Stefan Walter

Long-Term Railway Infrastructure Development

Expansion of the Integrated Timetable on Mixed-Traffic Passenger Railway Networks

DISSERTATION

to achieve the university degree of

Doktor der technischen Wissenschaften

submitted to

Graz University of Technology

Supervisor:

Peter Veit

Institute of Railway Engineering and Transport Economy

Second Supervisor:

Ulrich Weidmann

ETH Zürich

Graz, November 2016

Affidavit

I declare that I have authored this dissertation independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present dissertation.

Date Signature

Acknowledgements

This dissertation was made possible with the help of several people, whom I would like to thank at this opportunity.

First of all, I want to express my gratitude to my supervisors, Peter Veit and Ulrich Weidmann, who spent much time revising, restructuring, commenting, and finally examining this dissertation.

Furthermore, I want to thank Martin Fellendorf, who, although technically not my advisor, guided me all the way from my master's thesis in 2010 to this day and was a helpful support in all demand-related issues.

The idea to compile this thesis was conceived in many intensive discussions with Helmut Uttenthaler, Viktor Borza, János Földiák, Vít Janoš, and Karel Baudyš. Much of the practical experiences in this work is based upon Helmut's master's thesis as well as the practical experiences of all five in creating, evaluating, and putting into practice of Integrated Timetables.

A great portion of this dissertation is made up by methods usually not native to Civil Engineering, but Mathematics and Software Engineering.

Thanks a lot to Severin Kann for the insight into a mathematician's view on the problems of railway engineering, his hints to using SAT and Hitting Sets to solve hub type conflicts, and uncountable discussions about the mathematically adequate presentation of both problems and results. Thanks to Daniel Krenn for the simple, but far-reaching question "*...and what about a third dimension?*". Thanks to Martin Scheidt for serving as a pivot between Traffic Engineering and Mathematics, and bridging many gaps between these disciplines.

Many, many, many thanks, and many more, to Martin Mödlinger for the help with Python. Without him, the conflict diagnosis algorithm would have looked just like an algorithm looks like when designed by a Civil Engineer. With his help, this part now runs with decent running time and also looks beautiful. Thanks also to Jürgen Pammer for his introduction to Matlab and the first version of the three-dimensional visualisation script.

This dissertation was proofread, commented on, and thus enhanced by Martin Scheidt, Christopher Thompson, Stefan Marschnig, Viktor Borza, and Martin Smoliner. Especially Martin Scheidt's proofreading was the most extensive, most exact, and most valuable response to a work I have ever experienced. Thanks also to Martijn Sack for the help with Dutch railway experiences and hyphenation patterns.

The last six years at the Institute of Railway Engineering and Transport Economy at Graz University of Technology have provided me with a comfortable social environment to compile, write, and finally finish this dissertation.

My room mates and closest colleagues, Holger Bach, Anna Maria Frisee, and Martin Smoliner, shared the load of teaching with me, managed to survive the chaos surruounding my way of working, and loosened up the daily routines with both intensive discussions about just anything and a lot of laughter.

Matthias Landgraf and Georg Neuper shared the burden of finishing a dissertation with me, having valuable discussions with me, offering me support and help in difficult times and sharing their experiences in the vast field of bureaucracy with me.

The daily operations at the Institute are managed by Claudia Kaufmann, who not only supported me with all the possible help needed to work at a university, but managed to keep most of the bureaucracy far away from the us to allow for a decent working atmosphere. Many thanks for the humorous and pragmatic approach to daily life!

My colleagues, Manfred Bauer, Armin Berghold, Markus Enzi, Michael Fellinger, Fabian Hansmann, Stefan Marschnig, Johannes Neuhold, and Ivan Vidović, spiced up my life with open and vivid discussions, social distraction, and lots of fun during my time at the institute.

Ich möchte diese Gelegenheit nutzen, mich bei meinen Eltern, Maria und Ulf Walter, dafür zu bedanken, dass sie meinen Forschertrieb stets förderten, mich in allen meinen Entscheidungen unterstützten und nicht zuletzt mit dem grafischen und typografischen Rüstzeug ausstatteten, um diese Arbeit auch optisch zufriedenstellend verfassen zu können.

Abschließend, am wichtigsten, und in schriftlicher Form eigentlich nicht ausreichend darstellbar, möchte ich mich bei meiner Familie bedanken. Meine Frau Verena, mein Sohn Emil und meine Tochter Paula mussten diese letzten Jahre, in denen ich auch zuhause stets tief in die Dissertation versunken war und weder im täglichen Leben noch im Haushalt viel Präsenz zeigen konnte, oft alleine verbringen und die Hochs und insbesondere Tiefs in meiner Stimmung miterleben. Danke, dass ihr mich so unterstützt habt! Danke für die Hilfe! Und ich freue mich schon darauf, endlich wieder mehr Zeit mit euch verbringen zu können.

Abstract

Most transport policy strategies aim to achieve a modal shift towards public transportation. The key instrument for providing attractive public transport is the service offer and thus the timetable. Since any railway timetable needs an adequate infrastructure, the latter must be upgraded to accommodate a timetable. Furthermore, the long service lives of railway infrastructure and the long decision, design, and implementation periods call for long-term infrastructure development strategies.

The Integrated Timetable is a concept for the joint development of railway timetable and infrastructure. This work shall expand the ideas of the Integrated Timetable for networks with mixed traffic also. While the number of approaches to both timetable optimisation and infrastructure capacity analysis are numerous, there has to date been no advancement in approaches to their joint development.

To embed this approach into a consistent design environment, a new design approach was developed: The *Mixed Sequential-Iterative Design Process* allows for an iterative, interdisciplinary design process throughout all technical planning phases, while the phases themselves are sequentially arranged to allow for a transparent decision structure.

The design of a *Service Intention* comprises demand estimation, line planning, and the quantification of the service offer. A line network, the intervals and service concepts per line, and node flows per hub are passed over to the construction of a target timetable.

The key contribution of this work is the creation of this *Target Timetable*. In a three-dimensional network and train graph, different intervals and riding times can be treated simultaneously. First, hub types are classified by the occurrence of hub events. Second, these hub types serve as hinges for the construction of base trajectories per edge and per train system. Third, the hub types are converted to the base interval. With the list of hub types per edge and per train system, a truth function is set up assigning every trajectory a respective hub type combination. The truth function, which is initially unsatisfiable, is then analysed with a SAT-based diagnosis algorithm. This yields several of conflict sets, each denoting which train systems prevent satisfiability and thus a target timetable. With the node flows, the target timetables can be ranked in terms of passenger benefits. These target timetables are then passed over to construct a feasible timetable.

For the construction of a *Feasible Timetable*, the best-ranked target timetable is evaluated against the existing infrastructure condition. Iteratively, the infrastructure is provided with upgrade measures, timetable modifications, and a recalculation of the passenger demand, until the (i) infrastructure is fit to accommodate the timetable and (ii) the timetable is adequate to achieve the target mobility patterns.

To cope with policy changes or project delays, the set of target timetables is arranged in an alikeness graph. This way, a midway strategy shift can be accomplished with only minor timetable and infrastructure changes, thus allowing for both a long-term infrastructure and the possibility to react upon shifts in policy, budget, or demand.

Contents

1 Preface

1.1 Motivation

Railways, as a typical means of mass transport, exhibit an infrastructure expensive in construction and maintenance, while the marginal costs of transport are comparatively low. Furthermore, the most crucial components feature service lives of several decades each, calling for planning horizons of comparable duration.

However, typical periods of validity in transportation policy, the projection of demand, and the knowledge about future mobility development all comprise only a few years. Therefore, the development of railway infrastructure permanently needs to balance the threat of backlogs in upgrade, reinvestment, and maintenance (when setting the focus on the actual scope of running contracts) with the risk of stranded investments (when setting the focus on presumable, but unclear future developments).

When trying to group the major challenges in the development of railway infrastructure, in a long-term perspective, we can cluster them into three groups:

International High-Speed Passenger and High-Capacity Freight Lines: A considerable portion of the network in Europe has been defined to be part of the Trans-European Transport Networks (TEN-T). This implies the need to upgrade the lines in question to comply with Technical Specifications for Interoperability (TSI), touching alignment, signalling, and station design. These lines form the basic grid of Europe's railway network, aiming to offer uniform trans-European infrastructure conditions in an open railway market. All European high-speed railway lines form part of this network. The main challenge, apart from the homogenisation efforts, on these lines is *infrastructure upgrade*, comprising mainly speed increases, inclination decreases and the removal (capacity) bottlenecks, but restricted by budget restrictions, political decision processes, and legal approval routines.

Metropolitan Mass Transport: Around agglomerations, commuter traffic has increased considerably within the last decades and is expected to grow further. Partly on segregated tracks, but also partly combined with lines within the TEN-T network, these lines face a continuously increasing traffic load under increasingly difficult maintenance, reinvestment, and upgrade possibilities. The main challenge in this area is *capacity increase*, comprising node capacity, station layout, and sophisticated signalling systems, but restricted by limited space availability in urban areas.

Regional Transport: The other end of the spectrum in railway transport encompasses lines in remote rural areas, facing declining ridership, abandoned or minimal freight traffic, inadequate infrastructure conditions, and recurring discussions about line closings. These lines face the challenge of *system adequacy*, i. e. the question of when railway transport and thus infrastructure is adequate—and when not.

