order to meet our needs the algorithm will be fine-tuned. It has already been discussed for the fixed point iteration method, yet it turns out that here too it is better to spend CPU time exploring the initial solution space, rather than carefully optimizing from the initial solution.

In order to solve the timetable stabilization problem by the random-restart local search heuristic with any of the objective functions $\mathscr{F}_1$, $\mathscr{F}_2$, $\mathscr{F}_3$, and $\mathscr{F}_4$, solution neighborhoods, termination conditions and the type of local search have to be discussed.

5.2.1 Evaluation of the Objective Function

For every possible solution L during the local search procedure, the objective value has to be determined once by applying one of the four objective functions $\mathscr{F}_1$, $\mathscr{F}_2$, $\mathscr{F}_3$, or $\mathscr{F}_4$. As a result not only the objective value is determined but the *critical node* according to $\mathscr{F}_i$ is determined as well. A node is considered to be *critical* if it contributes most to the objective value; *e. g.* the train having the smallest time slot. In fact, not only the element, which contributes most to the objective function, but an ordered list is created according to the objective function $\mathscr{F}_i$, when a schedule L is evaluated. This list can henceforth be used in order to build the neighborhood solution and to decide which routes should be replaced (see also Section 5.2.2). The evaluation of a solution L is executed differently, depending on the objective function $\mathscr{F}_i$.

$\mathscr{F}_1$ **– Time Slot Maximization:** A solution L is evaluated by determining the time slot for each train according to Algorithm 4.1. The objective value is a weighted sum of the r smallest time slots. After the time slots for each train are known, the candidate nodes in the conflict graph can be ordered by their time slot size. The first element in this list is then the critical node because it belongs to the train having the minimal time slot. The routing combinations prohibit this train from having a broader time slot and hence its route should be exchanged. If no improvement to the solution (*i. e.* a wider minimal time slot) can be found for this train, then, according to the ordered list, routes of other trains will be probed to improve the solution.

When using the extended time slots as the objective function, the evaluation would take too long to construct this list (or a part of it) since time slots have to be calculated individually for every possible route for each train (see Remarks 4.6 and 5.2).

$\mathscr{F}_4$ **– Critical Train Influence Minimization:** Using the extended conflict graph model, for each node contained in solution L, the sum of the weights of the incident edges is calculated. The train corresponding to the node whose weight is maximal defines the critical node v_{crit}. In order to reduce the influence of this train, v_{crit} should be replaced and—if possible—a better route should be chosen instead. If no such replacement is possible, then the node with the second highest sum is probed, and so on.

$\mathscr{F}_2$ **– Expected Number of Conflicts Minimization:** As for the critical train influence minimization, the induced clique in the extended graph model is evaluated. The sum of all the weights of the edges belonging to the clique induced by L provides the expected number of conflicts. In order to decrease this number, the same critical node concept as for the train influence minimization can be used. Replacing the heaviest node v_{crit} according to the sum of the weights of the incident edges potentially decreases the sum of the weights in the clique.

$\mathscr{F}_3$ **– Weakest Link Minimization:** In contrast to the three previously mentioned evaluation methods, this function does not look at weights of nodes, but at weights of edges. The edge having the maximum weight in the clique induced by a solution L defines the weakest link in the schedule. In order to improve the weakest link, the maximum edge weight $e_{\max} = (u, v)$ is decreased by replacing either of the two incident nodes u or v or both at the same time. The order here is not clear: There is no evident clue indicating which node contributes more to the evaluation function. Therefore both u and v are considered to be equal when using the evaluation function $\mathscr{F}_3$.

5.2.2 The Solution Neighborhood

Closely related to the evaluation of a solution is the neighbor selection method. The construction of an ordered list of nodes during the evaluation of a solution L is crucial in order to find neighborhood solutions. Depending on the objective function, the neighborhood is differently structured. For the time slot optimization problem, it is not possible to introduce the length of the time slots as edge or node weights in the conflict graph since the time slot of a routing cannot be calculated before the complete routing has been made. As a consequence, the problem of maximizing time slots of routings cannot be directly modeled neither into the conflict graph nor in the extended conflict graph. Contrary to the time slot model, additional edges and edge weights have been introduced to the conflict graph model in order to obtain an extended conflict graph and to evaluate a solution directly by the evaluation functions $\mathscr{F}_2$, $\mathscr{F}_3$, and $\mathscr{F}_4$.

A solution L consists of a set of nodes and hence neighborhoods are defined over node sets. In Algorithm 5.1 a neighborhood N_k of nodes is created. In a k-neighborhood, at most k routes of k different trains are replaced, *i. e.* up to k nodes

are exchanged. The replacement routes have to be compatible to the $n-k$ routes that are not changed. The solution neighborhood can be defined for all four objective functions as follows:

Definition 5.3 (Solution Neighborhood) *Let L be a feasible solution, i. e. a node set respecting Constraints (5.6)–(5.8) of Problem 5.1, and let $k \in \mathbb{N}$ be a positive integer. Then the k-Neighborhood is*

$$N_k(L) := \left\{ L' \,\middle|\, L' \text{ is a feasible solution and } \left|L' \setminus L\right| \leq k \right\}$$

The nodes are substituted by other nodes according to $N_k(L)$. The set $N_k(L)$ is ordered according to the objective function $\mathscr{F}_i$ (see previous Section 5.2.1). The neighboring solutions L' are ordered according to the possible improvements of the current objective value. In the corresponding graph model, a node is only replaced by a node of the same train. This replacement step corresponds to assigning the corresponding train to another route and thus respecting Condition (5.6).

However, two problems with the definition of N_k occur in practice. First, the exploration of the neighborhood is not trivial, since only feasible solutions can be accepted to be in the set N_k. For k small (*e. g.* $k = 1$ or $k = 2$) a simple enumeration of all possibilities defines the set N_k. However, for larger k this is no longer practical because the computation time to determine N_k is tremendous. Secondly, the size of N_k could also be very large due to the number of different routing possibilities.

Therefore, not the entire neighborhood N_k is enumerated, but usually only a part of it is explored. The first step consists of choosing the nodes that should be replaced. According to the evaluation function (see previous paragraph on the evaluation functions), the k most critical nodes are determined and subsequently deleted from L. The resulting partial solution in which a set U of $n-k$ trains have fixed routes and k trains have no assigned route is called a *subsolution*, denoted by L_U. Denote by U^c the set of nodes whose corresponding k trains are not assigned to a route yet. Subsolutions always respect restrictions (5.7) and (5.8), yet not necessarily (5.6).

A new solution L' is constructively built by adding feasible routes from U^c to the subsolution L_U, whenever possible. If a subsolution cannot be extended to a fully feasible solution different from L, then the subsolution is infeasible and new subsolutions or neighborhoods have to be analyzed. In order to evaluate the objective value of a subsolution L_U, it is not necessary that all trains already have a route—those having no route ($x_{ij} = 0$ for all $j = 1, \ldots m_i$) are neglected in the evaluation of $\mathscr{F}_i(L_U)$.

If $\mathscr{F}_i = \mathscr{F}_1$ then unassigned trains get an artificial time slot γ that is chosen appropriately. A good choice for γ is the timetable period T since time slots cannot be larger than T. Hence, those trains do not have any influence on the objective function, *i. e.* the minimal time slot. If $\mathscr{F}_i$ is any of the other evaluation functions, then undecided nodes and edges get the artificial weight 0. Consequently, in the calculations they are neglected and only the partial clique induced by the subsolution L_U is

considered for the evaluations. Therefore, these artificial values do not change the critical elements and hence the corresponding nodes are (almost) neglected for the determination of the neighborhood $N_k(L)$ as their place is at the end of the ordered list.

Moreover, the target function value of a subsolution is always smaller or equal to the value of any of its completions. This is evident because introducing another train may result in worsening the objective value and never in an improvement. This implies that the objective value will not increase by successively constructing a solution by adding trains. Hence, a subsolution always gives a lower bound on the objective value of its extensions. This property is used when the objective value of the subsolution is compared with the value of the previous best solution. Extensions of the subsolution need only be considered if the subsolution has a smaller objective value than the current best solution. As a consequence, only a fraction of the neighborhood has to be examined. This leads to the definition of *k*-optimal solutions:

Definition 5.4 (k-Optimal Solution) *L is a k-optimal solution if*

$$\mathscr{F}_i(L) \leq \mathscr{F}_i(L') \text{ for all } L' \in N_k(L)$$

The random-restart local search technique finds *k*-optimal solutions to Problem 5.1 or stops when its computational budget has been exhausted. Furthermore, it uses the ordering of $N_k(L)$ in order to substitute the nodes according to their contribution to the objective function value.

Remark 5.5 *Note that due to the same choice of the neighborhood, the random-restart local search technique used to solve the timetable stabilization problem for $\mathscr{F}_2$ (expected number of conflicts) and $\mathscr{F}_4$ (critical train influence) is equal.*

5.2.3 Fine-tuning the Algorithm

Depending on the objective function and the instances, several possibilities exist to fine-tune the algorithm. The first point to look at is the neighbor substitution in the algorithm. Basically, there are two policies: During the exploration of $N_k(L)$ either the first feasible substitution that improves the objective function is taken (first-fit), or the the best among all neighbors (best-fit) is selected. The latter rule searches through the entire neighborhood for the best solution. Depending on the evaluation function and the size of the neighborhood, this may take a while. The first rule seeks for a substitution until a better solution is found. The gain in the objective value may be small and a great deal of iterations have to be made until the same objective value is reached.

If a solution can be evaluated quickly and if the neighborhood is not too large, then a best-fit rule is suitable. However, if the evaluation takes a long time and/or neigh-

borhoods are large, then the first-fit rule is preferred in most cases. Moreover, practice shows that the first-fit policy has the best performance in solving Problem 5.1—especially when using the time slots as the objective function. Since the computation of time slots is quite a time consuming operation (see Algorithm 4.1), the time required to achieve a 3- or 4-optimal solution is huge. Depending on the size of the problems, determining a solution to Problem 5.1 can last for days or even weeks only to solve the Bern test cases 3- or 4-optimally, yet it can take minutes to hours to come up with a 1- or 2-optimal solution.

When using $\mathscr{F}_2$, $\mathscr{F}_3$ or $\mathscr{F}_4$ as the objective function, then the most time consuming part is the generation of the extended conflict graph graph since all weights have to be calculated. The optimization finishes within seconds after this input graph has been determined. It is very attractive to have many different starting solutions, because then many different end solutions (schedule alternatives) can be calculated. Thus, it is important to have a method (such as the fixed point iteration algorithm) that generates different solutions for the initial feasibility problem.

The selection of the neighborhood is crucial for the performance of the algorithm. Using a large neighborhood-size results in a large set of substitution possibilities and therefore requires many possible evaluations of solutions. For example, if there are up to 1000 different routing possibilities per train, a neighborhood-size of 3 has to check up to one billion different substitutions. Even on very fast machines one such iteration step takes time. Therefore a neighborhood-size of 1 or 2 is usually preferred. Truncating the 3-neighborhood $N_3(L)$ by randomly selecting a part of the neighborhood to look at, does not perform any better than by looking at a full 2-neighborhood.

In a 1-neighborhood it is simple to complete a subrouting: Through the use of the conflict-graph, impossible completions are detected easily. However, having a larger neighborhood-size brings the feasibility problem back to the spotlight. The larger the neighborhood, the more computation time is needed to satisfy the feasibility problem instead to improve the solution of the timetable stability problem.

Among all test cases in this work, Bern East 2020 is the most challenging one (see also Chapter 3). The time needed to solve the feasibility problem is about 30–45 minutes; needless to say that this problem instance is too complicated to solve in *online* situations. In order to decrease the size of the model, the number of different routes through the station region has to be reduced. Most of the sixteen trains have more than 500 different routes to run through the station network to the designated platform or from the platform to the designated portal. The sum of all the possibilities is about 6'800 different routes, *i. e.* the conflict graph has 6'800 nodes. The number of edges having a weight of 1 (*i. e.* the number of conflicts) is about 7'000'000 of overall 21'000'000 possibilities. In other words, a graph with nearly 7'000 nodes and 21'000'000 edges has to be built for the clique optimization problems (objective functions $\mathscr{F}_2$, $\mathscr{F}_3$, and $\mathscr{F}_4$). Hence, vast computer storage is needed in order to

compute these input graphs and then to solve the corresponding optimization problem.

The only possibility of decreasing this problem instance size is through the reduction of the different routes per train: By randomly truncating the number of routes per train to at most 500, the number of edges in the extended conflict graph shrinks to about 13'500'000. By only allowing 300 different routes per train this number drops to 6'500'000 edges in the extended conflict graph—the number of nodes is still about 3'700. Although the non-truncated instance is feasible, the random cancellation procedure obviously does *not* preserve the feasibility of the problem. However, experience shows that many randomly truncated instances of the Bern East 2020 scenario—but not all—could still be solved, whereas the remaining are believed to be infeasible.

The truncation of the problem yields results that might not even be 2-optimal—in the sense that within the non-truncated variety of routing possibilities, better solutions could be detected. However, comparing the results shows that the truncated optimization problem is still solved fairly well.

Using the time slot objective function does not necessitate computing the entire extended conflict graph (which is a time consuming operation). Only at the end of the optimization, the schedule can be evaluated for the other stability indicator measures. This time, the extended conflict graph only consists of n nodes and $\frac{n(n-1)}{2}$ edges. Therefore, the evaluation of a solution L for *all* objective functions is possible and not too time consuming even for large instances.

5.3 Computational Results

In Appendix B the complete tables of results are shown. However, certain interesting results will be partially illustrated and discussed here. The four different test instances relate to the recent timetable 2003 or the possible future timetable 2020 and to either the west side or the east side of Bern. Note that in the recent 2003 timetable there are 19 train itineraries per half an hour for both the east and the west side. For the 2020 timetable there are 16 train itineraries per quarter of an hour on the east side and 11 train itineraries per quarter of an hour on the west side.

5.3.1 Scenario Setup

In addition to the network topology, the train service intention and a timetable are given; delay patterns for the trains have to be assumed in order to calculate the edge weights of the extended conflict graph. Below the *empirical*, *estimated*, *target*, and *pulse* delay distributions will be explained and used in order to optimize Problem 5.1. Yet, an optimized schedule for one of the delay patterns can also be analyzed for the other distributions, by fixing the routes and re-calculating the weights of the edges

connecting two chosen nodes according to another distribution. The four possible delay scenarios to be considered are (see Section 4.4 as well):

Empirical: Real data has been used to describe the delay distribution for both arrival and departure delays. Here, early arrivals are possible, whereas early departures do not exist. Furthermore, delays always belong to one of the classes ..., -15, 0, +15, ...

Estimated: Based on the delay measurements and on the assumption that delays are *wexp*-distributed, λ and μ have been estimated for both departure delays and arrival delays (see Chapter 4, Equation (4.20)). Note that in these scenarios early departures and arrivals are not possible. However, results show that the stability measures can be very similar compared to the scenarios using empirical delays. The weighted exponential distribution seems adequate to represent train delays, especially departure delays. Usually the stability measures using the estimated distribution are a bit smaller than for the corresponding scenario using the empirical distribution. This is not surprising since the estimation is too optimistic for small delays as Figure 4.8 shows. Hence, more trains are assumed to be on time and fewer conflicts arise when applying the estimated delays. However, the values are similar and the order of magnitude is equal in all result tables (see tables in Appendix B). The advantage of using the weighted exponential distribution is the calculation of random samples. Whereas for discrete distributions random numbers are drawn from a discrete set, the weighted exponential distributed numbers are drawn from $\mathbb{R}^+$. Random samples according to delay distributions will be used in the next chapter.

Target: The SBB has set a target that 75% of all arrivals should be at most one minute late and 95% of all arrivals should be within a time interval of 4 minutes to the scheduled time; moreover, departures should always be on time, of course. Thus, in the target delay scenarios, the *wexp*-distribution is used to model train arrival delays satisfying these demands. Departure delays do not exist in this scenario and thus departing trains are always on time. Hence, by comparison with estimated delay scenarios, the influence of late departures on the stability of the timetable is made visible for the schedules.

Pulse: With the possibility of managing the train speed on-line using ETCS/ERTMS technologies, delays of trains can be controlled. The *pulse* distribution is based on the empirical delay distribution, yet delays now belong to classes ..., -90, 0 , +90, ..., *i. e.* the delay data are pooled into wider classes.