What all these problem fields have in common is that we need (i) a long-term perspective as described, (ii) a comprehensive knowledge about—at least—the range of demand progression, and (iii) sufficient possibilities to re-scale projects when boundary conditions change.

1.2 Scope

This dissertation shall aim to find ways how to expand the current approaches for long-term timetable and infrastructure development for the creation of a long-term strategy for the joint development of both. While the isolated development of either has received considerable attention in recent years, a thorough methodology for their joint development is unheard of to date.

The goal of this work therefore is to evolve a strategy comprising both timetable and infrastructure development.

The timetable structure is to be (i) abstract enough to allow for the evaluation of long-term target timetables, (ii) robust enough to cope with midway changes of demand or transport policy, and (iii) detailed enough to derive infrastructure upgrade measures from this target timetable.

The infrastructure development strategy aims to (i) offer a detailed description of the target state, (ii) allow for feedbacks about infeasibilities and operational consequences at early planning stages already, and (iii) allow for a disintegration into functionally operable measure bundles.

These two pillars of the design process—timetable and infrastructure—need to be permanently checked against the passenger demand, so as to assess the emerging options against the passenger's reaction and thus the individual project's viability.

These prerequisites shall be adequately embedded within the design process of railway infrastructure and timetables, and be put in the context of the planning and decision processes in this field.

1.3 Objectives

Within this dissertation, the following aspects are to be tackled:

Long-term infrastructure strategy: Due to long infrastructure service lives, it is desirable to provide railway infrastructure managers with a long-term strategy concerning upgrade, reinvestment, and redimensioning. Since railway infrastructure is rarely supplied with sufficient funds for premature network reconstructions, any infrastructure measure needs to be justified for its long-term use and carefully scheduled along a long-term strategy.

Timetable-based infrastructure design: The very purpose of railway infrastructure is to allow for railway operations. So any infrastructure concept needs to be based upon a timetable, rather than vice versa.

Mixed-traffic timetable model: Current methodologies for timetable-based infrastructure upgrade strategies are either based upon manual work or focus on single-purpose networks only, while the vast majority of railway networks comprise mixed-traffic passenger transport. Instead of just one target riding time per edge, we need a set of parameters such as riding times, hub service layouts, and transfer conditions.

Spot and rank infrastructure measures: In order to be able to obtain the target timetable, the existing network needs to be altered. Since there is, on the one hand, a great variety of possible measures to achieve this and there might, on the other hand, be crucial measures without alternative, we need to spot and rank possible measures.

Provide alternatives: Due to the long-term nature of an infrastructure strategy, there might always be intermediate changes to the desired target service offer and/or some planned measures might not be taken. This traces back to a different set of target parameters and thus to a different overall timetable layout.

Track network-wide consequences: Small infrastructure or timetable measures might have great impact on the timetable pattern on one network part and even on the whole network. We need to be able to adjust the overall timetable to account for the altered situation on certain network parts.

2 Fundamentals

In order to set this work in an appropriate framework, these introductory notes shall clarify naming and graphic conventions, definitions and planning principles used throughout this work.

2.1 Symbol, Graphic and Naming Conventions

In this work, the following conventions for naming and graphics shall be used. As some deviate slightly or considerably from other works in this field, they shall be specified here.

2.1.1 Naming and Explaining

For consistency, the following determinations concerning naming and explaining shall apply:

Geographical names shall be written *exclusively* in their native form, i. e. *Vienna* will be written in the German form *Wien*, regardless whether English-language forms exist. This should allow for an easier alignment of the examples presented here with real-life timetable and traffic planning documents.

Explanation of graphics are given outside the graphic, in the text, to maintain a concise structure. Refences are highlighted in colour, so colour print of this work is essential for its understanding.

Symbols are introduced upon first appearance only; a comprehensive list of symbols can be found in the appendix.

Periodically recurring times are written in the form ".xx", i. e. a recurring departure 24 minutes after the full hour will be written as ".24".

2.1.2 Train Graphs

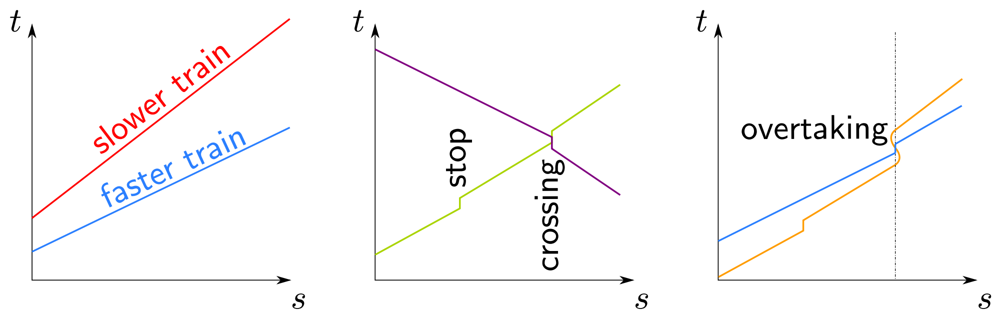
Train graphs will make up a significant portion of the graphical representations in this work. Conventions for train graphs exist to draw (i) vertical time, increasing downwards, and horizontal space, increasing right, or (ii) horizontal time, increasing right, and vertical space, increasing downwards (Pachl 2011: 167f.).

However, in this work, a displaying compatibility with three-dimensional network and timetable graphs (as described in the next section) is desirable. Therefore, all train graphs in this work shall feature, two- and three-dimensionally,

1. *vertical time*, increasing upwards or in z-direction, respectively, and

2. *horizontal space*, either increasing right or depicting the network in the x-y plane.

Figures 2.1a to 2.1c show example train graphs with typical applications. Faster trains therefore feature *less inclined* trajectories than slow ones, crossings and stops manifest in *vertical* trajectories and overtakings take place facing *upwards*.



(a) train speed (b) stopping, crossing (c) overtaking

Figure 2.1: Train graphs as used in this work

For the analysis of hubs, a compacted version of a train graph is used. It only incorporates the hub and its immediate surroundings, depicting all departure and arrival events happening in a hub. Figure 2.2 depicts such a graph. Arrivals and departures are indicated just by stub trajectories and, if necessary, connected through the hub to track continuous train runs.

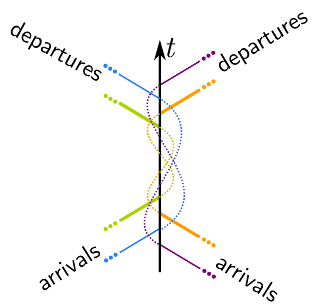


Figure 2.2: Compacted hub train graph

2.1.3 Network Graph

To display the network context of a timetable, a *network graph* (also called *reticular diagram* (Tron et al. 2010) or *schematic line diagram* (PTV Group 2016) represents the major hubs and their timetable attributes (departure, arrival, interval) in a simplified topological view. While not suited for the assessment of riding time possibilities, near-misses of transfers or crossings/overtakings, it provides a useful overview of a system timetable. Figure 2.3 shows the basic conventions in displaying a network graph. Every (logical) line is depicted by a separate (graphical) line, double lines depict half intervals, dashed lines depict double intervals, and dash-dotted lines depict peak-hour or singular rides. Note that the arrival and the departure are distinguishable by their distance to the hubs; the side of the line they are written depends on whether the convention is to denote right-hand or left-hand traffic on the railway network. For lines with double interval, i. e. those depicted in dashed lines, the font style (serifs in italics and underlined, respectively) denotes which time period an arrival/departure event is assigned to.

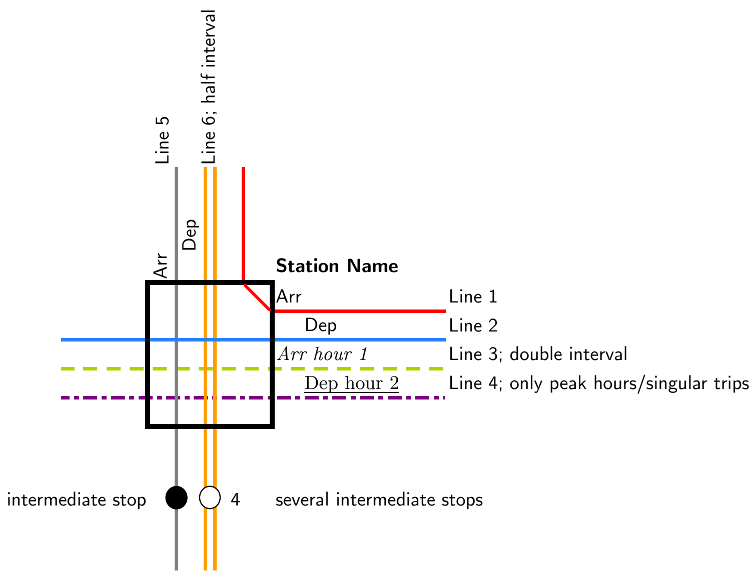


Figure 2.3: Graphic conventions in a network graph

2.1.4 Hub Clocks

In addition to analysing a hub with a compacted hub train graph, a *hub clock* is a common depiction of events in a hub. It typically features a big analogue clock where arrivals and departures are depicted as arrows. Like in the compacted hub train graph, dotted lines show continuous rides. This depiction is suited especially to depict periodically recurring events such as transfers that cross the limits of the basic interval T, i. e. the full hour in this example. Figure 2.4 shows the conventions within a hub clock.