The SBB has never investigated their current schedules according to the four stability measures $\mathscr{F}_1$ through $\mathscr{F}_4$ and the two additional stability indicators (bounds on

the schedule failure probability and extended time slots). Recall that here full scenarios are analyzed, in which a maximum number of trains are running—so called peak-hour scenarios. Moreover, recall that it has been assumed that the train delays are independent, this fact, however, obviously does not hold in reality. Nevertheless, the calculated objective values give good indications about the stability of a schedule, and relative comparisons are still meaningful and very insightful. In order to decide whether condensed timetables could be operationalized, the 2003 timetable will define the threshold to decide whether a schedule is operable or not and hence, the 2003 timetable will act as a reference (see Figure 5.1).

Note that in the 2003 timetable, the trains considered are all scheduled within half an hour, whereas for the future 2020 timetables only a quarter of an hour is needed. However, the stability indicator values are comparable as it can be assumed that improved operational procedures (*e. g.* the introduction of ETCS/ERTMS and the quasi-continuous supervision of the trains, see Appendix A) will compensate for the shorter time period. As the dispatchers will know earlier about potential conflicts, interventions can be planned earlier too, which will result in a better management of the trains.

In order to illustrate the results for the Bern test cases, *performance profiles* will be used. The notation $\boxed{\text{x}}$ will be used to show which objective function has been used to generate a certain schedule L. Points belonging to the same schedule are connected in the performance profiles. On the y-axis the four different objective functions are depicted using the short notion of *time slot* for $\mathscr{F}_1$, *clique* for $\mathscr{F}_2$, *edge* for $\mathscr{F}_3$, and *node* for $\mathscr{F}_4$. The x-axis is divided into seven classes, namely *intolerable*, *poor*, *inadequate*, *sufficient*, *good*, *very good*, and *excellent*. Many (though not all) values for the 2003 scenarios belong to the class *good* and therefore this class represents the «reference» class. Table 5.1 shows how the values are assigned to the classes.

	intolerable	poor	inadequate	sufficient	good	very good	excellent
Time slot	0-5	5-15	15-30	30-45	45-60	60-90	90+
Clique	7+	7-6	6-5	5-4	4-3	3-1.5	1.5-0
Edge	0.85+	0.85-0.8	0.8-0.7	0.7-0.6	0.6-0.4	0.4-0.2	0.2-0
Node	2+	2-1.75	1.75-1.5	1.5-1.2	1.2-0.8	0.8-0.4	0.4-0

Table 5.1: Classes for performance profiles.

5.3.2 Discussion of First Results

East 2003 and West 2003

(See Figure 5.2)

Comparing the east side and west side of Bern for the 2003 timetable, the east side is always inferior, no matter which optimization criterion and which delay distribution is used. For both the estimated and the empirical distribution, the difference in

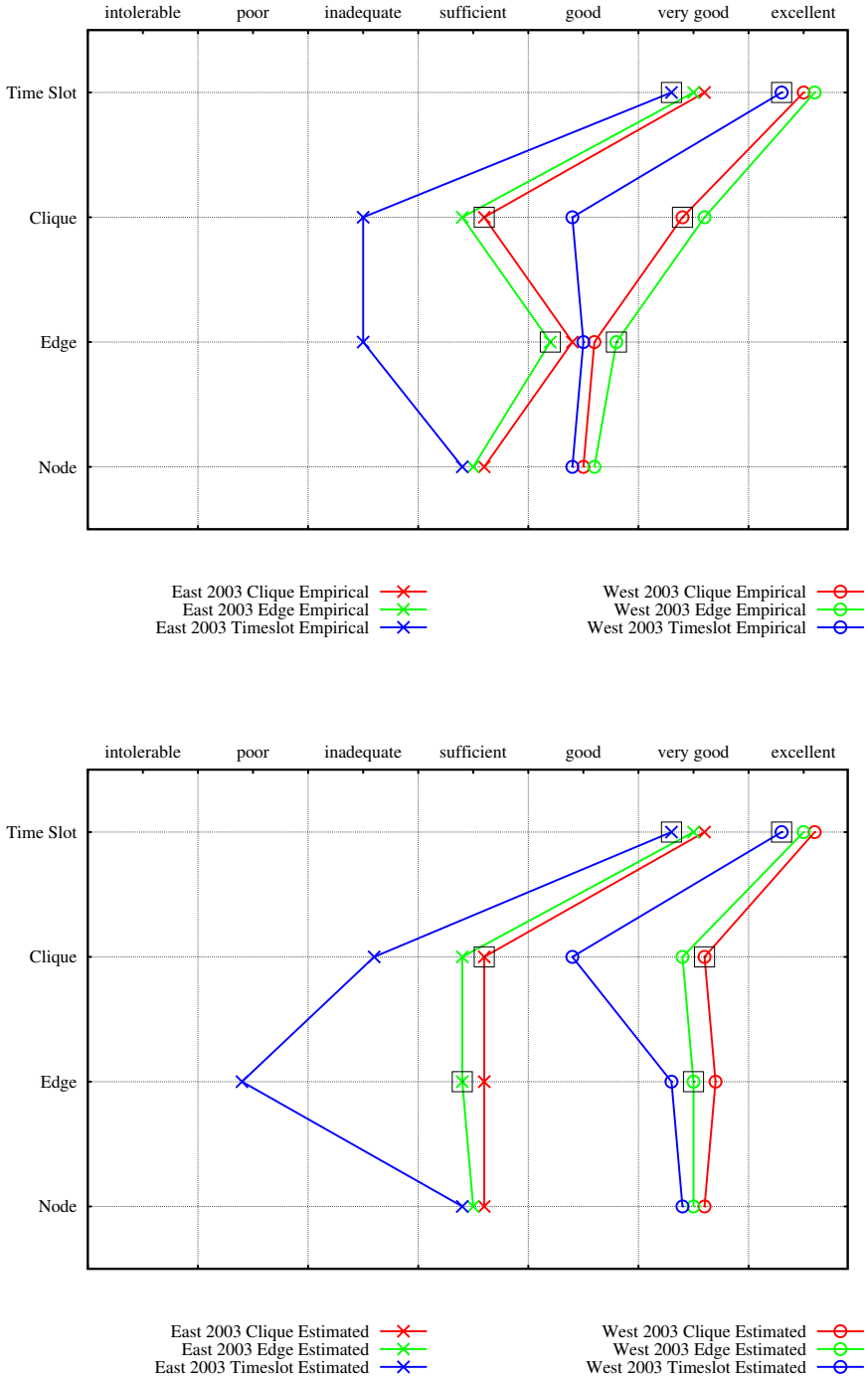

Figure 5.2: Performance profiles for Bern East and West 2003. The pictures show the performance of different optimized schedules using either the empirical (top) or the estimated (bottom) delay distributions.

the expected number of conflicts within a half an hour is roughly 1.5 (about 4 for the east and about 2.5 for the west). The time slot optimization shows that on the east side some trains only have about a minute of spare time for delays until the designated route becomes blocked. On the west side, there are about 2.5 minutes left for delays. However, as exemplified in Appendix B.2, the scheduled passing time is often found towards the *right* of the corresponding time slot. Thus, it appears that much of the space for earlier passing times (arrivals at portals *and* departures at platforms) remains. These buffer times seem to be inadequate and could be used for densification or reorganization of the timetable.

The bounds on the schedule failure show a similar picture: For both, the estimated and the empirical distribution, the schedule failure probability is at least 30% – 45% for the West and 55% – 80% for the East. These values correspond to the maximum edge weight of the induced clique.

Handling the east side is more difficult and therefore, it is indispensable to manage the trains on the east side carefully as the allocation of the resources have to be well thought-out. In order to enhance the stability, the dispatch procedures have to be followed exactly, which can be seen by applying the target delay distribution.

Target 2003

(See Figure 5.3)

Consider the following fictitious delay pattern. All train departures are exactly on time and the train arrival delays are *wexp*-distributed such that the SBB's quality of service rule holds. Minimizing the expected number of conflicts now cuts all clique weights roughly in half. Interestingly, the routes change compared to the other two delay scenarios so that there are fewer possible interactions between the different train itineraries. This validates the conjecture that the stability of timetables and possible interaction between train itineraries are strongly correlated. Moreover, this shows that the routing of the trains is very important in order to increase the stability of the timetable, since stability is only increased if the available resources are well distributed.

Moreover, there is another interesting difference in the numbers: The lower bound for the schedule failure probability dropped to about 25% – 35% on the east side whereas the value of about 30% remained for the west when using the expected number of conflicts as the objective function. If no delays of departing trains occur, then the east side is well organized, as the resources seem to be well allocated. With no other than the *edge*-objective functions this phenomenon could be detected. However, the upper bound on the schedule failure probability is still 1, as no better bound could be found.

Therefore, it can be concluded that on the east side the trains are organized better, than on the west side of Bern. For example, the four trains heading to Zurich or

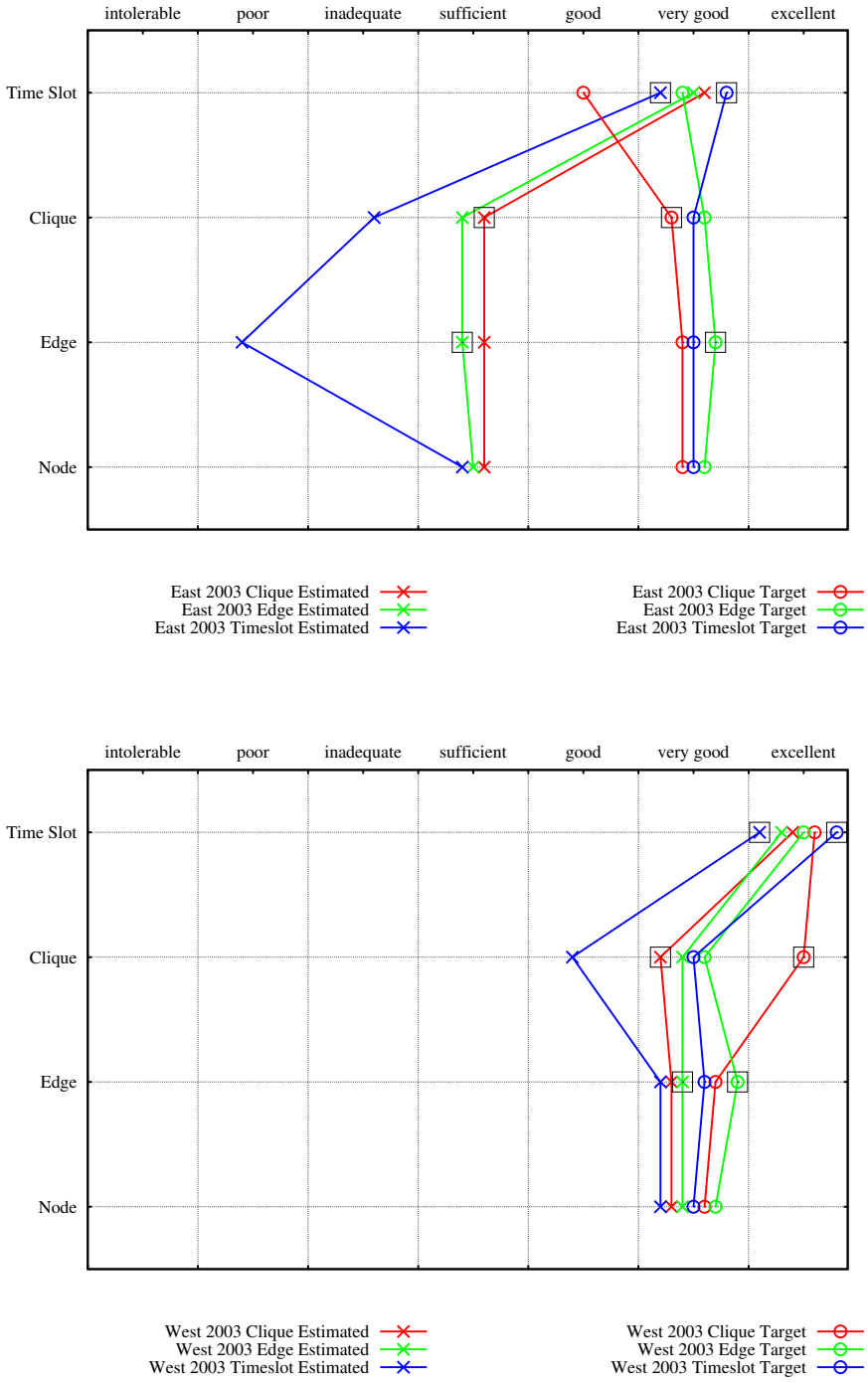

Figure 5.3: Performance profiles for Bern East and West 2003 using the target and the estimated delay distribution.

coming from Zurich respectively depart and arrive time-wise in close proximity to each other. The two trains form a «bundle» of trains traveling «together» through the station area. Roughly speaking, a bundle can be seen as one long train that reserves the track segments for a long time. Especially for arriving trains, the advantage is that the two trains can be interchanged in time. Such bundles simplify the dispatching and operational procedures, but are not considered in the models here.

Since the time slot concept is independent of any underlying delay distribution, the time slot lengths do not change if the delay distributions are altered. Nevertheless, they give a good picture of the «density» of a timetable. The smaller the train's time slots, the denser the timetable (see also [Burkolter, 2005]). Interestingly, the corresponding schedule's time slots, but not the extended time slots, shrink when the expected number of conflicts for the target delay scenarios is minimized. This is an indication that the weights ω should be chosen carefully in the objective function $\mathscr{F}_1$.

West 2003 and West 2020

(See Figure 5.4)

Whereas the train service intention contains 19 trains per half an hour for the 2003 timetable, the West 2020 train service intention foresees 11 train itineraries in a quarter of an hour. The timetable is slightly denser and as expected, the stability indicator values therefore become worse. For the estimated distribution the clique weight as well as the maximum edge weight increase slightly; whereas for the empirical delay distribution, the maximum edge weight even decreases slightly, the expected number of conflicts marginally increases. Moreover, both time slot lengths (simple and extended) also decrease and are about 40% of the length of the West 2003 scenarios.

Yet, looking at the target distribution scenarios, the clique measures are *decreasing* quite dramatically! This contradicts the expectation that a denser timetable is always more sensitive to delays. However, the explanation is that the West 2020 timetable is much more structured, *i. e.* the resource distribution is already well organized from the given train service intention. Yet, as rough line plans of the service intentions are given, it is concluded that the future 2020 line plan is better arranged than today for the 2003 timetable. Hence, delays have less impact on the schedule failure, which can be seen by looking at the corresponding bounds.