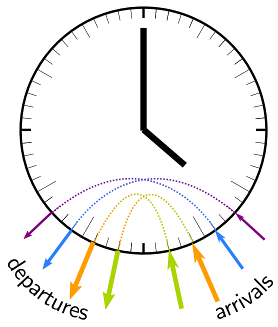


Figure 2.4: Hub clock with departures and arrivals

2.1.5 Three-Dimensional Network and Timetable Graph

For the use of both the network and the timetable, a *three-dimensional train graph* can be used. This graphical representation depicts both a *network view* in two dimensions drawn on the x-y plane, and a *network train graph* spanned on this network plane, tracing the topological routes in x-y direction and the timetable in the z-direction. This graphical depiction approach was first presented by Walter in 2013 and is used as the basis for all further considerations on timetables in this work. For a comprehensive presentation of this approach, refer to Walter 2013.

Figure 2.5 shows a sample three-dimensional network and timetable graph. We can notice four train systems: a regional train from C to A via B that is overtaken in B by an express train on the same route. Together they make B a transfer hub at minute .30. Furthermore, a second regional train connects to this hub at B and continues to D. There, it makes connections with yet another express train that goes to C and, in turn, connects with the first regional train again. As can be seen, the express train does not make connections in either A or C, since it misses the connections there by 15 minutes.

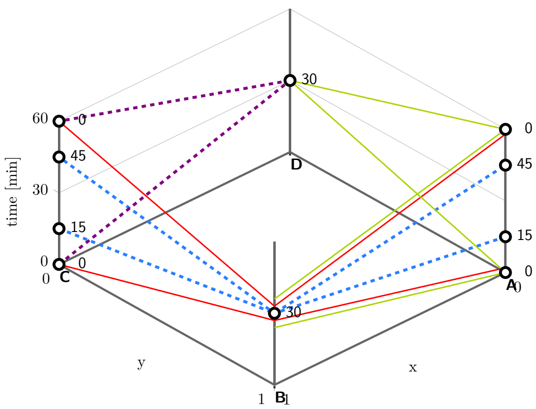


Figure 2.5: Three-dimensional network and timetable graph

2.2 Definitions

2.2.1 System Timetable

When dealing with timetables, it is worth noting that this work's scope strictly remains with *system timetables*, also referred to as *regular hour* (Johnson et al. 2006) or, to a certain extent, *(fully) periodic service intention* (Caimi et al. 2011). As opposed to a *daily timetable*, a system timetable does include all systematic train runs in a pattern repeating every interval T, while a daily timetable contains all rides throughout a day, including peak-hour deviations, off-peak service reductions and deviations from the system. Caimi et al. criticise that a system timetable (i) does not depict the temporal variations of a timetable adequately and (ii) that alternate timetables constructed just with a mere reduction of paths within a system timetable may offer a worse service during off-peak hours (Caimi et al. 2011).

The former point of criticism is definitely the case whenever the design of a timetable reaches close planning horizons, such as mid-term planning or tactical planning (see section 4.1). However, the scope of this work is long-term design of railway infrastructure, which is why minor temporal changes in the service offer are not relevant at this design stage. However—tackling also the second item of criticism—major temporal variations within a timetable must be taken into account upon construction of a system timetable, although a different approach than by Caimi et al. is used within this work. Refer to section 2.3.6 dedicated to this issue.

2.2.2 Periodic, Symmetric, and Integrated Timetable

For a systematic classification of timetable types, we shall use the methodology of Liebchen. In his works, periodic timetables are classified along the degrees of freedom present upon timetable creation. Apart from individually planned train runs, periodic timetables are classfied in three categories. *Periodic Timetables* denote such timetables that feature constant intervals between events on a line over a longer period of time (usually at least several hours, mostly throughout a day or a part of the day). In *Symmetric Periodic Timetables*, the train trajectories in opposite directions are bound together around the axis of symmetry (see section 2.3.3), i. e. the degree of freedom to choose a time slot for either direction is lost. Finally, *Integrated Fixed-Interval Timetables*, which will be referred to as Integrated Timetable (ITF) in this work, also lack the degree of freedom to choose any slot for any train, even if operationally feasible. As the reader presumably knows (and as described in section 3.3), the riding times in Integrated Timetables follow the location of timetable hubs, and only then feature some kind of freedom in slot construction (Liebchen 2005, Liebchen 2006).

2.2.3 Riding Time

Riding time calculation in railways does already encompass a remarkable effort when calculating the technical riding time, but for the Integrated Timetable, an even more complex approach to riding time is necessary. For conventional riding time calculation, refer to Pachl 2011: 31.

An Integrated Timetable derives its target riding times from the definition of edges and their respective riding times. As noted already, the hubs are to be spaced at riding times of $n \cdot T/2$. However, this riding time is calculated between the axes of symmetry in a hub, so it also comprises the *proportional hub stop time* $t_{h,\text{prop}}$, i. e. the portion of the hub stop time oriented towards the edge when viewed from the axis of symmetry. Since we follow an approach of *target riding times* in this work (rather than calculating existing technical running times and deriving a timetable from there), it is more relevant to calculate the required technical running time $t_{r,\text{net}}$ *backwards* from the edge riding time (Uttenthaler 2010: 36f. Pachl 2011: 29f.).

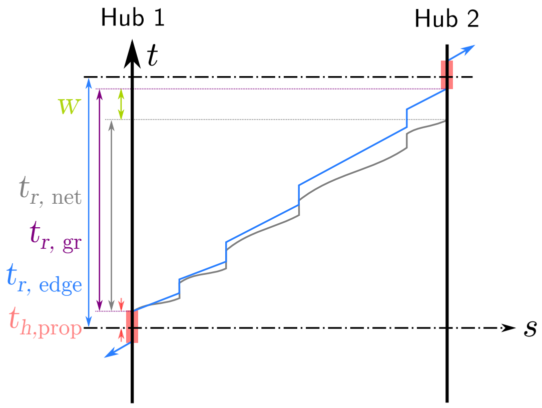


Figure 2.6: Riding time relations for an Integrated Timetable

Figure 2.6 shows the time relations required for an Integrated Timetable. As can be seen, the target technical riding time can be calculated from the edge riding time with the help of the *gross riding time* $t_{r,\text{gr}}$, which incorporates the recovery time w.

$$t_{r,\text{net}} = \frac{t_{r,\text{gr}}}{1+w} \tag{2.1}$$

$$t_{r,\text{gr}} = t_{r,\text{edge}} - t_{h,\text{prop},1} - t_{h,\text{prop},2} \tag{2.2}$$

$$t_{r,\text{edge}} = n \cdot \frac{T}{2} \tag{2.3}$$

2.2.4 Train Types

Classically, trains could be roughly split into fast trains, express trains and local trains, with possibly some more differentiation over the riding time. However, the mere *train type name*, such as Intercity (IC), Railjet (RJ), and the like, does not automatically imply a certain network-wide train system behaviour.

train type	RJ	ICE
Wien Hbf	12.30	12.50
W. Meidling	12.37	12.57
St. Pölten	12.00	13.22
Linz Hbf	13.46	14.12

(a)

train type	RJ	EC
Wien Hbf	13.42	14.42
Hegyesh.	14.28	15.28
Győr	14.50	15.50
Budapest k.	16.19	17.19

(b)

train type	EC	RJ	RJ
Wien Hbf	15.58	16.25	16.58
W. Meidling	16.05	16.32	17.05
Wr. Neust.	16.32	16.57	17.32
Mürzzuschl.	17.30	|	18.30
Kapfenb.	17.52	|	18.52
Bruck/M.	17.58	18.15	18.58
Graz Hbf	18.33	|	19.33
Villach		20.46	

(c)

train type	RE	IC
Bremen	10.53	11.53
Delmenhorst	11.03	12.04
Hude	11.12	12.13
Oldenbg.	11.33	12.33
Bad Zwisch.	11.44	12.44
Westerst.-O.	11.51	12.51
Augustfehn	11.58	12.59
Leer	12.24	13.22
Emden Hbf	12.42	13.42
Marienhafe	12.58	13.59
Norden	13.06	14.08
Norddeich	13.12	14.14
Nordd. Mole	13.16	14.20

(d)

train type	EC	ICE	IC	IR	IR	IC	TGV
Basel SBB	7.47	9.07	9.33	9.37	9.47	10.07	10.33
Rheinfelden	|	|	|	9.49	|	|	|
Frick	|	|	|	10.03	|	|	|
Liestal	7.57	|	|	|	9.57	|	|
Sissach	8.03	|	|	|	10.03	|	|
Aarau	8.24	|	|	|	10.24	|	|
Lenzburg	8.31	|	|	|	10.31	|	|
Brugg AG	|	|	|	10.19	|	|	|
Baden	|	|	|	10.29	|	|	|
Dietikon	|	|	|	10.37	|	|	|
Z. Altstetten	|	|	|	10.44	|	|	|
Zürich HB	8.52	10.00	10.26	10.49	10.52	11.00	11.26

(e)

Table 2.1: Relation between train type, stopping patterns, and riding times, 2016 timetables.

Tables 2.1a to 2.1e show the relations between train type, stopping patterns, and riding times for example lines. Table 2.1a shows a timetable excerpt between *Wien* and *Linz*, where Intercity Express (ICE) trains, normally the fastest train types, run in a slower slot than RJ trains. However, in table 2.1b, the RJ slot—served by the same RJ line as the last example—is identical to the Eurocity (EC) slot between *Wien* and *Budapest*. Furthermore, table 2.1c shows the stretch between *Wien* and *Graz/Villach*, where there are two RJ slot types that differ significantly in riding time and stopping pattern, while the slower slot is identical to the EC slot. On the *Emsland* railway line between *Bremen* and *Norddeich Mole* (see table 2.1d, a Regio-Express (RE) line and an IC line part each other's intervals to jointly offer an hourly service. As can be seen, the long-distance train is even a little slower than the regional train on this stretch.

Finally, table 2.1e shows a complex example on the line between *Basel* and *Zürich*: While EC trains run in the same slot as Interregio (IR) trains, there is a second type of IR slot 7 minutes slower. Furthermore, there are ICE, Train à Grande Vitesse (TGV), and IC trains in identical slots.

Within this work, *train types* always refer to a set of train types that is (i) as small as possible, so as to describe train types adequately, without taking into account train types purely used for marketing purposes; (ii) as large as necessary, so as to reflect the number of different riding time profiles that occur within the network, and (iii) as systematic as possible, so as to group train systems by riding speed rather than hierarchical position or, as already mentioned, marketing issues.

2.3 Planning Principles

For the design methodology presented in this work, some planning principles shall be introduced that will be built upon throughout this work.

2.3.1 Target Timetable

A *target timetable* is a target state, i. e. a timetable the current network needs to be designed upon. Target timetables stem from the basic principle of the ITF to upgrade the infrastructure to accommodate a timetable, rather than vice versa. A *target timetable* is, in most cases, *infeasible* on the current network, since it can usually not be offered without infrastructure modification. The railway strategies of Austria, Switzerland, and Belgium all encompass such a target timetable, depicting the target state for both timetable and infrastructure (ÖBB 2011, BAV 2016, Geerts 2013).

Note that this does *not* imply the target state has been purely defined by sophisticated infrastructure design processes as described in section 4.1; it is rather the case that many features of the named target timetables are (i) derived from *educated guesses*, (Scheidt 2016), (ii) based upon political wishes or (iii) the result of a restrospective justification of individually planned infrastructure projects.

However, these target timetables provide a helpful framework for the future development of timetable and infrastructure.

2.3.2 Trajectory Simplification

In the Target Timetable Phase (see chapter 5), we will use a simplified trajectory model for riding time calculation and further processing. All constituent parts of the riding time (see section 2.2.3), all intermediate stops, proportional hub stop time, and recovery times are added up and then used together to create one, straight trajectory, as shown in figure 2.7. The actual riding time, denoted in black, is approximated by a straight line, denoted in red.

The planning and effetuation phases covered in this work are long, so this simplification is a valid approach (See section 4.1 and Caimi 2009: 14), since the variations to be covered throughout implementation are still large.

We make intensive use of this simplification when assessing crossings, overtakings, and riding speeds. The approach is held as far as possible, so as to retain its simplicity, and is only replaced by exact riding time calculation in the Feasible Timetable Phase.

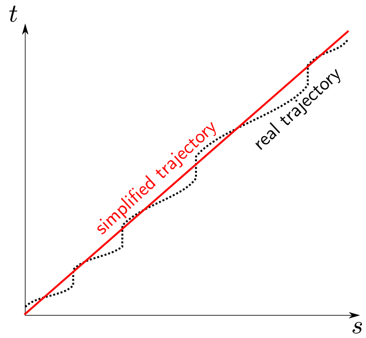


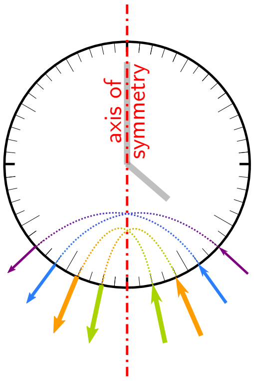
Figure 2.7: Simplified trajectories as used in the Target Timetable Phase

2.3.3 Timetable Symmetry

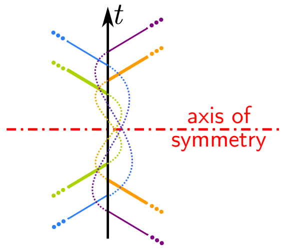
Timetable symmetry is one key issue of an ITF, since it drastically reduces computing and construction effort. Figures 2.8a and 2.8b show the axis of symmetry highlighted in both hub clock and train graph view. Note that "symmetry" does *not* imply that the arrival and the departure of a continuous train ride are symmetrical to the axis of symmetry. It rather implies that *events* are mirrored along the axis of symmetry; an *arrival* from one direction therefore mirrors a *departure* back into that direction. The dotted lines in figures 2.8a and 2.8b reflect that.

Basically, the axis of symmetry in any periodic timetable is the point in time when train runs of opposite directions cross each other; therefore any periodic timetable does have an axis of symmetry. However, in the context of this work, we consider a network *symmetric* if the axis of symmetry is common throughout a network. The reason for this requirement is that only network-wide symmetry allows for a systematic transfer design: In a symmetric environment, transfers work either in *both* directions or in *none*.

Nevertheless, slight deviations of the axis of symmetry are common, especially to account for varying grades of reliability: In the Austrian province of Vorarlberg, bus timetables with rail connections feature a slightly earlier axis of symmetry than the trains. This way, passengers find tight connections from train to bus, but looser connections vice versa, to make up for the poorer reliability of road-bound public transport. Table 2.2 shows the



(a) in a hub clock



(b) in a train graph

Figure 2.8: Axis of symmetry

example of *Dornbirn*: from the train to the bus, the transfer time is 4–5 min, but vice versa it is 6–7 min. This way, delayed buses possess a larger buffer to make up for traffic irregularities. Larger deviations of symmetry, however, render transfers in one direction either impossible or remarkably longer, since a one-minute shift in symmetry results in a two-minute shift in the transfer relations. Tables 2.3a to 2.3c show examples where significant shifts lead to transfers broken in one direction.

Dornbirn Bf	↓	↑	axis of symmetry
S1 from/to Bregenz	.29	.29	.59
S1 from/to Bludenz	.28	.30	
transfer time	*4–5 min*	*6–7 min*	
40 to/from Schoppernau	.33	.23	.58

Table 2.2: Different axes of symmetry for reliable transfers at *Dornbirn*, 2016 timetable.

A one-directional prioritisation of a transfer can be used to account for asymmetric demand. While this is seldom the case throughout a whole day, typical commuter routes do feature a strong direction of load. In cases where a symmetric transfer is not possible due to infrastructure infeasibilities or other boundary conditions, this can be overcome by providing two versions of timetables with complemental axes of symmetry that change direction of transfers during the day. However, passengers travelling against the direction of load face a non-transfer in any case. Therefore, as soon as there is a considerable amount of demand against the direction of load, this solution is a compromise rather than a target state.

Figure 2.9 shows such a case: a single-track line from A to B is to be attached to a transfer hub at A. The existing infrastructure allows for crossing in C only and a riding time so long that neither a round trip A–B–A is possible within T nor a solution of crossing at C and serving hub A adequately. Therefore, the timetable is shifted during the day. The

Graz Jakominiplatz	↓	↑	axis of symmetry
32 from/to Seiersberg	.06	.10	.00 $^1/_2$
transfer time	*4 min*	*10 min*	
1 to/from Mariatrost	.10	.00	.57 $^1/_2$

(a)

Salzburg Hbf	↓	↑	axis of symmetry
S3 from/to Bad Reichenhall	.49	.12	.00 $^1/_2$
transfer time	*11 min*	*27 min*	
S1 to/from Lamprechtshausen	.00	.45	.52 $^1/_2$

(b)

Werndorf Bf	↓	↑	axis of symmetry
S5 from/to Spielfeld-Straß	.08	.59	.03 $^1/_2$
transfer time	*6 min*	*23 min*	
S6 to/from Wies-Eibiswald	.14	.36	.55

(c)

Table 2.3: Unidirectionally broken transfers from different axes of symmetry, 2016 timetables (Graz: Sunday timetable).

transfer towards B is prioritised in the first period and the transfer originating in B in the second. Note that the respective other direction does not see a transfer at A. As can be seen, as soon as there is a considerable demand against the direction of load, this solution creates an unattractive transfer layout. If the infrastructure were upgraded to either allow for a crossing at minute .00 (about halfway between C and B) or to allow for a round trip within a riding time of $t_{r,\text{round trip}} = T$, a symmetrical transfer could be offered.

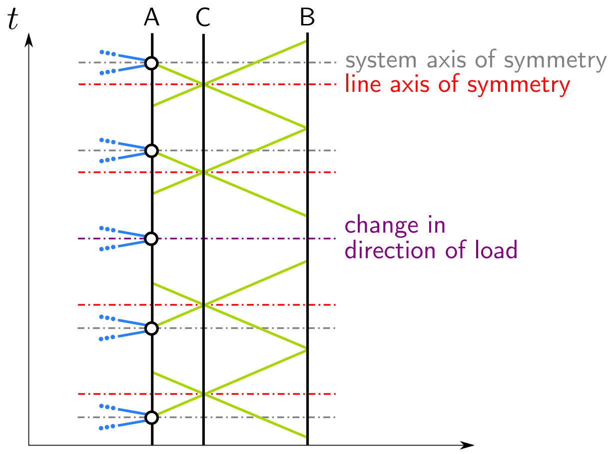


Figure 2.9: Deviant axis of symmetry for infrastructural reasons

A solution where symmetry is shifted the same way throughout the day, i. e. one transfer direction is systematically disadvantaged, would require a permanently asymmetric demand to be justified. This is, in fact, the case for many classical activity chains (such as home–work–shopping–home). However, to receive an asymmetric accumulation of these chains in the context of a regional railway network, this would require *functionally separated* hubs. This, in turn, would mean that settlements need to be separated functionally in such a strict way that the *absolute majority* of trips within a transfer hub runs asymmetrically and *always in the same direction.* This does exist on a microscopic scale[2], but is out of scope for the typical scales of railway networks.