Thus, in order to increase the capacity utilization of the network, timetables and line plans must be much more structured and organized than today and the train itineraries must be harmonized, otherwise the stability will not reach the level of the 2003 timetable. Especially by decreasing departure delays, timetables gain more stability. However, decreasing departure delays also means canceling more connections of late inbound trains, which decreases the quality of service for the customers. Line plans, condensed timetables, experienced delay patterns, and a suitable infrastructure have to be designed and well coordinated in order to increase train frequencies while

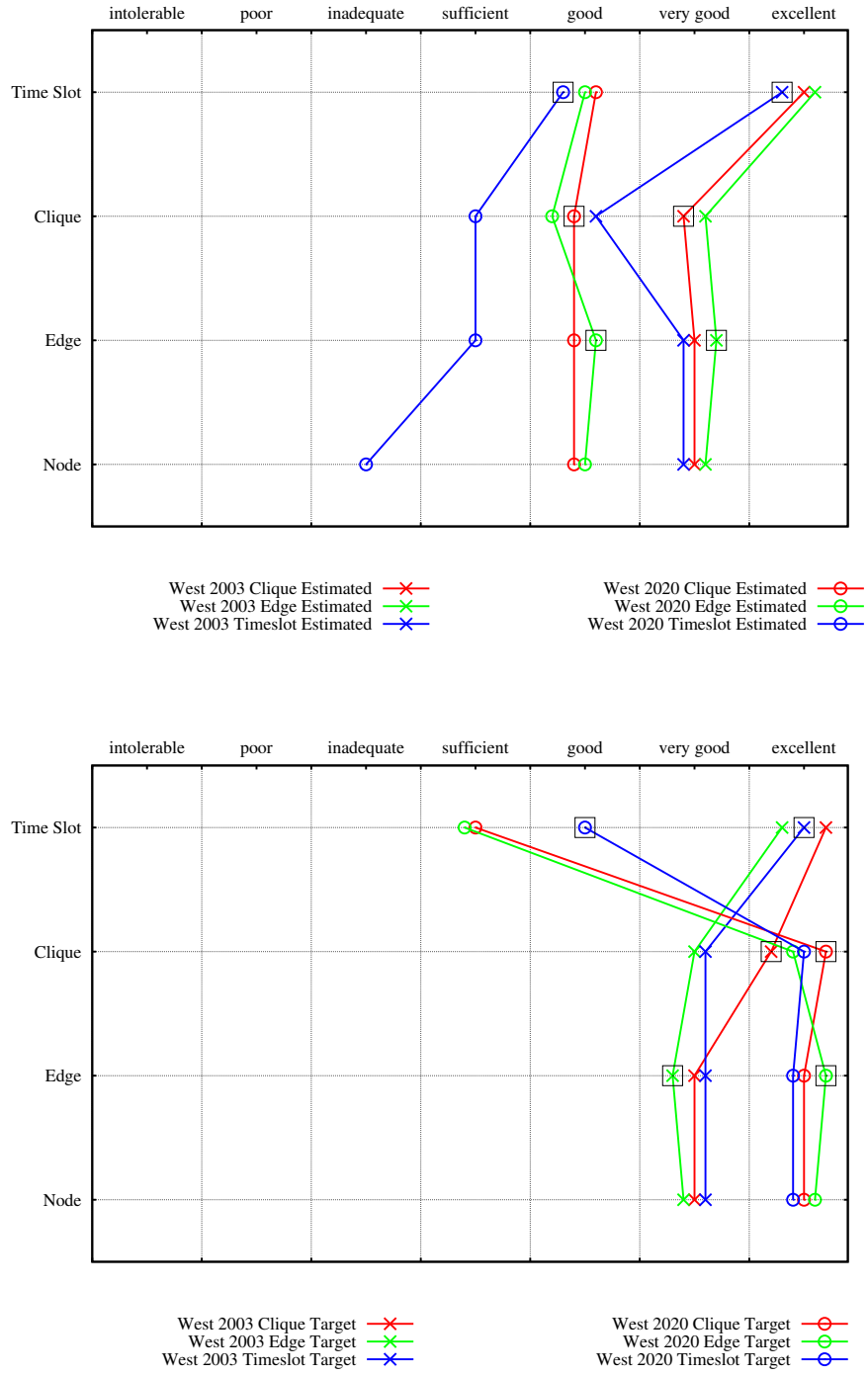

Figure 5.4: Performance profiles for Bern West 2003 and 2020 using either the estimated delay distribution (top) or the target distribution (bottom).

keeping the same quality of the stability of a timetable.

East 2003 and East 2020

(See Figure 5.5)

Compared to the west side, the east side contains five additional train itineraries (16) per quarter of an hour. Although the East 2003 timetable is already complex, the East 2020 timetable is far more complicated. First, the computing time needed to solve the feasibility problem is much longer than for any other scenario (see Chapter 3). Secondly, time slots almost vanish for both the simple and the extended time slot measures. Moreover, the critical train influence as well as the expected number of conflicts are more than doubled using the estimated or empirical distribution.

The stability indicators of the West 2020 timetable are worse than for the corresponding West 2003 timetable using the estimated or empirical delay scenario. Whereas the introduction of target delays helped in the west to increase the stability, in the east the expected number of conflicts for the East 2020 timetable is still on the same level as for the 2003 timetable.

Since the time period has been cut in half as well for the 2020 scenarios, the corresponding stability values do not meet the reference threshold of the 2003 timetable. Even if improved operational procedures will allow more conflicts to be dealt with, the east side of the 2020 train service intention is not operable as it is. But having a closer look at the numbers, one can see that for the target delay scenarios, the weight of *one* train exclusively contributes almost everything to the overall clique weight!

5.3.3 East 2020 Scenario Revisited

In the previously stated facts it was concluded that one input parameter of the East 2020 scenario must be changed. The goal is to alter the scenario inputs slightly in order to meet the same characteristic stability measures as for the reference scenario East 2003, *i. e.* 3–4 expected conflicts, not more than 70% for the maximum weighted edge, and about 20 seconds for the time slots (at least). The time slots may be smaller than they are today, but the schedule failure probability should be about the same. Since in the East 2020 scenario 16 trains are scheduled within 15 minutes, the ultimate aim would be having about 2 expected number of conflicts in order to achieve the same stability as today. Several changes can be applied to increase the stability, such as changes in the network topology, in the line plan, or in the number of trains, changes of the timetable or the underlying delay distributions:

(i) Change the network topology: What happens if additional tracks are introduced? The problem is the scarcity of available space for additional tracks. Therefore only additional switches and track underpasses are considered. Promising seems

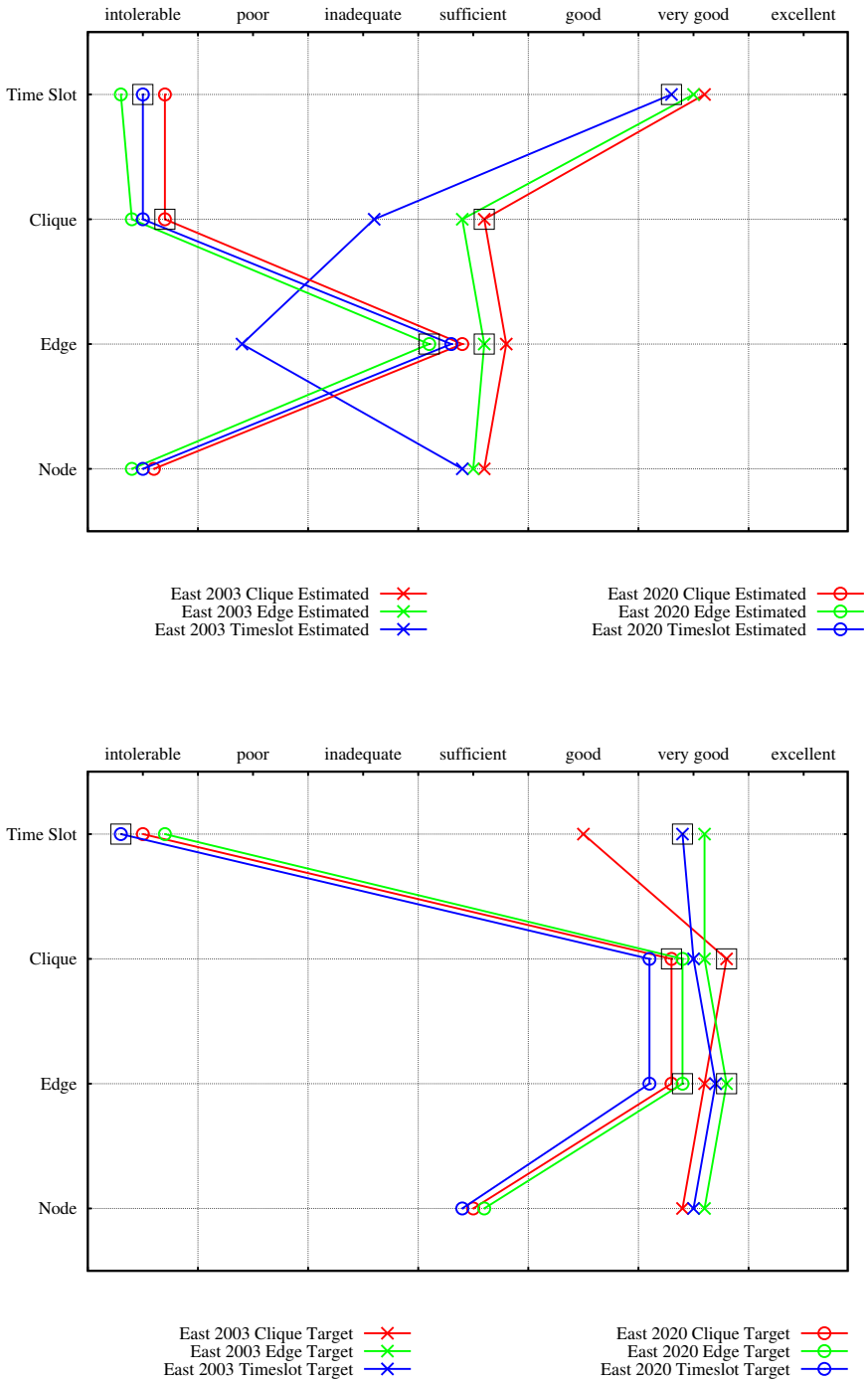

Figure 5.5: Performance profiles for Bern East 2003 and 2020 using either the estimated delay distribution (top) or the target distribution (bottom). Obviously, the condensed timetable is not operable and has to be changed.

to be the introduction of an underpass in the Wankdorf-Dreieck (see Figure 5.7, the direct connection between *A* and *B*).

(ii) Change the train service intention: In addition to the previous corrections two more changes are outlined here. On the one hand, lines could be altered, such that start and end nodes of an itinerary could also be changed in order to deconcentrate the train itineraries such that more buffer time is made available. However, this approach fails, as no better solution could be found. On the other hand, a train itinerary could be removed from the service intention. Naturally, this reduces not only the number of connections and possible interferences, but the attractiveness of public rail transport too. Nevertheless, the question arises of which candidate train should be removed. Since a reduction in the level of service is not acceptable, the question must be answered by including many factors. Here the focus is on operational aspects, as the critical train stability measure—*i. e.* the maximum node weight of the induced clique of a schedule—is a clear indication here (see Figure 5.6). Moreover, by looking at the tables in Appendix B, this approach is very promising, since there is one train having a node weight close to the overall clique weight using the target delay scenarios. Furthermore, the same train is the critical train when looking at the empirical or estimated delay scenarios, although it contributes only about one third of the overall weight.

(iii) Change the timetable: In order to gain stability with the same network topology and train service intention the timetable must either become less dense, *i. e.* the periodicity has to be increased, or the timetable must coordinate the trains better. As a consequence, the capacity utilization is reduced (see Chapter 2) when changing the timetable. Since this is not the topic of the thesis at hand, timetable corrections are not considered further.

(iv) Decrease the delays: A last action to implement is the decrease in delays and hence the change of the delay distribution. In order to gain stability it is necessary to improve arrival and departure procedures. If it is possible to manage the delays more efficiently, then more stable schedules could be developed. Therefore, the *pulse* arrival and departure delay distributions have been introduced (see Section 4.4.4) based on the available sample data. There, the distance between the possible delays is set to the safety distance, *i. e.* only delays as . . . , -90sec, 0sec, +90sec, . . . are possible.

Moreover, in order to see the influence of the more concentrated delay distributions around 0, four new discrete delay distributions having the same number of classes but different representatives are introduced (delay distributions A–D, shown in Figure 5.8). Note that these four distributions only allow small arrival

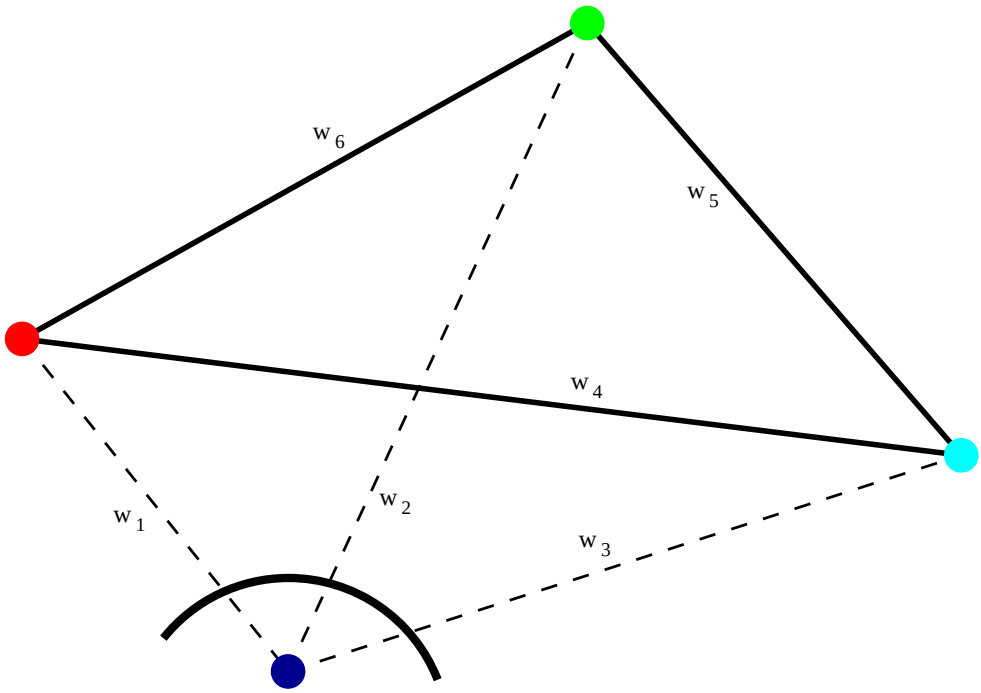

Figure 5.6: Removal of the critical train from the extended conflict graph. If the bottom vertex is deleted, then the weights w_1, w_2, and w_3 are removed as well. Yet, the sum $w_1 + w_2 + w_3$ corresponds to the expected number of conflicts of that train.

delays, up to 90 seconds for distribution A and up to 30 seconds for distribution D. Departure delays are similar: One percent (which in reality is too high!) of all trains depart early from the platform, but the delays do not exceed 90 seconds (distribution A) or 30 seconds (distribution D).

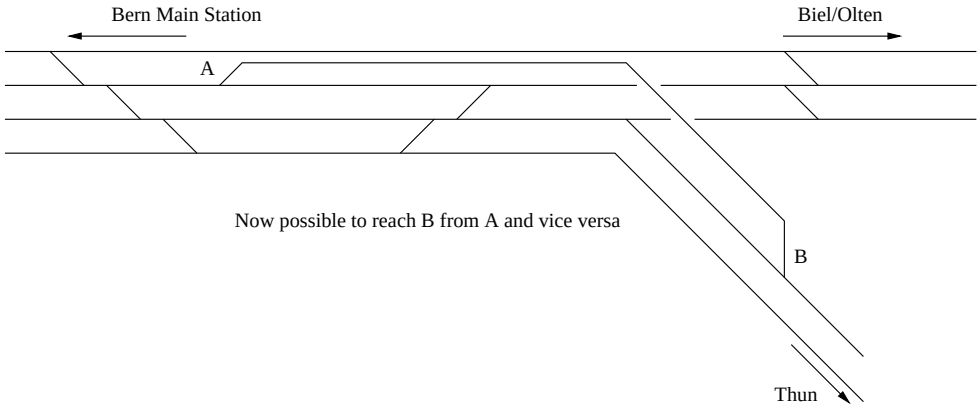

Figure 5.7: Underpass in Switch Region Wylerfeld. With this underpass it is possible to reach A from B and vice versa without disturbing trains traveling east-west on the third parallel track.