Finally, there is one situation in which an alternating axis of symmetry can be used implicitly without worsening the transfer conditions: When a (branch) line with a longer interval T_{branch} is to connect to a main line with shorter interval T_{main} (e. g. $T_{\text{branch}} = 2 \cdot T_{\text{main}}$), the axis of symmetry for the branch line can be chosen to be *one* of the axes of the main line, not just the one closer to .00. This can be helpful when connecting two branch lines to one main line with fewer vehicles. This solution must be limited to a sub-network where *all* the lines with longer intervals run at the same,

[2]Consider classical, functionally separated neighbourhoods in suburbs built around the 1970s: Commuters go to work in the Central Business District (CBD), and go shopping or for leisure on their way home, creating unidirectional, circular activity chains.

deviant, axis of symmetry. Otherwise, any contact to other lines with longer intervals, but usual axes of symmetry would lead to lost transfers in one direction.

Figure 2.10 shows lines 24 and 45 of *Regionalbus Unterland* in the Austrian province of Vorarlberg. Both offer alternative routes from the Rhine valley (*Schwarzach, Wolfurt*) to *Alberschwende*. To serve the full hub at minute .00 in *Alberschwende*, the buses, each at interval $T_i = 120\,\text{min}$, feature an axis of symmetry at minute .30 rather than .00. This way, every other hub time is served by an arriving bus of one line and a departing bus of the other. This way, one vehicle can serve both lines. If every hub time were served by the same line arriving *and* departing, i. e. if the axis of symmetry was at minute .00, one vehicle each would need to wait for over 90 minutes at *Wolfurt* and *Schwarzach*, respectively.

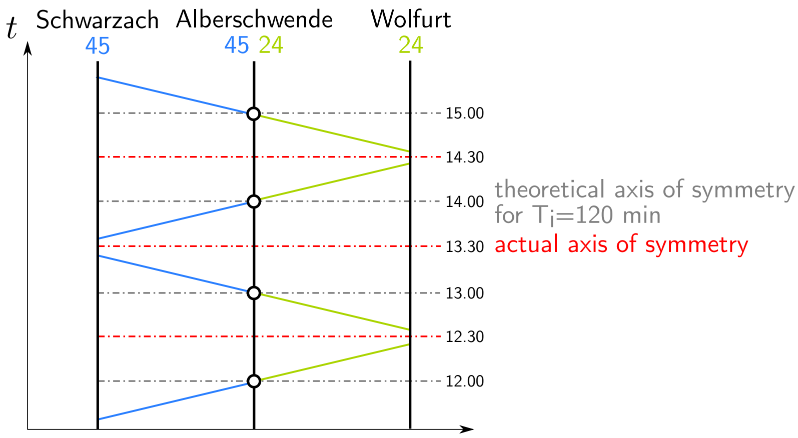


Figure 2.10: Deviant axis of symmetry for optimised vehicle circulation

It must be noted that there is vivid criticism on timetable symmetry (see, at least, Liebchen 2008: 366, Liebchen 2004, Liebchen and R. H. Möhring 2007: 24, and Liebchen 2005). The focus of the criticism can be summarised as follows: There are situations where the infrastructural situation does allow good transfer connections in one direction only, and a symmetrical solution would worsen even the one direction. For almost *any* railway network, examples can be found where this property holds. Timetable symmetry, therefore, must be questioned in terms of its feasibility for a *timetable design* approach.

However, the focus of this work is *not* to find a timetable that best suits the existing infrastructure (see section 3.1 for state-of-the-art timetable construction approaches), but on finding future timetables that best meet the demand and deriving the infrastructure needs from there.

2.3.4 Mixed Passenger Traffic

Mixed Traffic in this context shall denote mixed *passenger* traffic only. This approach does not directly incorporate freight traffic, but some thoughts on freight traffic shall follow in the next section and, finally, in section 8.3.

Single-purpose railway lines or, more generally, lines of public transport that feature one single public transport system at a time, can be characterised as a system with one riding time t_r and one interval T along the line. Subsequent trips on the same line will neither vary in their time lag nor in their order, nor in their riding speed. Therefore, they can be modelled by simply assuming a uniform interval and a uniform riding speed (and therefore riding time).

Any event along the line (departure, arrival, crossings etc.) will always happen at the same time (mod T) and any event in the opposite direction can be derived by mirroring it along the axis of symmetry.

Systems with mixed traffic, however, feature several additional problems that stem from the occurrence of more than one riding time and more than one interval.

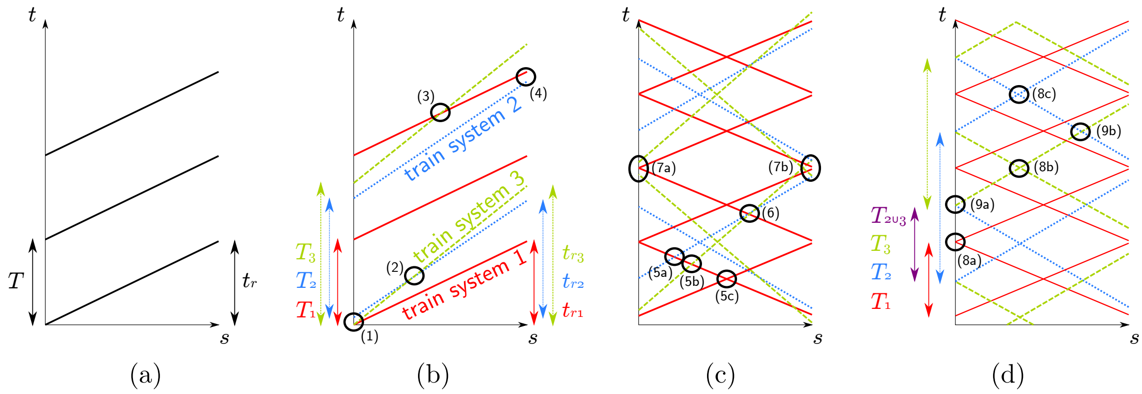


Figure 2.11: additional complexity in modelling mixed-traffic lines

Figure 2.11a depicts the base case in single-purpose lines with one interval T and one riding time t_r only.

Figure 2.11b shows a system with three different train types in one direction: Additional complexity is added by (1) simultaneous exits and entries at stations, (2–3) overtakings between different systems at different points along the line, and (4) fast trains closing up on slower trains.

Figure 2.11c expands the problem to a two-directional point of view: (5a–c) Crossings between opposite trains occur at different points along the track, which is especially important at single-track railway lines; (6) crossings and overtakings can accumulate at the same point, requiring complex track structures; and finally, (7a–b) integrated timetable hubs can develop to comprise complex structures.

Figure 2.11d, finally, shows a situation where two train systems are shifted towards each other to jointly run at half the interval. This way, not only do classical timetable hubs (8a–c) appear, but also two *semi hubs* (9a–b) are combined to form one full hub.

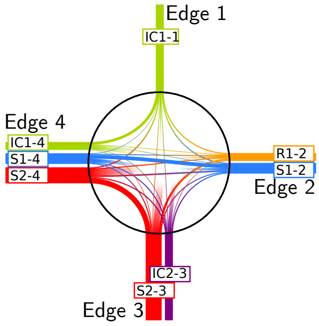
2.3.5 Integration of Demand Modelling

In the core portion of this work, *static node flows* are used to incorporate demand modelling. From a headway-based assignment upon the creation of a service intention, the demand model yields node flows q per hub, i. e. an Origin-Destination (OD) matrix between all train systems serving a station, including flows to and from the station itself. Figure 2.12a shows a sample node flow and table 2.12b shows the corresponding OD matrix. From this information, we can directly retrieve a prioritisation of transfer relations, which we need when it comes to conflict resolution in section 5.4.6. Note that, by the nature of the Target Timetable Phase as described in sections 4.5 and 5.4.6, the node flows used here are *static*. The *dynamic* network-wide demand calculation happens before, in the Service Intention Phase (see section 4.5), and afterwards, in the Feasible Timetable Phase (see section 6.3.2). Static node flows pose the disadvantage that a rerouting via alternative routes is considered not to take place. In urban networks, this is a drawback that needs to be handled by setting breakpoints upon resolution of *hub type conflicts* (see section 5.4.6) at the latest, i. e. when transfers are potentially broken. In regional and national networks, which this work covers primarily, the danger of rerouting is comparatively small given the large mesh size we are dealing with. Since a complete traffic assignment would be necessary to receive modified node flows in each network redesign step, a dynamic demand in this stage is considered to cause too much extra complexity, without relevant additional benefit for the application in regional and national networks.

Note that, when referring to *lost transfers* in this context, we are dealing with transfers lost *by design*, i. e. planned lost transfers. Transfers lost due to reliability issues are out of scope for this work.

For the use in this work, the node flows are to be weighted. This is to account for (i) the different average travel distances s_r found in different train types and (ii) the different intervals T_i.