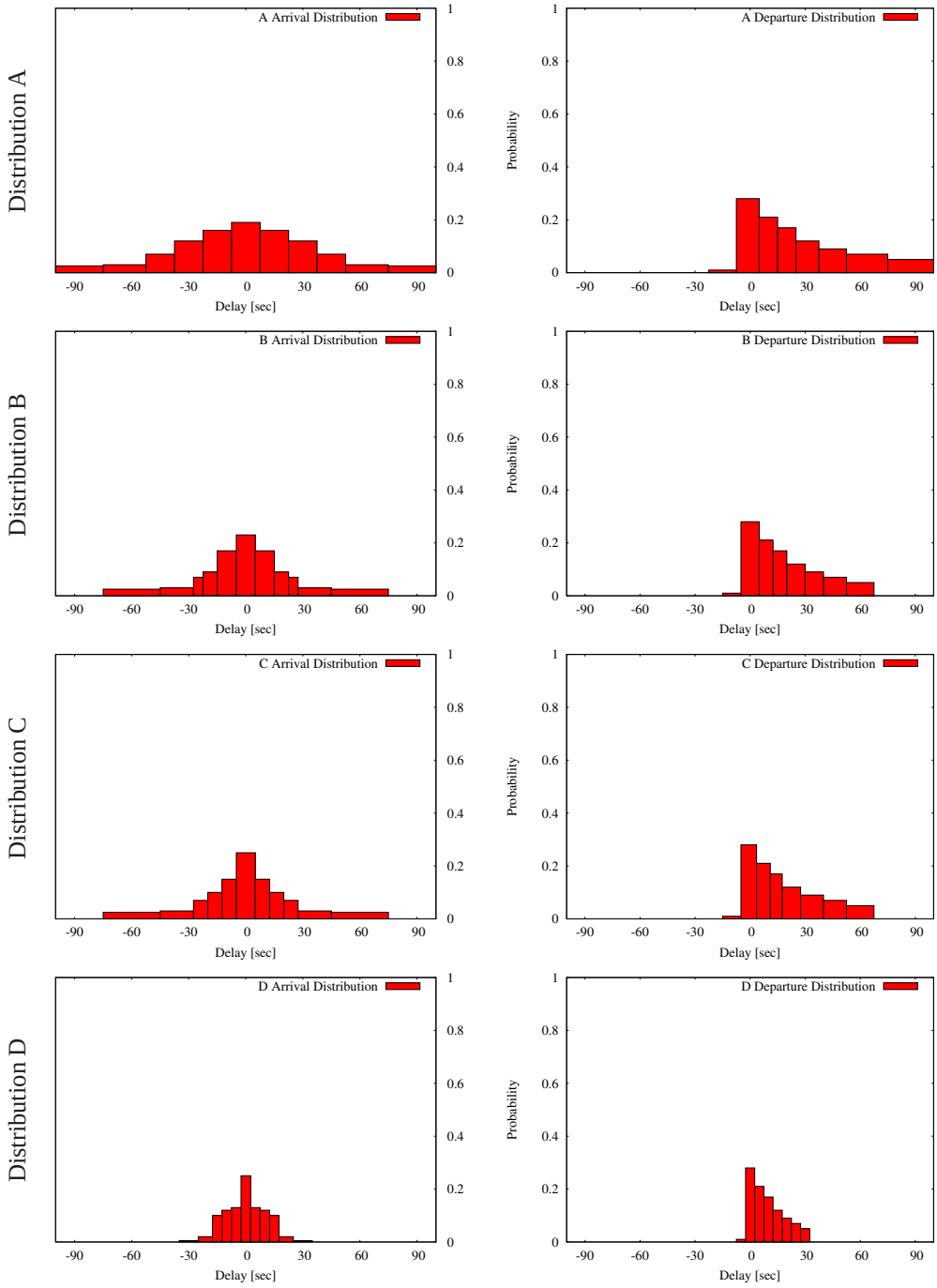

Figure 5.8: Distribution A – Distribution D. In order to focus on small delays the four fictitious discrete distributions A, B, C, and D are constructed.

Comparison of Single Interventions

In a second test series, three adjustments have been applied to the East 2020 scenario. First, the described underpass has been introduced in the network topology. For the second intervention, the most critical train has been removed from the train service intention, and third the *pulse* delay distributions have been introduced. The benefit of all those interventions can be seen in Table 5.2 and Figure 5.9. For the original, the underpass, and the removed train variants, the empirical delay scenario has been used. All four scenarios have been optimized individually using the expected number of conflicts as the objective function.

	East 2020 empirical	East 2020 empirical underpass	East 2020 empirical Critical train removed	East 2020 *pulse*
Expected Number of Conflicts (clique weight)	8.770	7.605	5.512	6.965
Influence of Critical Train (max. node weight)	2.915	1.957	2.361	2.510
Simple Time Slot	1.7	5.1	14.5	1.7
Extended Time Slot	1.7	5.5	15.8	1.7
Schedule Failure Probability (Lemma 4.9)	[0.736,1]	[0.435,1]	[0.578,1]	[0.632,1]

Table 5.2: Stability increase for different interventions.

It is not very surprising that the removal of the critical train has the largest impact on the stability indicators. The removal of the critical train corresponds to the removal of all the nodes in the extended conflict graph corresponding to the critical train (see Figure 5.6). The new schedule has one node less and thus the weight of the clique has to be decreased by at least the expected number of conflicts of the critical train. By a re-routing of the remaining trains an additional gain could be achieved. Here however, an increase of at least 2.915 for the expected number of conflicts is expected, yet a little surplus of 0.343 expected number of conflicts is achieved by calculating a new routing for the East 2020 scenario with removed critical train. Interestingly, by introducing *pulse* delay patterns a reduction of about 1.8 expected numbers of conflicts can be achieved—this is a reduction of about 20% only by changing the delay patterns! Recall that the *pulse* classes are . . . , -90, 0, 90, . . . and that the data

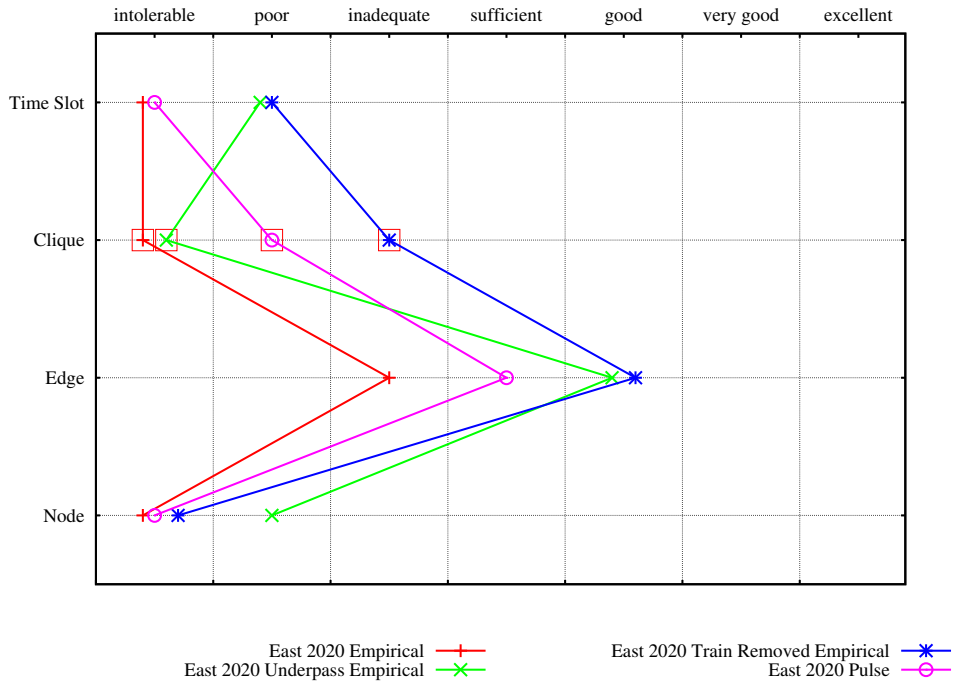

Figure 5.9: Performance profiles for Bern East 2020 with different adjustments.

has been pooled into these classes by using the «always round up rule» for arrival delays and the «half-half rule» for the departure delays (see Figure 4.9). This means that the *pulse* delay distribution assumes *worse* values than the delays coming from real data (empirical distribution). Yet still, with *pulse* delays the expected number of conflicts is smaller than with the empirical distribution. Also the underpass has a certain impact, although it is not that high (see Figure 5.9).

Effects of Narrow Delay Distributions

For the second test runs, the *pulse* delays have been used. Yet, other discrete delay distributions could also be used, since the benefit of concentrated delay probabilities seems to be large. In a third test series, different discrete distributions are applied (distribution A – distribution D, shown in Figure 5.8) in order to observe the impact of more concentrated delays around 0 on the stability of the timetable. Whereas the spectrum for arrival delays has a width of 180 seconds in distribution A, in the corresponding Distribution D case the delays are already quite concentrated around 0 having a width of only 60 seconds. The idea behind the concentration of delays being around 0 is that delays should then more easy to handle; following the intuition that the likelihood that the schedule will fail is small, for small delays. The comparison among the different distributions with respect to the expected number of conflicts is shown in Table 5.3 and Figure 5.10.

	empirical	pulse	Dist. A	Dist. B	Dist. C	Dist. D
Expected Number of Conflicts (clique weight)	8.770	6.965	6.649	5.922	5.663	4.550
Influence of Critical Train (max. node weight)	2.915	2.510	1.950	1.658	1.777	1.355
Simple Time Slot	1.7	1.7	1.7	1.7	1.9	2.0
Extended Time Slot	1.7	1.7	1.7	1.7	1.9	2.0
Schedule Failure Probability (Lemma 4.9)	[0.736,1]	[0.632,1]	[0.653,1]	[0.682,1]	[0.6813,1]	[0.679,1]

Table 5.3: Stability increase for different discrete distributions. The *pulse* distribution serves as a reference value.

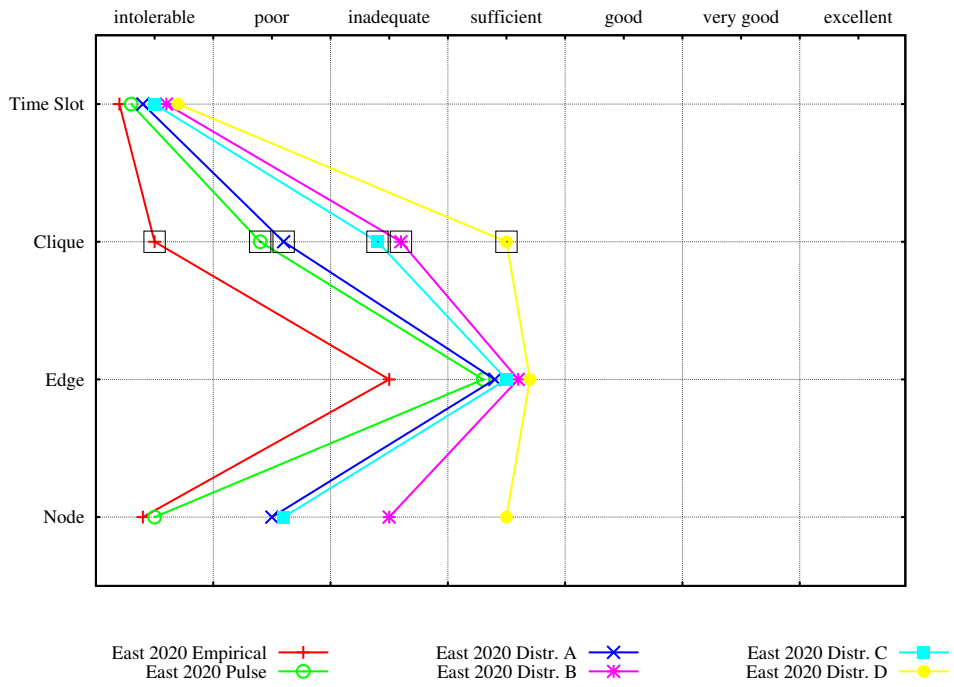

Figure 5.10: Performance profiles for Bern East 2020 with concentrated delay distribution.

Combination of Interventions

Surprisingly, narrowing the distribution does *not* help that much to achieve the ultimate goal of two expected number of conflicts. The *pulse* distribution—for which large delays are possible—is not (much) worse than the distribution A. While tightening distribution A, the value of the expected number of conflicts is decreasing by about two; yet even with a very tight delay distribution such as shown in example distribution D the lower aim of about four expected number of conflicts could not been achieved. It seems that the intended level of service is too tight and that even small delays cause the schedule to fail. Therefore, a combination of the proposed interventions has to be applied. Table 5.4 and Figure 5.12 show the combination of the interventions where 2 or 3 improvements have been applied. The schedule using an underpass introduced to the topology, a reduced train service intention and the *pulse* delay distribution is close to the goal of four expected number of conflicts. Although the requirements are not met, this schedule is considered to be almost sufficient. In order to improve the value of the time slot, the timetable could be adjusted (see Remark 6.4). Note that the results in Table 5.4 depend on the sequence of the applied improvements. Figure 5.11 shows which sequence has been applied to obtain the results. Note that the critical train (the train to remove from the train service intention) has not been the same for the different elimination possibilities, *e. g.* introducing the underpass also changed the critical train.

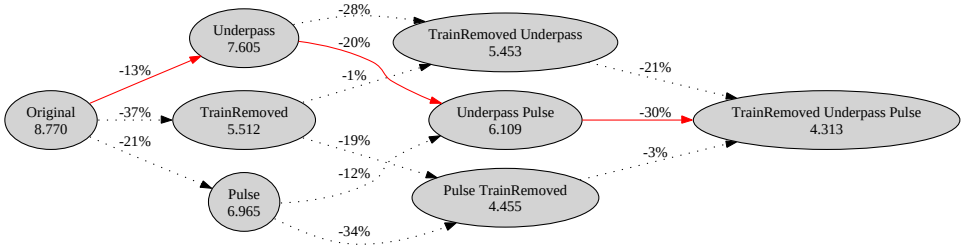

Figure 5.11: Sequence of improvements. The solid line shows the sequence which performed best.

	empirical	Pulse underpass	Pulse train removed	Underpass train removed	Pulse underpass train removed
Expected Number of Conflicts (clique weight)	8.770	6.109	4.455	5.453	4.313
Influence of Critical Train (max. node weight)	2.915	1.193	0.897	1.106	0.980
Simple Time Slot	1.7	5.6	7.0	5.5	5.5
Extended Time Slot	1.7	5.6	7.0	5.5	5.5
Schedule Failure Probability (Lemma 4.9)	[0.736,1]	[0.420,1]	[0.288,1]	[0.407,1]	[0.420,1]

Table 5.4: Stability increase for different combinations of improvements. The original problem serves as a reference value.

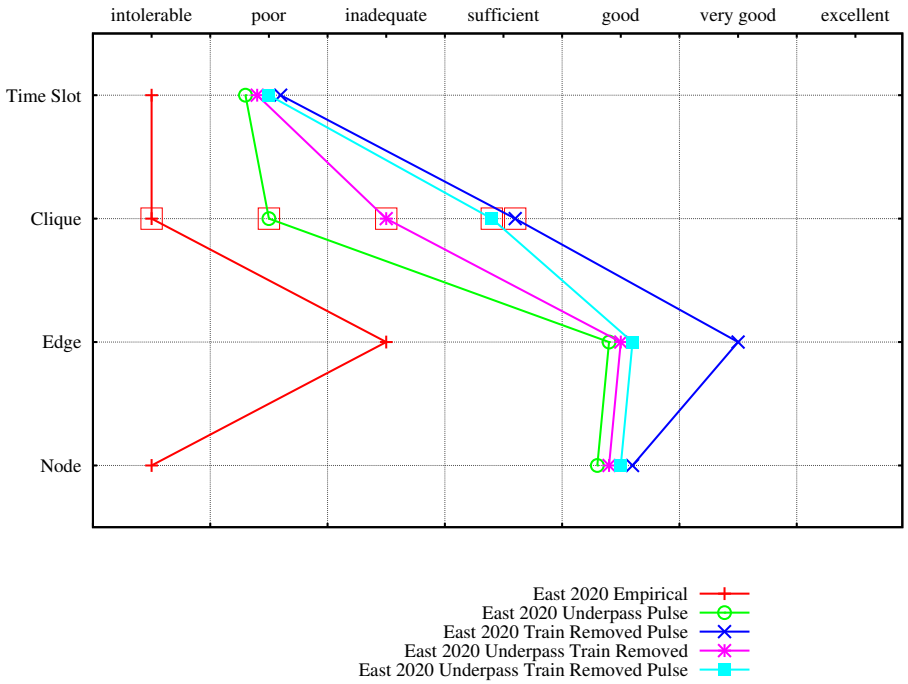

Figure 5.12: Performance profiles for Bern East 2020 with combination of adjustments.

5.3.4 Conclusion

In order to operationalize the full East 2020 train service intention, the introduced combination of interventions (respectively at least one of them) must be applied. The additional infrastructure will lead to a smaller increase in stability than the others, but the flexibility will be increased (see time slot values). Narrowing the delay distribution does help to maintain the schedule, but not as much as expected. Moreover, it seems unrealistic that the train delays could be managed and condensed in such a way that they coincide with one of the distributions A,B,C or D. In addition it would be very difficult—yet not unrealistic—to manage the trains between the node stations in such a way that a *pulse*-like delay distribution can be realized. Nevertheless, in order to increase train frequencies, the design of the timetable is very important. If a delay probability distribution with narrow predefined time slots as sketched in Figure 5.13 could be achieved, then this new kind of delay distribution could be taken into account while constructing new timetables. Although these *pulse* timetables (timetables optimized for *pulse* delay distributions) do not use the track network to full capacity, it seems that a great deal of stability could be gained. Nevertheless, the construction of such timetables is subject to further research activities.

Hence, not yet having *pulse* timetables available, the train service intention on the east side of Bern should be changed in order to meet the same timetable stability as today. The construction of the underpass will help to increase the stability and flexibility of the timetable, as it will be shown in Chapter 6. Since other (more extensive) changes in the network topology are not presently available (yet), the only appropriate solution is through the removal of one or more trains so as to maintain the present level of stability. By improving the punctuality of train operations—especially for departing trains—a tremendous gain in stability could be achieved. If the operational procedures became more reliable and hence the trains' punctuality more accurate, the additional train would not have to be removed.