Consider the node flow as described in table 2.12b. If no further distinction was made, trains with a large number of passengers would be prioritised over those with fewer passengers irrespective of their travel distance. Long-distance trains might therefore face a comparatively poor priority even though the passengers on board travel large distances and therefore account for a larger portion of the transport volume. While the goal of this



(a) Node flows

	IC1-1	IC1-4	S1-4	S2-4	S2-3	IC2-3	S1-2	R1-2	Hub
IC1-1		5,000	1,000	2,000	1,000	1,000	2,000	2,000	2,500
IC1-4			–	–	1,000	1,000	1,000	1,000	7,000
S1-4				–	2,000	2,000	7,000	2,000	6,500
S2-4					10,000	2,000	3,000	2,000	12,000
S2-3						–	2,000	4,000	12,000
IC2-3							2,000	1,000	7,000
S1-2								–	2,000
R1-2									2,000

(b) OD matrix

Figure 2.12: Demand output from Service Intention Phase. "Hub" denotes passengers bound for the hub itself.

work is strictly *not* to prioritise long-distance trains over regional trains in the design process just by their hierarchy, a prioritisation of passenger travel distance does take into account their share in revenues.

Ahrens et al., Bussieck as well as the German Bundesnetzagentur evaluated the average distance travelled in different train systems in Germany. However, none of these figures incorporates an evaluation across all train systems. For Austria, we can only estimate these results with the help of Brezina et al., the travel time distributions from Ahrens et al. and the evaluation of riding speeds per train system in section 7.1.3. Table 2.4 summarises the findings. The estimate approximately aligns with the German values, but does account for the generally shorter distances in Austria. We can therefore deduct a rough estimate on how to weight train systems by their travel distance. We introduce a relative weight $k_{sr} = {}^{s_{r,i}}/_{s_{r,\max}}$ to keep the values $0 \leq k_{sr} \leq 1$.

Furthermore, the interval of the respective train systems also has an influence on the graveness of a potentially lost transfer. An interval can simply be described by its divisor $\nu_i = {}^{T_i}/_{T}$. This directly corresponds with the weight of a connection in relation to all trains, so xthere is no need for a modified parameter in this context.

This information can be combined into a *weighted node flow* to incorporate both travel distance and interval. Since node flows incorporate demand between different train types, an estimation of the influence of either parameter onto the transfer process must be carried out. However, at this point and considering the estimative nature of the data, a selection of the *greater* travel distance and the *shorter* interval (i. e. the *greater* value of ν_i) per transfer[3] shall suffice. The weighted node flow q_{mod} can therefore be expressed in the form

[3] We select the greater travel distance to account for long-distance passengers using regional transport as a feeder; we select the shorter interval since the maximum additional transfer time in case of a—planned—lost transfer equals the shorter interval.

train type	abbrev.	s_r[km]	k_{sr}
Germany, Ahrens et al.			
S-Bahn	(S)	24	
Regional trains	(R)	41	
Germany, Wittwer			
Long-distance trains	(IC/ICE)	183	
Germany, Bussieck			
Intercity	(IC)	200	
Interregio	(IR)	120	
Germany, Bundesnetzagentur			
Regional trains	(R)	21	
Long-distance trains	(IC/ICE)	280	
Austria, estimate			
S-Bahn	(S)	24	0.14
Regional trains	(R)	28	0.16
Express trains	(REX)	36	0.20
Intercity	(IC)	118	0.67
Railjet	(RJ)	176	1.00

Table 2.4: Average travel distance s_r and relative weight k_{sr} by train system (aggregated from Ahrens et al. 2015a, Ahrens et al. 2015b, Ahrens et al. 2015c, Ahrens et al. 2015d, Ahrens et al. 2015e, Ahrens et al. 2015f, Ahrens et al. 2015g, Wittwer 2016a, Wittwer 2016b, Bussieck 1998, Bundesnetzagentur 2016b, Bundesnetzagentur 2016a, Brezina et al. 2014)

$$q_{\mathrm{mod}} = q \cdot k_{sr,\max} \cdot {}^{1}\!/\!_{\nu_{i,\max}} \tag{2.4}$$

With the values of ν_i stated in table 2.5a, we can modify the node flows to consist of q_{mod} only, as stated in table 2.5b.

train system	IC1-1	IC1-4	S1-4	S2-4	S2-3	IC2-3	S1-2	R1-2
k_{sr}	0.67	0.67	0.14	0.14	0.14	0.67	0.14	0.16
$1/\nu_i$	1	1	0.25	0.25	0.25	1	0.25	0.5

(a) List of k_{sr} and ν_i for sample node flows

	IC1-1	IC1-4	S1-4	S2-4	S2-3	IC2-3	S1-2	R1-2	Hub
IC1-1		3,350	168	335	168	670	335	670	1,675
IC1-4			–	–	168	670	168	225	4,690
S1-4				–	70	335	245	80	228
S2-4					250	335	105	70	420
S2-3						–	335	80	4,690
IC2-3							335	80	4,690
S1-2								–	70
R1-2									160

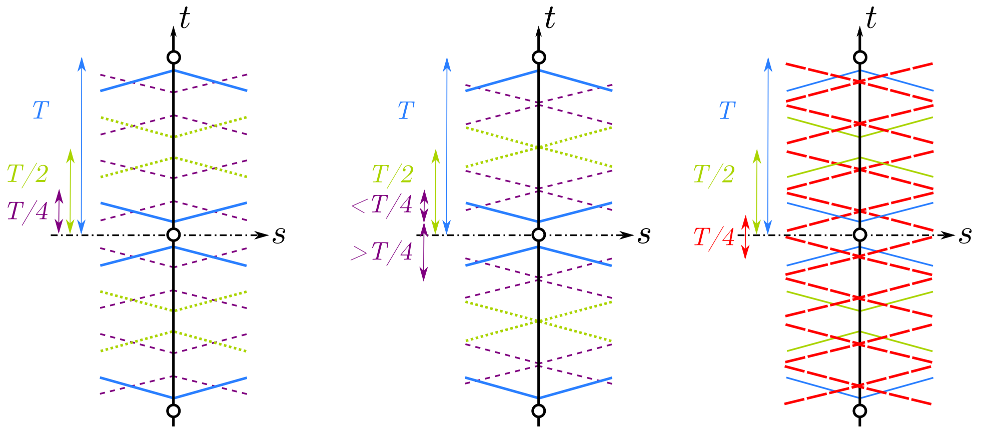
(b) Modified sample node flows

Table 2.5: Node flow modification

2.3.6 Temporal Timetable Variations

Public transport networks usually feature a considerable change in both route network and timetable during the day, the week, and the year. We shall only focus on the question of service changes during the day and during the week, i. e. a rough subdivision of timetables into *peak*, *daytime off-peak*, *evening off-peak* and *weekend off-peak* (Schwager 2003: 3-67). The exact subdivision in this context is not relevant, since from a long-term infrastructure and timetable development point of view, the only relevance is how big a difference between different timetables has to be taken into account.

In railway timetabling, a common approach to create a timetable for both peak and off-peak hours is what Liebchen and R. H. Möhring call *balanced service reduction*. In this planning paradigm, every line features a basic interval trajectory that runs throughout the day, with further rides added into the gaps to subsequently halve the intervals for increased daytime and peak services (Liebchen and R. H. Möhring 2007: 21). When inserted into a network context of long-distance trains that do *not* usually feature a significant service variation (i. e. services are offered throughout the day and the week), the reason for this becomes obvious: in order to deliver an adequate link with the invariant long-distance network, one of a similar nature is built up local traffic and then extended to denser versions.



(a) regular interval (b) reduced stopping time (c) separate timetables

Figure 2.13: Balanced reduction of service

Figure 2.13a depicts a situation where a train system features a basic trajectory at base interval $T_i = T$ with subsequent extension to $T_i = {}^T\!/2$ and $T_i = {}^T\!/4$. As can be seen, the basic trajectories have a longer stop at the central station to account for the transfers in the timetable hub. If the additional trips halving the intervals are inserted to exactly halve the basic interval, they consequently also feature a long stop at the central station. So do the peak hour trips that halve the daytime interval again, such that finally long stops without a purpose outnumber those actually stopping for transfers.

Figure 2.13b shows an alternative: When the additional trips pass through the central station without extra stopping time, the problem just addressed is solved, but the interval becomes irregular, with three trips spaced at ${}^T\!/4$ and one with an offset of half the additional stopping time.

Finally, figure 2.13c shows a solution where the trips for $T_i = T$ and $T_i = {}^T\!/2$ are constucted like in figure 2.13a, but a different timetable is offered for $T_i = {}^T\!/4$, shifting all trajectories in peak hours by ${}^T\!/8$. The additional benefits of such timetables will be covered in section 5.1.3, but in this context it is important to note that certain interval densities call for a separate timetable rather than a balanced service reduction approach.

Another approach to cope with both maintaining the hub structure and achieving attractive riding times in peak hours is to use the time otherwise spent stopping to continue past the hub. When a denser interval in peak hours is required around the hubs, but not necessarily along the open track, the basic timetable structure can be retained, but the way it is served differs by the hours of service.