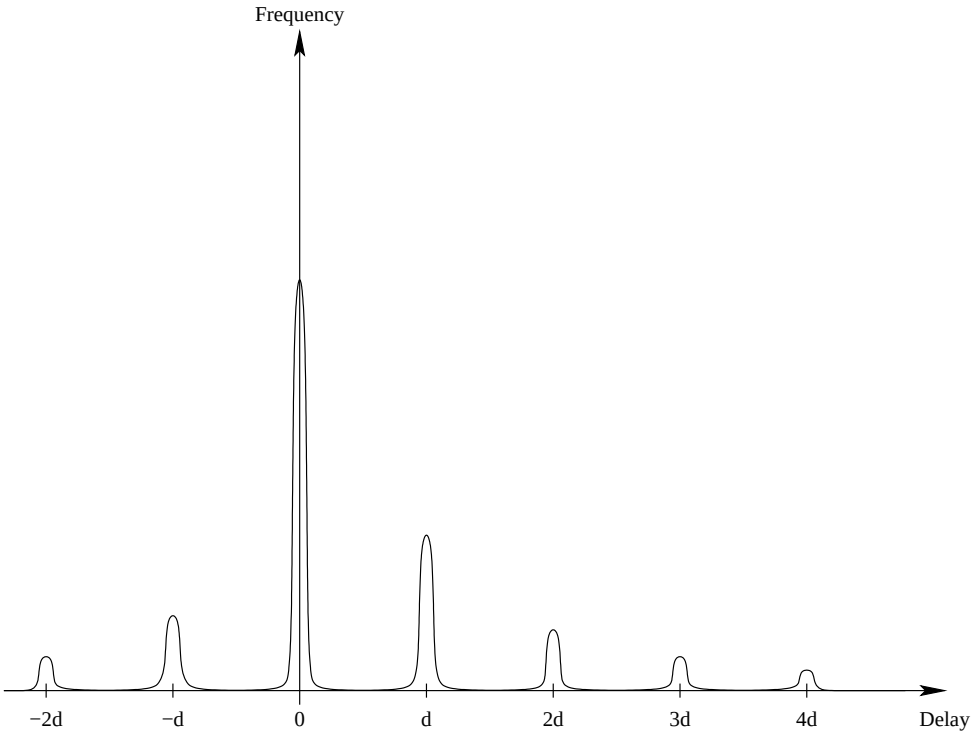

Figure 5.13: Sketch of a continuous distribution with predefined narrow slots. The idea to have condensed delays at certain points $k \cdot d$ in time seems to have promise.

Chapter 6

Sensitivity Analysis

> The possibilities are numerous once we decide to act and not react.
>
> *George Bernard Shaw (1856–1950)*

Whereas in the previous chapter the four stability functions $\mathscr{F}_1$, $\mathscr{F}_2$, $\mathscr{F}_3$, and $\mathscr{F}_4$ have been used to generate stable timetables, the chapter at hand addresses the topic of the sensitivity of timetables and briefly the subject of operational flexibility. Since the delay distributions are assumed to be known in advance, the train routing is determined to enhance the performance of the system whilst still providing a certain level of flexibility for future adaptation in the case of delayed trains. In station areas, the flexibility in reconfiguring the system mainly comes from the routing and to a certain extent from the timetable. In general, the flexibility of a dispatcher is comprised of the ability to re-route the trains within its area of concern (leaving the timetable unchanged) and to develop a new timetable for each train on-line. The latter task does not concern the station area only, but also a greater region containing possibly several station areas and the connection zones in between. As these tasks demand a certain level of coordination between station areas, they are not considered here.

So far, all examination has been done off-line, meaning that trains have been assigned to routes without looking at perturbed situations. The schedules have been generated to develop *plans*. By solving Problem 5.1 with the random-restart local search method, *efficient* schedules have been computed. Moreover, by comparing future condensed scenarios with reference timetables (*e. g.* current timetables), it can be decided whether or not the intended service meets the minimal stability thresholds. Moreover, by analyzing the schedule according to the different stability measures and

delay distributions, certain aspects of the schedule are known too, *e. g.* the train having the minimum time slot or the trains that form the weakest link of the schedule. Furthermore, it turned out in practice that the interval for the schedule failure probability is quite large and thus the question remains of what the «true» schedule failure probability is.

Solving feasibility and optimization problems in order to assign a route to each train in the train service intention, has hold our attention so far. Stability indicators have been developed in Chapter 4 and used as objective functions in Chapter 5 in order to assure a certain stability of the timetable against small deviations. Yet, the question of whether the objective functions yield schedules that are insensitive to small disturbances, has not yet been answered. Furthermore, every (theoretically) operational timetable has been considered to be executable (in practice) so far, but this claim obviously does not hold. A timetable that fails almost certainly cannot be considered to be executable (see also Chapters 2 and 4).

6.1 The On-Line Instance—an Outlook

Recall that a timetable perturbation is given by a triple $(\mathbb{X}, V, \Theta)$, where $\mathbb{X}$ is the vector of delays, V the vector of positions and Θ the vector of relative reveal times of train delays (see Definition 4.1). Usually the perturbations are not known well in advance. While the trains are running in the system, delays may occur at any location in the network. Consider a schedule L and a certain point in time when a delay of a train is revealed. The feasibility problem now depends on the reveal time of the delay, as well as on the current situation, *i. e.* the current positions of the trains in the network. Potential collisions have to be detected and resolved according to the current situation. Such a problem instance is called an *on-line instance*:

Definition 6.1 (On-Line Instance) *An instance consisting of a network topology, a train service intention, a (feasible) schedule L and a given timetable perturbation $(\mathbb{X}, V, \Theta)$, is called an on-line instance.*

If the reveal times ϑ_i are neglected, then an on-line instance is a perturbation of the timetable for which all the delays are already known *in advance*. Hence the problem of determining a route for each train such that the safety restrictions are met and the timetable is adhered to, can be considered as an off-line problem. The on-line timetable feasibility problem, for which the reveal times cannot be neglected, is similar to the timetable feasibility problem (compare to Problem 3.2 in Chapter 3):

Problem 6.2 (On-Line Timetable Feasibility Problem) *Given an on-line instance (network topology, train service intention, timetable, and timetable perturbation); if possible determine for each train a route in the network, such that all safety restrictions are met and the perturbed timetable is adhered to.*

Including the reveal times opens up a new facet: Depending on the reveal times ϑ_i, the on-line instance could be feasible or not in the sense that a routing could or could not be computed. Imagine the following situation: Some trains are running in the system and suddenly information about a delay is revealed. The closer to 0 the relative reveal time is, the harder is it to find a suitable intervention. Many routing alternatives could have been eliminated due to the fact that the delay has been known of too late. This could then cause routing problems, or even timetable problems, as no suitable routing could be found. Vice versa, knowledge of the delay well in advance could have less impact, as the problem could be resolved easily by assigning new routes to the trains. Hence, the smaller the reveal times are, the more the problem looks like an off-line problem, since the delays are known well in advance and many routing alternatives are still available. Therefore, the values of the ϑ_i are important. The smaller the ϑ_i the earlier the operator is informed about delays and hence, more time is left to find «good» alternative schedulings. Thus, the flexible reaction on delays crucially depends on the reveal time of delays.

There is usually a wide spectrum of intervention possibilities for operators. Especially when deviations are large and the trains have large delays, operators have to intervene and react according to the situation. The aim of the sensitivity analysis is to show, which delays can be handled by on-line re-routing (if necessary) and for which kind of delay scenarios other interventions are required in order to ensure safe travel. Since the topic of re-schedulings (computation of new timetables) is beyond the scope of this thesis, on-line instances for station regions are here restricted as follows:

(i) Since early information about delays is considered to be advantageous, $\vartheta_i = 0$ for all $i = 1, \ldots, n$, *i. e.* the reveal times are all equal to zero. The same on-line instances with smaller ϑ_i's are less difficult to solve and in this sense $\vartheta_i = 0$ is the most challenging case.

(ii) The vector V of positions consists only of initial nodes, *i. e.* portal nodes and platform nodes only.

Due to the uncertainty in the system, trains do not arrive at or depart from their initial node on time. In contrast to the off-line situation, in the on-line case perturbations occur while the trains might be running inside the station region. The first crucial question to answer is whether for a given perturbation of the schedule, the planned routes could still be used. If yes, then the perturbation does not cause an intervention and the trains can run as planned. In the other case, the operators face a situation in which intervention is needed.

Moreover, assume a specific situation in which each train has a certain amount of delay (some maybe 0) and the designated routes are no longer feasible. The second question is whether or not the timetable is still feasible. This problem can be reduced to the feasibility problem in Chapter 3 and described as follows: The train service

intention remains the same, yet the timetable is changed. Since by this assumption only the initial nodes of the itineraries are affected by the delay perturbation all nodes on the designated paths receive the same amount of delay as the corresponding initial nodes, *i. e.* passing times are translated by the amount of the delay. Apart from the changed passing times nothing else changes and Algorithm 3.2 can be used to determine the feasibility of an on-line instance. The reveal times indicate at which point in time a new solution to the on-line feasibility problem has to be determined and thus at which point in time Algorithm 3.2 has to be used. This will now be explained in more detail.

Note that due to the above assumptions, re-routings are allowed, yet not any change of the timetable. According to the perturbation, on-line instances can be characterized as follows:

(i) The on-line instance is feasible and the designated routes can still be used.

(ii) The on-line instance is feasible but the routes must be changed (according to the situation).

(iii) The on-line instance is infeasible.

Depending on the perturbation and its reveal times, the entire variety of routing possibilities is no longer available. Since certain trains might be in the station area when information about a train's delay is revealed, their spectrum of possible routes is delimitated. Therefore, the following situation could occur: An on-line instance is infeasible, but by neglecting the reveal time of delays, the instance can be feasible. Thus, the reveal time values are important in order to classify a certain on-line instance into one of three classes (i)–(iii). Delimitated routing possibilities due to reveal times have to be respected while determining re-routings.

In order to decide for a given on-line instance to which category a perturbed instance belongs, the fixed point iteration method can be reused (see Algorithm 3.2). The perturbation of the initial vertex passing times causes a change of some passing times in the timetable. If for the delayed passing times no conflict among the designated vertices arises then the schedule and thus the routing is still feasible. Hence, the decision whether a predefined routing is feasible in the on-line sense is independent of Θ, the reveal times vector.

If two routes become incompatible and the minimum required safety distance is no longer met due to the perturbation, then the schedule cannot be executed as planned. In this case maybe a simple re-routing of the trains proves that the perturbation can still be handled and neither the timetable nor the train service intention is further affected. The information about a delayed train is only available after the reveal time and thus some trains may already have covered some distance *inside* the network. If due to a train's delay a re-routing is necessary, then the full spectrum of routes is no

longer available for those trains already traveling inside the network. Whether the trains can be re-routed or not is then easily decided as follows. Let t^0 be the time at which it has to be decided whether or not the trains can be re-routed. At this time, each train i has a certain position v_i in the network. Either it is already traveling inside the network, *i. e.* the initial node has been left and the end node of its itinerary has not yet been reached ($v_i \neq v_i^{p_a}$, $v_i \neq v_i^s$, $v_i \neq v_i^{p_d}$), or it is still waiting in its initial node ($v_i = v_i^{p_a}$ or $v_i = v_i^s$). Then a new off-line instance can be created by taking v_i as initial node for each train i and the perturbed timetable as new draft timetable. If for this new instance, the fixed point iteration Algorithm 3.2 finds a solution, then a re-routing of the trains is possible. If the fixed point iteration fails then the new instance is believed to be infeasible.

6.2 Results for the Bern Station

For testing the different routing possibilities, each train has its designated route, which it follows according to the schedule L. Additionally, timetable perturbations for a train i are generated with the following properties: (i) Either the arrival at a portal node is delayed, or the departure from the platform node is late, (ii) the amount of delay X_i is distributed according to one of the introduced delay distributions, and (iii) the relative reveal time of the delay is set to 0.

For each single perturbation, the amount of delay is determined by sampling a number from a delay distribution, which depends on whether the train is arriving or departing. Delay distributions as they have been introduced in Chapter 4 will be used to sample delays. For such a generated scenario the goal is to decide whether an intervention from the operators is needed, and if yes, whether a re-routing procedure can be applied to the scenario.

Overall 34 different schedules have been examined (13 for 2003, 16 for 2020, and 5 for 2020 with an additional underpass). 3800 timetable perturbations have been applied using the different delay distributions (*wexp*-distribution, empirical, target (no departure delays), *pulse* (with and without departure delays), A, B, C, and D). An overview of the results is given in Table 6.1.

Table 6.1 shows an interesting phenomenon: Whereas the ratio of infeasible scenarios for West 2003 is about 70%, the East 2003 scenarios fail only in about half of the cases. However, designated routes are feasible in one fourth of the cases on the west side, whereas only about 15% of the schedules on the east side remained feasible when perturbations have been included. Yet, in almost one third of the east side scenarios a simple re-routing algorithm could be successfully applied, whereas on the west side the re-routing procedure failed very often.

For the 2020 scenarios, the picture is the other way around. On the west side the

Pooled schedule	Designated routes feasible	Successful re-routing	Infeasible on-line scenario
West 2003	23.96%	3.87%	72.17%
West 2020	9.27%	49.80%	40.93%
East 2003	14.17%	32.50%	53.33%
East 2020	4.38%	0%	95.63%
East 2020 underpass	4.79%	3.54%	91.67%

Table 6.1: Ratio of feasible, re-routable, and infeasible schedules. The different off-line generated schedules for the same instances have been pooled into the three Bern east side and into the two Bern west side scenarios.

probability that the schedule and the re-routing procedure would fail as well dropped to about 40%, whereas it raised to 90%-95% on the east-side, whether the underpass would have been available or not. Constructing the additional underpass helps to stabilize the timetable in the following way: Although the amount of designated routes remaining feasible marginally increased, some scenarios could be solved by applying the re-routing procedure. Interestingly, the additional underpass increases the operational flexibility—although only a little. Only in a few cases did a re-routing procedure of the trains restore a feasible schedule. This strengthens the conjecture that the East 2020 train service intention is overloaded, even if the underpass were to be constructed.

Objective function

Four different train stability problems have been stated (see (5.5)–(5.8)), each focussing on different stability aspects. Which one had the best performance, *i. e.* which schedule was least sensitive to the perturbations? As Table 6.2 shows, the time slot optimization showed the best performance when looking at the probability of a schedule failing. However, when a schedule became infeasible, then a re-routing procedure was applied less successfully to restore a feasible schedule.

Since conflict and cluster stability optimization are solved by the same heuristic, minimal, maximum node weight and minimal, maximum clique weight are almost always obtained by the same schedule. In the cases where the schedules do not coincide, the values are not completely different and the on-line scenarios could be similarly resolved. Surprisingly flexible are schedules that are optimized for the maximum edge weight, *i. e.* for their structural stability.

Remark 6.3 *20 of the 34 different schedules were optimized for the maximum clique weight, and only for 7 schedules the objectives were the minimum time slots. As more schedules are* clique-*optimized than* time slot-*optimized—especially for the 2020 scenarios—the numbers provided in Table 6.2—to some extent—are misleading. The 2020 scenarios are tighter and therefore more perturbation sensitive, depreciating*

Objective	Designated routes feasible	Successful re-routing	Infeasible on-line scenario
Time slot	19.94%	12.97%	67.10%
Clique	10.10%	21.75%	68.15%
Node (*)	6.44%	65.91%	27.65%
Edge	10.88%	38.47%	50.65%

Table 6.2: Schedule failure probability for the four objective functions. (*) Note: Since the same heuristic is used to determine the *clique*- and the *node*-optimized schedules, the corresponding schedules often coincide. The only case for which quite different schedules could be found is the Bern West 2020 scenario.

the performance of the clique*-optimized schedules. However, the* time slot*-optimized schedules also performed best when looking at the number of feasible designated routes for each of the five scenarios individually and they performed worst when re-routing procedures were applied.*

Delay distribution

In the on-line scenarios, different distributions are used to generate delays for the trains. Using the empirical and the *wexp*-distribution shows an interesting result: Almost all schedules fail and furthermore, they seldom can be re-routed (see Table 6.3)! A complete explanation of this behavior does not exist. Undoubtedly one facet is the probability of long delays: Although for both delay distributions the probability of having more than 5–10 minutes delay *for a single train* is small, the probability of *at least one* train having a delay of more than 10 minutes is not negligible.

A second aspect is the correlation between the train delays. Delays in reality are not independent! For example, arriving trains have to wait for free tracks that are occupied by (delayed) departing trains; and the other way around, departing trains have to wait for late inbound trains. Especially, late departing trains seem to cause serious problems when trying to solve the on-line timetable feasibility problem by a re-routing procedure. This conjecture is confirmed by using delay distributions for which departing trains never become delayed (see below). These dependencies among arriving and departing trains are difficult to analyze and subject to further research.

No departure delay

Those scenarios having no departure delays are more robust against timetable perturbations. Interestingly, *all* feasible scenarios for the East 2020 timetables have been generated by applying a distribution having no departure delay.

Overall, scenarios having no departure delay are most likely to be feasible for the designated routes and these scenarios are also the most flexible ones, especially for the

Pooled Distribution	Designated routes feasible	Successful re-routing	Infeasible on-line scenario
Based on real data	1.10%	10.58%	88.32%
Small delays	5.66%	35.04%	59.29%
No departure delays	34.24%	24.04%	41.13%

Table 6.3: Schedule failure probability depending on the delay distribution used to generate the amount of delays.

Bern West 2020 scenarios. Having no departure delay also shows that the timetable becomes more stable than any other implemented action. On the other hand, no departure delays also implies that train connections are canceled and passengers would miss their direct connections. Of course, no departure delays is an illusion in the real world, but these results show that by introducing exact operational procedures the schedules can be stabilized locally in node stations. On the one hand, no departure delays prevent delays to propagate through the system, as connections are removed as a source of further delays («domino effect»). On the other hand, in a network that is highly utilized—such as node stations—it is important to de-allocate the resources as quickly as possible, *i. e.* outgoing trains have to leave the station region as soon as possible, since the resources are needed for inbound trains. Nonetheless, a global consideration (several node stations and connection zones in between) of this strong conjecture has to be confirmed.

West scenarios

In contrast to the West 2003 scenarios, the West 2020 scenarios are *more* stable if re-routing is included; although the number of trains has been increased from 19 to 22 in half an hour. This shows that the design of the 2020 train service intention (especially of the line plan) is more structured and, furthermore, allows the more efficient use of the spare time between the trains. The level of interaction between the trains is smaller and handling the delays by re-routing procedures is easier. However, the probability of the schedule failing is somewhat larger for the West 2020 scenarios, but the available flexibility of re-routing possibilities is larger and compensates for the higher schedule failure probability.

East scenarios

Having the additional underpass or not does not help to raise the stability of the East 2020 scenarios to the present level. Moreover, re-routing possibilities are rare and therefore the operator has to adjust the timetables in many cases. As the 2003 timetable for the east side of Bern has some flexibility for re-routing, the future intended 2020 timetable is too dense. Remarkably, comparing the two present 2003

timetables, the re-routing procedures can successfully be applied on the east side, whereas the west side seldom allows re-routing methods.

System design

Delay distributions have been used twofold in the scenarios. On the one hand they are used for an underlying assumption on the delays for optimizing schedules, and on the other hand, they are used to generate timetable perturbations. Using schedules that have been optimized for a certain delay distribution are less sensitive against perturbations generated with the same distribution. For example on the west side, using the 2020 scenario and the target distribution to generate delays showed that in 36.70% of the cases the designated routes are feasible for the schedules that were optimized for distributions based on available data, whereas 41.28% of the scenarios remained feasible for the no departure delay optimized schedules. Similar effects could be observed for other distribution combinations for the 2003 scenarios (both sides), though not for the East 2020 scenario. Here, once more, it seems that the train service intention is too tight in order to detect such effects—the different schedules are too similar and the delay generating distributions have only a small impact on whether or not the on-line scenario is feasible.

It is therefore conjectured that for dense timetables, the delay distribution is less important; the layout of the network and a corresponding sound train service intention are more important. On the other hand, for less tight service intentions, stability can be gained by adjusting the routing to the delay distributions. A combination of smart train service intentions and sound operational procedures yields a stable schedule. Moreover, *pulse*-like delay distributions (as outlined in Figure 5.13) should be preferred to rather concentrated delays. Delays narrowly distributed around 0, but still having a remarkable width (*e. g.* one minute for arrival delays and half a minute for departure delays, like distribution D) are more sensitive to perturbations. Therefore, it is conjectured that it would be better to have exact and manageable delays rather than concentrated but still scattered delays around 0.

Schedule failure probability

In Lemma 4.9 only bounds on the schedule failure probability are given. Usually the calculated interval is rather large but it seems that the «true» schedule failure probability is between 75% and 95% and hence rather close to the upper bound. However, these values could be improved by taking the reveal times into account, and exploiting the knowledge of delays in advance. In this sense, the achieved schedule failure probability is an upper bound on the «real» upper bound.

6.3 Conclusion

Summarizing the results for the sensitivity analysis for the Bern scenarios yields several interesting conclusions. Surprisingly, the time slot optimization shows a very good performance with respect to the schedule failure probability, although it does not respect any delay distribution. Tight timetables should be optimized for time slots, since time slot optimized schedules best absorb small delays (the «common» cases). The routing for not so dense timetables should respect other stability characteristics as well as delay distributions. However, in practice train delays are not independent and what the influence of the dependencies on the schedule failure probability is, remains an open question.

Unlike the time slot objective function, the other three objective functions are a relative measure, because they allow the schedules to be compared more deeply. It is concluded that the three clique characteristics are suitable for analyzing the schedules, rather than for optimizing the routing. This statement holds especially for tight timetables. It could still be investigated whether a combination of time slot optimization and *e. g.* clique weight optimization would improve the results.

If the train service intention and the operational aspects are harmonized, then best performance of the system is allowed. It seems that a great deal of stability can be obtained by a carefully planned train service intention. Moreover, the impact of an additional underpass on the stability and the flexibility has been shown for the Bern station. Here, the questions remain of whether the tested underpass is the most suitable extension to the existing infrastructure, and how such adequate infrastructure adjustments in the topology are obtained in general.

Remark 6.4 *For small timetable perturbations, the designated routing often remained feasible. Yet, these small perturbations can also be seen as a change in the timetable, since only departure times at platforms and passing times at portal nodes have been changed. Then, it is interesting whether the stability measures are increasing or decreasing. In most of the cases the values got worse but there were still some new timetables that yield better values and for which the initial node passing times are only marginally changed. Hence, this shows that additional stability can be gained by jittering the initial times and thus changing the timetable slightly. Yet, investigating adaptions in the timetable to increase the stability are beyond the scope of this thesis. However, the best ever found timetable for the Bern East 2020 scenario (only differing slightly from the original timetable) has an expected number of conflicts of 7.873, whereas the original timetable has 8.770 expected number of conflicts—an improvement of over 10%. However, the other stability characteristics changed only insignificantly.*

Chapter 7

Conclusions

A conclusion is the place where you get tired of thinking. *Arthur Bloch (*1948)*

The planning of public rail transport systems is very complex. A suitable track network has to be established and adapted from time to time, line plans have to be designed, a public timetable has to be published, and schedules for trains, their drivers and other employees must be generated. All of these tasks have to be coordinated in order to meet all the hard and soft restrictions.

A substantial but very sophisticated task is the generation of train schedules, especially when the frequency of the services is to be increased. Passenger and freight transport will considerably increase over the next years and therefore the service of trains on the existing track networks must be optimized; in particular the efficiency must be increased. Since by condensing the timetables, capacity bottlenecks are assumed to appear at main stations, the railway traffic in main station areas has to be exceptionally well coordinated. Therein, the routing of trains through the station's network is of great interest. Whereas the routings are manageable by an expert today, the increase of the service level necessitates a computer-aided routing, since the routing alternatives will be tremendous if timetables are further condensed. Moreover, detection of potential conflicts and stability issues will become more important when designing new timetables.

7.1 Summary

The two level approach presented in this thesis allows two difficult tasks, which occur during the generation of a sound schedule, to be separated: Generation of timetables and routing of trains are separated into two parts. Along with the generation of timetables goes the maximization of the track utilization, whereas stability aspects of a schedule are handled by routing the trains through the network. While for generating timetables only aggregated routes (in an aggregated topology) are needed, routing and stability aspects of timetables have to be discussed on exact topologies. Analyzing the stability of timetables has been done so far by the introduction of buffer times. Although additional time constraints inherently inhibit the network from a higher utilization, they are needed in order to deal with perturbations of the timetable. In order to keep the used buffer times as small as possible, the train traffic in station areas has to be well coordinated.

In the thesis at hand, the routing of the trains through a given network has been addressed for different problem setups. As a basis, a fast probabilistic algorithm has been developed that quickly solves even large train routing problems. Especially for dense timetables, the presented algorithm worked very well.

The presence of train routing allows the timetable to be analyzed for its stability. In order to meet the numerous demands on the stability of a timetable, four different stability measures and two additional stability indicators have been introduced. Whereas the concept of time slots is independent of any assumed train delay distribution, the concepts of conflict-, structural, and cluster-stability of a schedule are based on assumed departure and arrival delay distributions. In practice, train delays are not independent and therefore the calculated stability measures are imprecise. However, the order of magnitude of the several characteristic stability measures seems to be correct.

The presented method to calculate stability indicator values allows the underlying delay distributions to be changed without changing the algorithms. Both discrete and continuous distributions have been used and it has been shown that the only crucial computation necessary to analyze the stability of timetables is the convolution of the difference of two random variables describing the delays. For discrete random variables a general approach has been shown, whereas for a continuous distribution the weighted exponential distribution has been used. Both types of distribution have advantages and disadvantages, which have been addressed.

It has been shown that stability itself is difficult to measure, since it depends on many factors. First of all, stability can only be measured when a timetable and thus a train service intention and a train network are available. Secondly, stability also depends on the assumption of train delay distributions. Changing the delay distribution changes the values, but it also shows what level of stability can be achieved by

altering the distribution. Thus, the notion of stability thirdly depends on the system design. Surprisingly, narrowing the distribution does *not* help to gain stability for tight timetables, which has been shown for the Bern East 2020 test case. In order to increase the stability it is better to have broad but exact delays instead of concentrated but scattered delays. Moreover, it has been shown that for station areas, the departure process of trains has to be improved since departing trains have a great influence on the level of stability. The tighter the timetable is the more important this issue is!

According to the selected stability measures four optimization problems have been stated. In order to solve the problems a simple but effective random-restart local search technique has been applied. Since the algorithm solves the problems only heuristically there is uncertainty about the quality of the obtained «optimal» results. However, the given values are bounds on the optimal values and it is conjectured that they are nearly optimal values.

Along with the stability measures a critical train can always be identified, which mostly prevents the objective function from decreasing (when minimizing the objective function). According to this notion of critical trains, the impact of the removal of the critical train on the stability of a timetable could be shown. Additional modifications of the input parameters, such as the delay distributions or additional infrastructure, are easily implemented and show that stability can be improved by changing the train service intention or the network. The question remains, how suitable infrastructure adjustments in the topology are obtained in general.

In a third part, on-line scenarios have been generated in order to test the sensitivity of the schedules. These tests support several conjectures. First of all, it could be shown that in several cases a simple re-routing can be applied and no changes in the timetable are necessary in order to meet all safety restrictions. This supports the hypothesis that stability and flexibility in the choice of the routing are strongly correlated. However, it could also be shown that for very dense timetables there is not sufficient spare room to deal with perturbations, since re-routing algorithms failed to find feasible solutions. The best performance, with respect to the schedule failure probability, was obtained by time slot optimized schedules, although delay distributions are not included in the optimization. The tighter the timetable the more important the time slots are in order to cope with small delays (the «common» cases).

Therefore, the price for a condensed timetable (as for example Bern East 2020) is the loss of stability and flexibility. Either the trains can be managed in such a way that they do not miss their designated time window (upon arrival *and* departure) or else the schedule is very likely to fail. Moreover, designated time windows are very narrow, and concentrated around specific values. If the train service intention and the operational aspects are harmonized then best performance of the systems is allowed. For example the adjustments of the delay distributions is made available through new technologies. The introduction of *pulse* delay distributions, where the

delays are distributed on some few discrete values with equal distance in between, seems to be a promising approach. It is conjectured that the safety time distance is a suitable value for the interval length between two possible delays. However, if such a rigid system is unacceptable, then the train service intention and thus the utilization of the track network has to be decreased in order to meet the same level of stability as today.

7.2 Outlook

While looking at the stability of a schedule, several interesting questions are left to be considered in detail. A topic of great interest is the modeling of train delays. First, if stronger bounds on the schedule failure probability can be found then the schedule failure probability is another good stability indicator. In order to achieve this goal the probability that three or more trains have conflicting routes has to be determined. Together with this more elaborated calculation of the probabilities, the problem of correlation of the train delays has to be addressed.

If dependencies between the train delays are introduced, then the distributions have to be adjusted as well. Other distributions may indeed fit the current data of train delays better than the weighted exponential distribution, but in order to manage dense timetables it is indispensable to design the operation newly from scratch. Specially designed discrete distributions have a tremendous impact on the stability values. Train departures and arrivals at main stations must not be widely scattered around their calculated time, but very concentrated around few time points for which the trains are easy to manage. An initial approach using this concept has been made by designing the Bern 2020 timetables and it is assumed that delays occurring in multiples of the safety distance are much easier to handle than other, even more concentrated delays. Following this design principle, it is very interesting to see whether dense timetables that make use of the special delay distributions, can be generated in order to gain stability.

Seamlessly attached to this problem is the coordination of the trains *between* the main stations. It has to be shown that the trains *can* be managed in such a way that they arrive at predefined narrow slots. As a further consequence, trains have to leave the station exactly at the designated time, otherwise, they have to wait for a next «open time window». Yet, this causes the well-known problems of late occupied tracks that cannot be used for incoming on-time trains. It must be discussed whether provided connections can be canceled in order to execute on-time departures at the main stations. Obviously, each removal of connections reduces the quality of the service and increases the unpleasantness for passengers. Hence, the rules for canceling a connection must be pinpointed inside the company and communicated to the customers.

From a capacity and stability point of view, the utilization of the tracks can be

increased by a subtle planning of the schedule without decreasing today's stability. Having ETCS Level 2 available, the increase of utilization of the track network can be expanded, yet the question of whether the usage of ETCS Level 3 has a similar impact on the utilization of a station's track network has to be discussed. It may be worth looking at several «small» investments that locally improve the management of the trains, in order to fully exhaust the available operational improvements.

In order to cope with delays in tight scenarios it is important to know which «flexibility» is available and which can be used to solve the problems. Like train service intention design, flexibility aspects must be considered while generating new (dense) timetables. Discussing the operational flexibility, another very challenging field of improvements opens up: What shall be done when a timetable perturbation necessitates an intervention that cannot be resolved by re-routing the trains? Sophisticated algorithms are needed that can handle perturbations on-line and adjust both the routing and the timetable. In order to implement such methods in practice, the algorithms have to be transparent (no black boxes) and exceptionally fast (real-time problems). Re-scheduling problems are very difficult to solve algorithmically due to the on-line configuration of the problem. Therefore, the influence of the reveal time of perturbations on the re-scheduling problem has to be discussed. Having reveal times as early as possible helps to solve the re-scheduling problem, since the available flexibility can be better utilized. Moreover, more time is left to find solutions and hence the performance of the real-time algorithm can be improved.

Appendix A

European Train Control System

A method to further condense train traffic in station areas is the reduction of the safety times by introducing *ETCS* (European Train Control System, see [SBB, 2005c], and [The Union of European Railway Industries UNIFE, 2005]). A goal of ETCS is to replace the various signaling and train protection systems currently in use all over Europe. Today, trains are equipped with many different national protection and navigational systems. Each one is extremely costly, as it takes up space on-board and increases the maintenance costs. Moreover, a train crossing from one European country to the next must switch between operating standards when crossing the border and it has to be guaranteed that the different safety systems are not interfering each other.

ETCS enables denser traffic, as the block distance can be reduced, permitting a smaller headway between two trains, without reducing today's safety standards. Since 2000, different aspects have been tested and implemented on designated routes in various countries such as Germany, Austria, Italy, France and Switzerland. Rail companies expect from the implementation of ETCS:

(i) a reduction in the cost of the infrastructure, as stationary traffic signs can be reduced,

(ii) cheaper purchasing and maintenance of new hard- and software,

(iii) an improvement in transnational traffic,

(iv) an increase in capacity utilization,

(v) and a decrease in travel times.