Figure 2.14 shows a situation where the basic trajectories for $T_i = T$ and $T_i = {}^T\!/2$ are arranged just like in figure 2.13a. However, the step towards $T_i = {}^T\!/4$ is taken by a reorganisation of train runs: runs from the basic trajectories and the additional trips

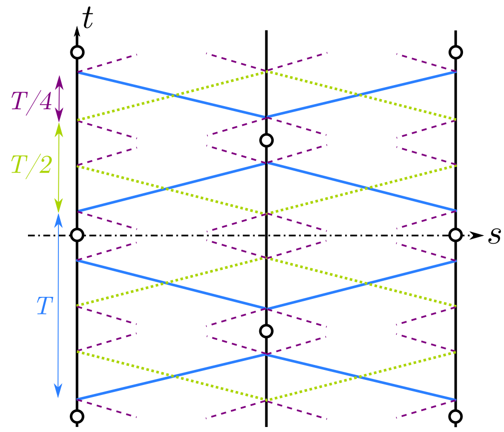


Figure 2.14: Joined basic and peak hour trips

continue onwards on the paths of the peak hour trips to create a denser interval. Likewise, peak hour trips start before the hubs and directly feed the basic trajectories and the additional trips departing from the hub. This way, the disadvantage of long stopping times can be alleviated at least during the peak hours. Note that, for the example of $T = 60$ min, the minimum stopping time in a hub during off-peak hours amounts to 15 min in this approach, i. e. all trips arrive and depart 7.5 min from the axis of symmetry *or* the additional trips need to run at slightly shifted intervals.

Finally, one method of dealing with a varying service offer is to introduce proper lines or line segments for peak hours. This way, it is not the timetable of lines that is changed, but the number of lines. As long as the network is easily conceivable by passengers, this offers a possibility to both offer a stable framework service throughout the week, but also to react upon increased demand without the problems sketched before. The threshold for the network size cannot be directly obtained by network parameters, but rather must be individually evaluated. However, the number of networks with this approach has been decreasing after a rise in the 1990s (Sparmann 2006).

The bus network of Liechtenstein is one of the networks where this approach is used. It features four main bus lines that overlap on several stretches. Changes in service offer are accomplished by adding or subtracting lines or line segments.

Figure 2.15 shows the network and timetable graph for the main lines. The basic offer is drawn in thick, continuous lines; the off-peak service on weekdays is drawn in thinner, dashed lines; and the additional peak-hour trips are drawn in thin, dotted lines. The backbone of the network is line 11, serving the whole route from *Feldkirch* to *Sargans* half-hourly daily and from 5 am to midnight without deviations. On weekdays, its interval is halved by line 14 in the west and line 13 in the east. Both lines 14 and 13 also serve a central stretch in the basic offer as well (*Feldkirch–Tisis* and *Eschen–Nendeln–Schaan*, respectively) and have their trips extended during daytime. Additionally, the trips are extended even further during peak hours. Line 12, offering a basic service between *Buchs*

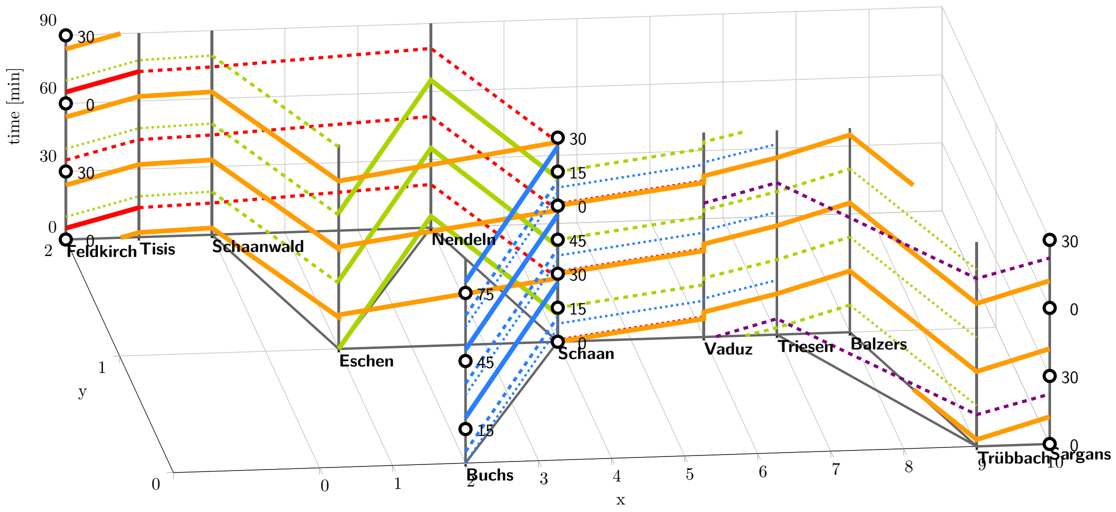


Figure 2.15: Network and timetable graph of the main lines in Liechtenstein's bus system, 2016 timetable.

and *Schaan*, also extends its peak-hour trips to *Triesen*, offering in total six trips per hour between *Schaan* and *Triesen.* Finally, line 12E connects *Vaduz* to the timetable hub at *Sargans* during daytime.

All main lines meet in *Schaan* to achieve a full timetable hub, working every hour in the basic offer and half-hourly during the day.

2.3.7 Freight Traffic

As noted, freight traffic is explicitly *not* tackled in this work. In most planning frameworks, most recently presented by Nachtigall, Noll, et al., periodic passenger traffic is considered first and then a set of freight traffic *catalogue slots* is set into this timetable (Nachtigall, Noll, et al. 2014). Freight traffic features (i) great variations in train dynamics, (ii) a significantly worse level of on-time performance than passenger traffic and (iii) little intrinsic (i. e. economic) motivation for periodicity, such that an approach as presented here is bound to fail if applied directly. However, the *catalogue slot* approach can be used to provide an interface, taking into account the prerequisites of either type of traffic (Pöhle et al. 2012). Section 8.3 features some thoughts on the integration of freight traffic in this framework.

3 State of the Art

In order to thoroughly elaborate on the contribution of this work, a survey of the status quo in research and application is useful. Since this work aims to develop a method for the construction of a timetable-based infrastructure development, the survey shall be divided into (i) timetable construction approaches (see section 3.1), (ii) infrastructure development approaches (see section 3.2) and (iii) the Integrated Timetable as a combined approach to jointly develop timetable and infrastructure (see section 3.3).

3.1 Timetable Construction Approaches

Timetable construction approaches deal, at large, with the development of timetables on *existing* infrastructure. Apart from manual timetable construction strategies[4], the approaches to constructing timetables have meanwhile grown to an uncountable number of competing and consecutive models, of which the main philosophies shall be presented here in brief.

3.1.1 Periodic Event Scheduling Problem

The Periodic Event Scheduling Problem (PESP) is perhaps the most used and most intensively developed approach to railway timetable construction. Developed from an approach to optimise traffic lights (Serafini et al. 1989) and adapted for railways (Kroon et al. 2004, Liebchen 2006), it has since developed to a large set of different enhancements. Refer to Cacchiani et al. 2012 for a comprehensive review on this field.

The basic concept of PESP is to model a railway network as a graph where *events* form the vertices; the edges are used to connect these events and can take, at least, the form of *traveling edges* to denote actual trips between events, *transfer edges* to denote passenger transfers, *standing edges* to denote dwell times, and *headway* edges to model route conflicts. A base interval T is used in form of the modulo operator to model the periodicity of events and edges can take lower and upper bounds to reflect the possibility of their variability. A basic explanation of PESP can be found in most works on the matter, refer to Liebchen 2006 and Stergidou et al. 2013 for a comprehensive documentation.

[4] See Wieczorek 2006: 45 for an extensive overview of construction and analysis tools for railway operations, both manual and algorithmic.

From the first practical application to the Berlin metro network in 2002, the complexity and power of PESP has grown considerably, such that to date, PESP can be used to model interval partings (Herrigel-Wiedersheim 2015), timetable symmetry (Liebchen and R. H. Möhring 2007), demand and mode choice (Chierici et al. 2004), disturbances (Kroon et al. 2004), hierarchical (Liebchen and R. H. Möhring 2007) and spatial decompositions (Herrigel-Wiedersheim 2015), line bundling, train coupling and parting, turnovers, track occupancy (Liebchen and R. H. Möhring 2007), large-scale path allocation (Nachtigall, Noll, et al. 2014), different intervals (Nachtigall and Opitz 2008), the insertion of singular trips into a regular timetable (Streitzig et al. 2010), the combination of periodic and non-periodic timetables (Yang et al. 2010), bottleneck analysis (Nachtigall, Noll, et al. 2014), automated microscopic routing (Pöhle et al. 2012), and much more.

As noted by Liebchen and R. H. Möhring, the scope of PESP potentially reaches from parts of network design down to vehicle scheduling. Given a predesigned hub structure, even Integrated Timetables, though initially out of scope for PESP, can be modelled consistently in PESP. TAKT, the standard software package developed alongside PESP, is used for the German long-term timetable design (Nachtigall, Noll, et al. 2014).

However, all approaches towards PESP focus on *timetable optimisation* on *existing infrastructure*. Since the infrastructure is modelled in such great detail, the approaches towards infrastructure upgrade require a *predefined* set of infrastructure options in order to evaluate them (Liebchen and R. H. Möhring 2007: 36f.).

By the nature of a *timetable optimisation* tool, albeit powerful, a *target timetable* of provably good quality can be obtained, but on a predefined infrastructure only. Therefore, the question of timetable-based infrastructure development (i. e. which long-term target timetable should future infrastructure upgrades be based upon) cannot be answered by this approach. However, upon construction of a preliminary hub structure and a presumably adequate infrastructure as presented in chapter 5, *the modelling power of PESP* (Liebchen and R. H. Möhring 2007: 3) can be used again.