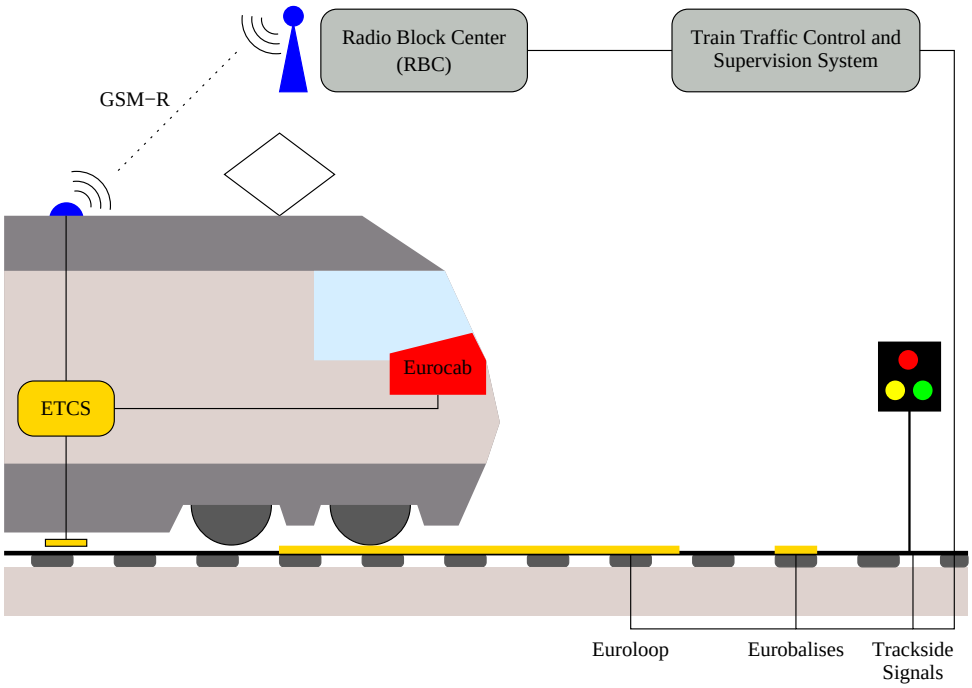

Figure A.1: Schematic overview of the ETCS components.

There are three different levels of ETCS, which allow to realize new different operational functionalities. Within the new system, safety and control management of the trains are merged. Today the tracks are partitioned into fixed segments (so-called blocks). The safety rule assures that at most one train is in a block at any time.

Up to ETCS Level 2 a block system is still needed, yet the block length is much smaller. This enables a quasi-continuous supervision of the trains and an approximation of the moving block system. The information about whether a train is allowed to enter the next block is transmitted by *Eurobalises* and *Euroloops*. These are stationary or wire-based transponders that transmit the information and commands to a driver's control console (*Eurocab*) inside the cab. ETCS standardizes the information transfer between trackside safety installations and engines. Instead of gathering the information on occupied blocks by outside signals, a train driver gets all the information on-board. An engine that is equipped with a Eurocab is technically allowed to travel on any ETCS-track.

ETCS Level 1 complements existing infrastructure elements. The train driver still receives travel authorities from trackside signals. The Eurobalises are responsible for a pointwise control of the train. They transmit the signal status and the current maximally allowed speed. Neither a Radio Block Center nor radio technologies are used to manage trains. The difference from conventional signals and control systems

is the continuous on-board detection and storage of current speed and acceleration values, which are continuously supervised.

In ETCS Level 2, the Eurobalises only locate the position of the trains; they are not responsible for the broadcast of other information. With the help of radio and radar technology the current position is verified with the Eurobalises, but the ETCS computer in the train transmits the position via Euroradio to the central computer of the rail control center. On the other hand the rail control center transmits new travel commands (stop, speed, ...) to the trains. However, the tracks are still partitioned into fixed segments, and the safety rules do not allow two trains to be on the same segment.

The main difference between ETCS Level 2 and Level 3 is the removal of the fixed track segments. The rail control center continuously checks the distance between the trains. With this level the trains are allowed to tailgate independent of block segments but only depending on speed and acceleration.

With the introduction of a *Radio Block Center (RBC)* for ETCS Level 2, the installation of *Euroradio* and *Global System for Mobile communication - Railways (GSM-R)* all data and driving commands are transmitted by a radio broadcast system. Trackside signals can be removed, although they can still be used as a fall-back safety system. In Figure A.1 the ETCS components are illustrated.

The reduction of the headway is possible if a rail company implements at least ETCS Level 2. Some European railroad companies implemented and tested the first two levels of ETCS. The Italian rail company FS tested Level 2 between Florence and Arezzo in 2000. The German railways DB and the French company SNCF did the same on the track Halle/Leipzig to Jüterbog and Marles-en-Brie to Tournan respectively in the same year. The Austrians followed one year later, when they tested ETCS between Vienna and Nickelsdorf. In Spain, Austria, Italy, Hungary, Romania, Bulgaria and Greece ETCS Level 1 has been successfully introduced between 2002 and 2005. The Swiss Federal Railways SBB tested Level 2 in 2002 between Zofingen and Sempach. The DB tests Level 2 on the same tracks as Level 1, and the SBB plans to implement Level 2 between Olten and Bern in 2007 (on the new route that has been introduced to the network in December 2004). In the next few years several new tracks (especially tracks belonging to the highspeed network) will be equipped and operated with ETCS Level 2.

The German rail company DB estimates the time for a pan-European introduction of ETCS to 15 to 20 years. The cost forecast is about 500 Mio. Euros in Germany, and about 8 Bln. Euros for entire Europe [Wikipedia, 2005]. More information about ETCS in Europe is available at several internet sites, such as [The Union of European Railway Industries UNIFE, 2005] or [SBB, 2005b].

Appendix B

Tables of Optimization Results

In the following tables, the different objective functions, the underlying delay distributions and the different scenarios for the Bern test cases are distinguished. One result is shown—the best solution ever found, which best optimizes the corresponding criterion. Note that for the time slot optimization, a weighted average of the four smallest time slots has been used for the objective function. Therefore, the shown value corresponds to the *weighted* average of the smallest time slots. The first three column headers describe clique properties, then simple and extended time slots are shown and finally, the last column describes the interval of the schedule failure probability. The clique weight corresponds to the expected number of conflicts, whereas the maximum node weight corresponds to the influence of the critical train and the maximum edge weight to the weakest link.

Within the *estimated delay distribution*, arrival and departure delays are *wexp*-distributed. The parameters have been estimated using real data. The *empirical distribution* uses the empirical data, but grouped into classes of 15 seconds. In *target delay distribution* scenarios, there are no departure delays, but arrival delays that are *wexp*-distributed.

B.1 Maximization of Minimal Time Slot

	Maximum edge weight	Maximum node weight	Clique weight	Time slot	Extended time slot	Schedule failure probability bounds
Estimated Delay Distribution						
East 2003	0.816	1.455	5.970	83.4	83.4	[0.816,1]
East 2020	0.694	2.977	10.520	2.0	2.0	[0.694,1]
West 2003	0.339	0.688	3.309	148.6	148.6	[0.339,1]
West 2020	0.607	1.699	4.619	56.9	56.9	[0.607,1]
Empirical Delay Distribution						
East 2003	0.750	1.357	5.920	83.4	83.4	[0.750,1]
East 2020	0.736	3.137	9.776	2.0	2.0	[0.736,1]
West 2003	0.443	0.818	3.309	148.6	148.6	[0.443,1]
West 2020	0.566	1.535	4.259	56.9	56.9	[0.566,1]
Target Delay Distribution						
East 2003	0.281	0.643	2.182	83.4	83.4	[0.281,1]
East 2020	0.388	1.431	1.801	2.0	2.0	[0.388,1]
West 2003	0.300	0.539	1.769	148.6	148.6	[0.300,1]
West 2020	0.137	0.256	0.403	56.9	56.9	[0.137,0.403]

Note: The largest time slot that could be found for the Bern East 2003 scenario resulted from a schedule that has been optimized for the maximum weighted edge property, using a target delay distribution. Using the minimal time slot as the objective function, an objective value of 73.6 could be found (which however could also be obtained by a clique weight optimized schedule).

B.2 Time Slot Before and After Scheduled Passing Time for an Example Timetable 2003

	Time slot before	Time slot after
Thun – Bern	114 / 114	5 / 5
Bern – Thun	69 / 69	115 / 115
Schwarzenburg – Bern	4 / 4	182 / 184
Bern – Langnau	1 / 8	206 / 206
Belp – Bern	35 / 35	320 / 320
Bern – Biel	259 / 259	3 / 3
Laupen – Bern	581 / 581	82 / 82
Bern – Thun	445 / 445	2 / 7
Bümpliz Nord – Bern	- / -	354 / 354
Bern – Burgdorf	11 / 11	3 / 3
Langnau – Bern	- / -	150 / 151
Bern – Schwarzenburg	103 / 276	70 / 70
Thun – Bern	149 / 149	446 / 446
Bern – Fribourg	146 / 146	510 / 510
Biel – Bern	2 / 2	293 / 293
Bern – Belp	319 / 319	- / -
Burgdorf – Bern	2 / 2	411 / 411
Bern – Bümpliz Nord	81 / 89	62 / 62
Konolfingen – Bern	5 / 5	251 / 252
Bern – Schwarzenburg	629 / 685	- / -
Fribourg – Bern	- / -	104 / 104
Bern – Fribourg	509 / 509	569 / 570
Neuchatel – Bern	- / -	461 / 505
Bern – Neuchatel	134 / 134	82 / 83
Murten – Bern	117 / 117	36 / 37
Lucerne – Bern	205 / 266	6 / 6
Bern – Geneva	569 / 569	- / -
Zurich – Bern	99 / 99	164 / 164
Biel – Bern	- / -	12 / 262
Bern – Biel	89 / 89	- / -
Bern – Neuchatel	426 / 564	- / -
Bern – Olten	292 / 292	- / -
Geneva – Bern	243 / 248	627 / 628
Bern – Zurich	78 / 101	149 / 208
Basel – Bern	253 / 253	267 / 267
Bern – Brig	114 / 114	90 / 90
Zurich – Bern	163 / 163	55 / 126
Bern – Interlaken	253 / 389	- / -

Note: In an entry *x* / *y*, *x* corresponds to the simple time slot and *y* to the extended time slot. The «Time slot before/after» columns mean that the train may pass (arrive or depart) its initial node at most *x* seconds prior/later to the designated passing time and still find its designated route free, assuming all other trains travel on-time. It may also arrive *y* seconds earlier/later, but then it may have to change its route, assumed all other trains have no delay. A «- / -» means, that none of the considered trains is blocking the route (altough *e. g.* a train of the next nodal time may block the designated route).

B.3 Minimization of Maximal Edge Weight

	Maximum edge weight	Maximum node weight	Clique weight	Time slot	Extended time slot	Schedule failure probability bounds
Estimated Delay Distribution						
East 2003	0.649	1.355	4.114	73.6	74.4	[0.649,1]
East 2020	0.694	2.977	10.520	2.0	2.0	[0.694,1]
West 2003	0.339	0.759	2.511	138.4	138.8	[0.339,1]
West 2020	0.452	1.068	3.768	48.9	48.9	[0.452,1]
Empirical Delay Distribution						
East 2003	0.575	1.349	4.192	73.6	74.4	[0.575,1]
East 2020	0.736	2.915	8.770	1.7	1.7	[0.736,1]
West 2003	0.443	0.817	2.822	145.9	145.9	[0.443,1]
West 2020	0.398	1.158	3.576	48.9	48.9	[0.398,1]
Target Delay Distribution						
East 2003	0.281	0.643	2.182	83.4	83.4	[0.281,1]
East 2020	0.388	1.446	1.685	1.2	1.9	[0.388,1]
West 2003	0.300	0.539	1.769	148.6	148.6	[0.300,1]
West 2020	0.119	0.231	0.231	30.8	32.2	[0.119,0.231]

Note: Using the maximum edge weight as the objective function always gave the same maximum edge weight as indicated above, yet the clique weights or the time slots were worse, *i. e.* the indicated schedules often dominated the edge weight optimized schedules.

B.4 Minimization of Clique Weight

	Maximum edge weight	Maximum node weight	Clique weight	Time slot	Extended time slot	Schedule failure probability bounds
Estimated Delay Distribution						
East 2003	0.649	1.355	4.114	73.6	74.4	[0.649,1]
East 2020	0.694	2.770	9.578	1.7	1.7	[0.694,1]
West 2003	0.339	0.759	2.511	138.4	138.8	[0.339,1]
West 2020	0.452	1.138	3.095	52.9	53.2	[0.452,1]
Empirical Delay Distribution						
East 2003	0.575	1.349	4.192	73.6	74.4	[0.575,1]
East 2020	0.736	2.915	8.770	1.7	1.7	[0.736,1]
West 2003	0.443	0.922	2.641	138.2	141.7	[0.443,1]
West 2020	0.398	0.926	2.824	52.9	53.2	[0.398,1]
Target Delay Distribution						
East 2003	0.342	0.655	1.768	56.6	58.6	[0.342,1]
East 2020	0.388	1.446	1.685	1.2	1.9	[0.388,1]
West 2003	0.300	0.586	1.240	138.2	141.7	[0.602,1]
West 2020	0.119	0.231	0.231	30.8	32.2	[0.119,0.231]

B.5 Minimization of Maximal Node Weight

	Maximum edge weight	Maximum node weight	Clique weight	Time slot	Extended time slot	Schedule failure probability bounds
Estimated Delay Distribution						
East 2003	0.649	1.289	5.610	52.6	54.4	[0.649,1]
East 2020	0.694	2.770	9.578	1.7	1.7	[0.694,1]
West 2003	0.339	0.688	3.309	148.6	148.6	[0.339,1]
West 2020	0.465	1.052	3.436	50.2	50.2	[0.465,1]
Empirical Delay Distribution						
East 2003	0.575	1.349	4.192	73.6	74.4	[0.575,1]
East 2020	0.736	2.915	8.770	1.7	1.7	[0.736,1]
West 2003	0.443	0.817	2.822	145.9	145.9	[0.443,1]
West 2020	0.398	0.926	2.824	52.9	53.2	[0.398,1]
Target Delay Distribution						
East 2003	0.360	0.562	1.948	73.6	74.4	[0.360,1]
East 2020	0.388	1.403	1.695	1.2	1.9	[0.388,1]
West 2003	0.300	0.539	1.769	148.6	148.6	[0.300,1]
West 2020	0.119	0.231	0.231	30.8	32.2	[0.119,0.231]

Bibliography

[Adenso-Diaz et al., 1999] Adenso-Diaz, B., Olivia González, M., and González-Torre, P. (1999). On-line timetable re-scheduling in regional train services. *Transportation Research Part B*, 33:387 – 398.

[Ahuja et al., 1993] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows – Theory, Algorithms, and Applications*. Prentice Hall.

[Akian et al., 1994] Akian, M., Cohen, G., Gaubert, S., Quadrat, J., and Viot, M. (1994). Max-plus algebra and applications to system theory and optimal control. In *International Congres of Mathematicians, Zurich, Switzerland*.

[Alon et al., 1999] Alon, N., Arad, U., and Azar, Y. (1999). Independent sets in hypergraphs with applications to routing via fixed paths. In *RANDOM–APPROX*, pages 16–27.

[Anderegg et al., 2002] Anderegg, L., Penna, P., and Widmayer, P. (2002). Online train disposition: To wait or not to wait? Working Paper.

[Baccelli et al., 2001] Baccelli, F., Cohen, G., Olsder, G. J., and Quadrat, J.-P. (2001). *Synchronization and Linearity — An Algebra for Discrete Event Systems*. Web-Edition.

[Barber et al.,] Barber, F., Salido, M., Ingolotti, L., Abril, M., Lova, A., and Tormos, P. An interactive train scheduling tool for solving and plotting running maps.

[Blanchini et al., 2001] Blanchini, F., Miani, S., Pesenti, R., Rinaldi, F., and Ukovich, W. (2001). *Robust Control of Production-Distribution Systems*, chapter 2, pages 13 – 28.

[Blum and Roli, 2003] Blum, C. and Roli, A. (2003). Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. *ACM Computing Surveys*, 35(3):268–308.

[Boppana and Halldórsson, 1992] Boppana, R. and Halldórsson, M. (1992). Approximating maximum independent sets by excluding subgraphs. *BIT*, 32:180 – 196.

[Bracewell, 1999] Bracewell, R. (1999). *The Fourier Transform and Its Applications*, chapter 5, pages 69–97. McGraw-Hill, New York, 3 edition.

[Burkard, 2000] Burkard, M. (2000). *A Continuous Relaxation Based Heuristic for a Class of Constrained Semi-Assignment Problems*. PhD thesis, Swiss Federal Institute of Technology, Zurich.

[Burkolter, 2005] Burkolter, D. (2005). *Capacity of Railways in Station Areas using Petri Nets*. PhD thesis, Swiss Federal Institute of Technology, Zurich.

[Burkolter et al., 2005] Burkolter, D., Caimi, G., and Herrmann, T. (2005). *HEART Reference Manual*. Institute for Operations Research, Swiss Federal Institute of Technology, Zurich.

[Bussieck et al., 1996] Bussieck, M., Kreuzer, P., and Zimmermann, U. (1996). Optimal lines for railway systems. *European J. Oper. Res.*, (96):54–63.

[Bussieck et al., 1998] Bussieck, M., Lübbecke, M., Winter, T., and Zimmermann, U. (1998). Discrete optimization in rail transport. Extended abstract, TU Braunschweig, Mathematische Optimierung, Pockelstrasse 14, D-38106 Braunschweig, Germany.

[Bussieck et al., 1997] Bussieck, M., Winter, T., and Zimmermann, U. (1997). Discrete optimization in public rail transport. *Mathematical Programming, Series B*, 79(1–3):415–444.

[Caimi, 2004] Caimi, G. (2004). Routing mit Zielfunktion für den Bahnhof Bern. Master's thesis, Swiss Federal Institute of Technology Zurich.

[Caimi et al., 2005] Caimi, G., Burkolter, D., and Herrmann, T. (2005). Finding delay-tolerant train routings through stations. In *Operations Research Proceedings 2004*, pages 136 – 143. Springer.

[Caprara et al., 2000] Caprara, A., Fischetti, M., and Toth, P. (2000). Modeling and solving the train timetabling problem. To appear in *Operations Research*.

[Caprara et al., 2001] Caprara, A., Monaci, M., Toth, P., Fischetti, M., Guida, P. L., and Sacco, G. (2001). Solution of real-world train timetabling problems.

[Cochand, 1993] Cochand, M. (1993). A fixed point operator for the generalised maximum satisfiability problem. *Discrete Applied Mathematics*, (46):117–132.

[Feo and Resende, 1995] Feo, T. A. and Resende, M. G. C. (1995). Greedy randomized adaptive search procedure. *Global Optimization*, 6:109–133.

[Frauenfelder and Herrmann, 1999] Frauenfelder, P. and Herrmann, T. (1999). Betrachtung und Behandlung von Störungen in einem Taktfahrplan. Documentation of term project, Institute for Operations Research, Swiss Federal Institute of Technology, Zurich.

[Garey and Johnson, 1979] Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability; A Guide to the Theory of NP-Completness*. W. H. Freeman.

[Giuliari et al., 2000] Giuliari, M., Pellegrini, F., and Savio, S. (2000). Moving block and traffic management in railway applications: The EU project COMBINE.

[Goemans, 1994] Goemans, M. X. (1994). On-line algorithms. Lecture notes.

[Goverde, 1998a] Goverde, R. (1998a). Optimal scheduling of connections in railway systems. Paper prepared for the 8th WCTR, Antwerp.

[Goverde and y Koelemeijer, 2000] Goverde, R. and y Koelemeijer, G. S. (2000). *Performance Evaluation of Periodic Railway Timetables: Theory and Algorithms*. Number S2000/2 in Trail Studies in Transportation Science Series. The Netherlands TRAIL Research School, Delft.

[Goverde et al., 1998] Goverde, R. M., Bovy, P. H., and Olsder, G. J. (1998). The max-plus algebra approach to transpotation problems. Paper prepared for the 8th WCTR, Antwerp.

[Goverde, 1998b] Goverde, R. M. P. (1998b). Synchronization control of scheduled train services to minimize passenger waiting times. Technical report, Transportation Planning and Traffic Engineering Section, Faculty of Civil Engineering and Geo Sciences, Delft University of Technology.

[Goverde and Odijk, 2002] Goverde, R. M. P. and Odijk, M. A. (2002). Performance evaluation of network timetables using peter. In Allan, J., E., A., Brebbia, C. A., Hill, R. J., Sciutto, G., and Sone, S., editors, *Computers in Railways VIII*, WIT Press, Southampton.

[Grötschel et al., 1999] Grötschel, M., Buckler Powell, W., and Zimmermann, U. (1999). Tagungsbericht 44/1999, traffic and transport optimization. *http://www.mfo.de/Meetings/Meeting_Program_1999.html#T9945.*

[Grötschel et al., 2001a] Grötschel, M., Krumke, S. O., and Rambau, J. (2001a). *Online Optimization of Large Scale Systems — State of the Art*, chapter Online Optimization of Complex Transportation Systems, pages 705 – 729. Springer-Verlag Berlin Heidelberg.

[Grötschel et al., 2001b] Grötschel, M., Krumke, S. O., Rambau, J., Winter, T., and Zimmermann, U. T. (2001b). *Online Optimization of Large Scale Systems — State of the Art*, chapter Combinatorial Online Optimization in Real Time, pages 679 – 704. Springer-Verlag Berlin Heidelberg.

[Gunawardena, 2001] Gunawardena, J. (2001). From max-plus algebra to nonexpansive mappings: A nonlinear theory for discrete event systems. *http://www.jeremy-gunawardena.com/.*

[Halldórson et al., 2002] Halldórson, M., Iwama, K., Miyazaki, S., and Taketomi, S. (2002). Online independent sets. *Theoretical Computer Science*, 289(2):953 – 962.

[Hedinger, 1996] Hedinger, E. R. (1996). *Qualitätsmanagement von Eisenbahnstrecken am Beispiel der Einfädelung bei hoher Geschwindigkeit*. PhD thesis, Swiss Federal Institute of Technology, Zurich.

[Heusch et al., 1997] Heusch, P., Meisgen, F., and Speckenmeyer, E. (1997). CATS-Computer Aided Tram Scheduling. Technical Report 97-262, Institut für Informatik, Universität zu Köln, Germany.

[Hürlimann, 2002a] Hürlimann, D. (2002a). *Objektorientierte Modellierung von Infrastrukturelementen und Betriebsvorgängen im Eisenbahnwesen*. PhD thesis, Swiss Federal Institute of Technology, Zurich, Switzerland.

[Hürlimann, 2002b] Hürlimann, D. (2002b). Opentrack, simulation of railway networks. *http://www.opentrack.ch/.*

[Ireland et al., 2004] Ireland, P., Case, R., Fallis, J., Van Dyke, C., Kuehn, J., and Meketon, M. (2004). The canadian pacific railway transforms operations by using models to develop its operating plans. *Interfaces*, 34(1):5 – 14.

[IRSE, 2004] IRSE, I. T. C. (2004). Quality of service in railway traffic management systems. Technical Report 7^{th} Report, Institution of Railway Signal Engineers.

[Kall and Wallace, 1994] Kall, P. and Wallace, S. W. (1994). *Stochastic Programming*. John Wiley & Sons.

[Kaminsky, 2001] Kaminsky, R. (2001). *Pufferzeiten in Netzen des spurgeführten Verkehrs in Abhängigkeit von Zugfolge und Infrastruktur*. PhD thesis, Universität Hannover, Germany.

[Kannengiesser and Wiche, 1987] Kannengiesser, F. and Wiche, B. (1987). Einfluss der Fahrzeitenzuschläge und Pufferzeiten auf die Pünktlichkeit. *Die Bundesbahn*, 63(11):1001–1007.

[Kimbler, 1997] Kimbler, D. (1997). Petri nets. *http://taylor.ces.clemson.edu/ie340/files/340-16.htm*.

[Köhler et al., 2002] Köhler, E., Möhring, R. H., and Skutella, M. (2002). Traffic networks and flows over time. Technical Report 752, TU Berlin, Germany.

[Korte and Vygen, 2000] Korte, B. and Vygen, J. (2000). *Combinatorial Optimization — Theory and Algorithms*, volume 21 of *Algorithms and Combinatorics*. Springer.

[Laube and Schaffer, 2001] Laube, F. and Schaffer, H. (2001). Kapazitätssteigerung — Schlussbericht Phase 1. Technical report, Swiss Federal Railways.

[Liebchen and Peeters, 2001] Liebchen, C. and Peeters, L. (2001). Some practical aspects of periodic timetabling. Technical report, TU Berlin and Rotterdam School of Management.

[Liebchen et al., 2004] Liebchen, C., Proksch, M., and Wagner, F. H. (2004). Performance of algorithms for periodic timetable optimization. Technical Report 21, TU Berlin, Germany.

[Lindner, 2000] Lindner, T. (2000). *Train Schedule Optimization in Public Rail Transport*. PhD thesis, Technische Universität Braunschweig, Germany.

[Lucchini et al., 2001] Lucchini, L., Curchod, A., and Rivier, R. (2001). Transalpine Rail Network: A Capacity Assessment Model (CAPRES). Institute of Transportation and Planning, EPFL, Lausanne, Switzerland.

[Lüthi, 2005] Lüthi, M. (2005). Train Delays and the LogLogistic-Distribution. Private Communication.

[McGettrick, 2000] McGettrick, M. (2000). Partial synchronicity and the (max,+) semiring. *http://arxiv.org/abs/math.OC/0010095*.

[Meng, 1991] Meng, Y. (1991). *Bemessung von Pufferzeiten in Anschlüssen von Reisezügen*, volume 46. Verkehrswirtschaftliches Institut der RWTH Aachen.

[Middelkoop and Bouwman, 2001] Middelkoop, D. and Bouwman, M. (2001). SIMONE: Large scale train network simulations. In Peters, B., Smith, J., Medeiros, D., and Rohrer, M., editors, *Proceedings of the 2001 Winter Simulation Conference*, pages 1042–1047.

[Möhring, 2000] Möhring, R. H. (2000). Scheduling under uncertainty: Optimizing against a randomizing adversary. Technical Report 681, TU Berlin, Germany.

[Montigel, 1992] Montigel, M. (1992). Representation of track topologies with double vertex graphs. In Murthy, T., Young, F., Lehmann, S., and Smith, W., editors, *Computers in Railway*, volume 2 of *Computational Mechanics Publications*, Washington D.C.

[Montigel, 1994] Montigel, M. (1994). *Modellierung und Gewährleistung von Abhängigkeiten in Eisenbahnsicherungsanlagen*. PhD thesis, Swiss Federal Institute of Technology, Zurich.

[Odijk, 1997] Odijk, M. A. (1997). *Railway Timetable Generation*. PhD thesis, Delft University of Technology, Netherlands.

[Parkes and Ungar, 2001] Parkes, D. C. and Ungar, L. H. (2001). An auction-based method for decentralized train scheduling. In *AGENTS'01*, Montreal, Canada.

[Pelillo, 1999] Pelillo, M. (1999). Heuristics for maximum clique and independent set.

[SBB, 2004] SBB, editor (2004). *Sondernummer Bahn 2000*. Number 8 in via. Vogt-Schild/Habegger Medien AG, Solothurn.

[SBB, 2005a] SBB (2005a). *http://www.sbb.ch*.

[SBB, 2005b] SBB (2005b). ETCS Projects in Europe. *http://mct.sbb.ch/mct/infrastruktur/informationen_projekte/etcs/etcs-europa.htm.*

[SBB, 2005c] SBB (2005c). ETCS Technologie – Funktionsprinzip. *http://mct.sbb.ch/mct/infrastruktur/informationen_projekte/etcs/etcs-technologie/etcs-technologie-funktionsprinzip.htm.*

[Schwanhäusser, 1974] Schwanhäusser, W. (1974). *Die Bemessung der Pufferzeiten im Fahrplangefüge der Eisenbahn*, volume 20. Verkehrswirtschaftliches Institut der RWTH Aachen.

[Serafini and Ukovich, 1989] Serafini, P. and Ukovich, W. (1989). A mathematical model for periodic scheduling problems. *SIAM J. Disc. Math.*, 2(4):550–581.

[Siemens and Alcatel,] Siemens and Alcatel. Rail Guard – Newsletter. Newsletter.

[Stalder and Laube, 2004] Stalder, O. and Laube, F. (2004). The efficient railway – a field of action for formal methods. Invited Paper at FORMS 2004.

[Stork, 2001] Stork, F. (2001). *Stochastic Resource-Constrained Project Scheduling*. PhD thesis, Technische Universität Berlin, Germany.

[Subiono and Van Der Woude, 2000] Subiono and Van Der Woude, J. (2000). Power algorithms for (max,+)- and bipartite (min,max,+)-systems. Technical report, Faculty of Mathematics and Science, Surabaya Indonesia and Subfaculty of Technical Mathematics and Informatics, Delft University of Technology.

[The Union of European Railway Industries UNIFE, 2005] The Union of European Railway Industries UNIFE (2005). ERTMS, European Rail Traffic Management System. *http://www.ertms.com.*

[Ullius, 2004a] Ullius, M. (2004a). Reale Zugsverspätungsdaten basierend auf OpenTimeTable. Private Communication.

[Ullius, 2004b] Ullius, M. (2004b). *Verwendung von Eisenbahnbetriebsdaten für die Schwachstellen- und Risikoanalyse zur Verbesserung der Angebots- und Betriebsqualität*. PhD thesis, Swiss Federal Institute of Technology, Zurich.

[Ullius, 2005] Ullius, M. (2005). Delay analysis of rail 2000 1^{st} phase using opentimetable. In 5^{th} *Swiss Transport Research Conference*, Monte Verità, Ascona.

[van Egmond,] van Egmond, R. An algebraic approach for scheduling train movements. Delft University.

[van Egmond, 1999] van Egmond, R.-J. (1999). *Railway Capacity Assessment, an Algebraic Approach*. Number S99/2 in Trail Studies in Transportation Science Series. The Netherlands TRAIL Research School, Delft.

[Verweji and Aardal, 1999] Verweji, B. and Aardal, K. (1999). An optimisation algorithm for maximum independent set with applications in map labelling. In *LNCS*, volume 1643 of *LNCS*, pages 426–437. ESA '99, Springer.

[VIA, 2000] VIA (2000). *Werkzeuge für die Planung und Führung des Bahnbetriebs*, volume 3 of *Eisenbahnbetriebswissenschaftliches Kolloquium*, VIA, RWTH Aachen.

[Weidner, 1997] Weidner, T. (1997). Verspätungsübertragung im Netz des Personenfernverkehrs. Master's thesis, Rheinisch-Westfählische Technische Hochschule Aachen, Germany.

[Weisstein, 2005] Weisstein, E. W. (2005). Delta function. *http://mathworld.wolfram.com/DeltaFunction.html.* From MathWorld–A Wolfram Web Resource.

[Wiendahl, 1997] Wiendahl, H.-P. (1997). *Fertigungsregelung*. Hanser.

[Wikipedia, 2005] Wikipedia (2005). European train control system. *http://de.wikipedia.org/wiki/European_Train_Control_System.*

[Zimmermann, 2001] Zimmermann, A. (2001). *TimeNet 3.0 — User Manual*. TU Berlin, 3.0.3 edition.

[Zwaneveld et al., 1997] Zwaneveld, P. J., Kroon, L. G., and Romeijn, H. E. (1997). Routing trains through railway stations: Complexity issues. *European Journal of Operational Research*, 98(3):485–498.

[Zwaneveld et al., 1996] Zwaneveld, P. J., Kroon, L. G., Romeijn, H. E., Salomon, M., Dauzère-Pérès, S., Van Hoesel, S. P., and Ambergen, H. W. (1996). Routing trains through railway stations: Model formulation and algorithms. *Transportation Science*, 30(3):181–194.

Curriculum Vitae

Personal Data

Name:	Thomas Michael Herrmann
Date of birth:	July 31st, 1975
Place of birth:	Winterthur, Switzerland
Nationality:	Swiss
Marital status:	Married

Education

1982–1988	Primary School in Winterthur, Switzerland
1988–1994	Secondary School in Winterthur, Switzerland
1994–2000	Student of Mathematics at the Department of Mathematics of the Swiss Federal Institute of Technology (ETH) in Zurich, Switzerland Graduated as dipl. math. ETH
2000–2005	Dissertation at the Institute for Operations Research (Department of Mathematics, ETH Zurich) on capacity and stability measures in railroad processes under the supervision of Prof. Dr. Hans-Jakob Lüthi

Experience

2000–2005	Scientific assistant in research and teaching at the Institute for Operations Research (IFOR), ETH Zurich
2001–2005	Member of IT Support Group at the Institute for Operations Research (IFOR), ETH Zurich
1999	Assistant at the Department of Mathematics, ETH Zurich