3.1.2 OptiTakt

OptiTakt, as developed from 1998 by W. Hesse, Guckert, and R. Hesse, uses *desired hub locations* and *presumably achievable riding times* to define a grid network for the construction of an Integrated Timetable. *Minimal travel times* and transfer connections are then used as boundary conditions to construct an Integrated Timetable. An OD matrix is then used to compute *ideal riding times*, based on pure minimal riding times and minimal transfer times, to be compared with the timetable scenarios. Just as sketched in sections 2.3.5 and 3.3.5, the *sum of loss times* is calculated and used to rank competing timetable models to allow for a transparent scenario decision.

This model has been applied on north-eastern Bavaria (W. Hesse, Guckert, Scheider, et al. 2000); the cross-border links between Bavaria, Thuringia, Saxonia, and the Czech Republic; northern Franconia; the long-distance and medium-distance network between

Franconia and Saxony (W. Hesse, Guckert, and R. Hesse 2005: 14f); and the long-distance network of the Czech Republic (W. Hesse, Baudyš, et al. 2010, Baudyš et al. 2009). Since the input data incorporates presumably achievable riding times, rather than actual riding times, the solution space is considerably large and allows for a comprehensive variation of timetable scenarios. The development has been discontinued past 2010, but the tool is in use for planning in the mentioned German States and the timetable solutions developed there are largely still in place (W. Hesse, Guckert, and R. Hesse 2005: 14f. Opitz 2009: 14f.).

This approach is perhaps the timetable development approach closest to timetable-based infrastructure development, since from the selection of *presumably achievable riding times*, a methodology such as the Feasible Timetable Phase presented in this work could be attached.

3.1.3 HiTT

HiTT was developed by Kolonko et al. from 2001. Its main focus is on the evaluation of timetable concepts, but it includes a fine-grained timetable construction module. A set of possible paths is combined to feasible combinations and then randomly permutated to receive scenarios, which are then evaluated.

With this approach, possible infrastructure measures can also be evaluated, since the sum of necessary investments is also calculated upon evaluation. However, every measure needs to be known in terms of topological impact and costs, causing a predetermination of possibilities (Kolonko et al. 2002).

3.1.4 FASTA

The *FASTA* approach has already been developed in 1992 to evaluate competing timetable concepts. Tzieropoulos et al. took up the idea again in 2008 for a fundamental redesign. The idea behind FASTA is a *sketch simulation*, i. e. an evaluation of timetable concepts based upon their basic structure rather than a detailed infrastructure and timetable representation. With the input of a service intention, a set of possible timetable solutions is created, tested on stability and demand reaction, and then ranked.

However, this approach also relies upon a given infrastructure, such that no infrastructure upgrade measures can be derived from the timetable (Tzieropoulos et al. 2008: 21).

3.1.5 Further Approaches

The approaches of finding optimal timetables for timetable robustness evaluation, for single-track lines, for individually scheduled freight trains, and for dense urban timetables is countless. Since this work focuses on timetable and infrastructure design in passenger transport in regional and national networks, these approaches will not be discussed here in further detail; refer to, Kroon et al. 2004, Cacchiani et al. 2012, and Stergidou et al. 2013 for a comprehensive listing.

The first notable approach to timetable design in this context is the approach by Lee et al., where a network timetable is created by arranging periodically running trains in such a way as to achieve the best tradeoff between infrastructure load and passenger impact; in the approach, sections of the train paths are permutated to allow for a flexible train order and thus to improve capacity utilisation (Lee et al. 2009).

In a second notable approach, Schröder et al. developed a method to shift predefined paths of both periodical and singular trips such that the impact on passengers is minimised. Passenger impact is based on a categorisation of transfer qualities rather than transfer time so as to account for the annoyance of nearly missed transfers and the perception of waiting times (Schröder et al. 2008).

Finally, Vansteenwegen et al. based their timetable construction approach on both recovery time[5] optimisation and transfer time reduction. Starting from a current timetable, buffer times are derived from a delay and delay propagation simulation, and fed into a matrix of optimal recovery times. Then, just as in PESP, but with a linear rather than a cyclic time notion, the recovery time, the stopping time, the interval, and the transfer times are assigned ranges for valid values to be obeyed upon optimisation. Using linear programming, an optimal timetable is created. The objective function then requires the minimisation of the total costs of extended stopping times, extended transfer times, and reduced recovery times. From this perspective, both the user's benefit and the system's stability can be assessed at once. However, only the current infrastructure can be modelled, with any upgrade measures requiring a complete timetable reassessment.

3.2 Infrastructure Upgrade Approaches

The infrastructure upgrade approaches presented here cover a large field from bottleneck analysis down to the integrated development of timetable and infrastructure. Note that most of these approaches take a *target timetable* or a *service intention* as the basis for the infrastructure development, while approaches allowing for feedback between timetable and infrastructure development are considerably less common. This is will give an overview, sorted from the least mutual influence of the timetable and infrastructure development to a full integration of either. For an overview of a wide range of approaches with reference to infrastructure and operational design, refer to Wieczorek 2006: 70.

[5]Recovery time is named *running time supplement* in their work and measured in minutes rather than in percentage.

3.2.1 Bottleneck Analysis

There is much research being conductued in the field of *Bottleneck Analysis*, such that in this work, only a representative sample of works will be presented. Since these concepts focus on the identification, rather than the resolution, of bottlenecks, they can offer only an insight into the interdependencies of both infrastructure and operational concepts. For a comprehensive compilation of bottleneck analysis approaches, refer to Dewilde et al. 2013 and Chen et al. 2015.

Hantsch et al. and Li et al. model bottlenecks microscopically on basis of a detailed operational concept; Schaer make use of a set of comparatively coarse long-term timetables to be analysed for robustness and then ranked. The approach of Uhlmann et al. introduces helpful *interval clocks* for an interval-based justification of infrastructure elements: Clocks, just as used in this work (see section 2.1.4) are used to allocate periodic slots and derive train path conflicts (Uhlmann et al. 2004, Hantsch et al. 2013, Schaer 2013, Li et al. 2014).

3.2.2 Infrastructure Development by Macroscopic Abstraction

Sewcyk as well as Schlechte et al. follow an approach of (comparatively) minor infrastructure measures by abstracting the network just enough to dispose of minor constraints (such as inadequate presignalling distances or concatenated track dependencies) and allow for a judgement of upgrade potentials. From a definition of a timetable (Schlechte et al.) or a target number of trains (Sewcyk), respectively, an infrastructure load is derived, such that, finally, both upgrade and downgrade options in the infrastructure can be quantified by their (i) timetable, (ii) operational, or (iii) technological purpose (Sewcyk 2004, Schlechte et al. 2011).

3.2.3 Line and Station Standards

Melzow describes an approach to systematically categorise the German railway network. From the basic categories depicting train types (passenger/freight traffic transport only, mixed traffic, regional traffic), the further subdivision is carried out along design speeds[6]. These are derived from the presumable riding time demand as required by the Integrated Timetable, such that a mere basic type does not suffice to define the design speed. Furthermore, the design speed is considered a guideline rather than a strict design parameter, since the goal is to obtain target edge riding times. Furthermore, the design of stations is derived from a target timetable rather than design values. Contrary to earlier design approaches in Germany (but also in most European countries), no compromise, however, is made when it comes to switch speed and presignalling distances,

[6]The design speed is named *Leitgeschwindigkeit* rather than *Entwurfsgeschwindigkeit*, so as to refer to its primary use as a guideline.

even on regional lines, so as to remove speed restrictions (and therefore timetable restrictions) in train stations that purely stem from inadequate infrastructure details (Melzow 1997).

Pachl expanded this idea to standardise stations as well. Following the demand from an Integrated Timetable, the design principles for stations are derived. The design focus lies primarily on switch geometry, access/egress dimensioning, and platform design. The concept can be seamlessly interwoven with Melzow's approach, since the line standards directly influence the station layout. Following an analysis of the functional relations between the train systems, it is possible to identify the basic geometric and topological elements. This leads to a categorisation, the corresponding design standards, and finally the track layout per station (Pachl 1999).

Since the basic idea of this approach is to allow for an Integrated Timetable to be operational and at the same time drastically reduce the amount of different lines and designs, it can be used as a long-term design strategy when a target timetable exists.

3.2.4 Travel Time Based Infrastructure Upgrade

Reinold et al., Lai et al., and Kang et al. each pursue an approach of achieving a minimum riding time on a given relation. The goal is to first define a riding time goal, then construct a set of alignment measures to best fulfil this goal, and then to evaluate this goal economically. Reinold et al. based their analysis on a preliminary gravitation-based passenger potential analysis, where the projected amount of passengers is indirectly proportional to the riding time, thus calling for a change in service offer. A favourite alignment alternative is then selected by the ratio of costs per minute saved. Finally, a service offer is constructed on the target infrastructure (Reinold et al. 2012).

Lai et al. followed an algorithmic approach also aiming at minimum costs per minute saved. Instead of stating a target riding time, a set of scenarios is established, each reflecting a budgetary scenario of investment funds available. For each scenario, an optimal combination of measures is computed such that the largest possible decrease in riding time can be achieved. The measures combine infrastructure and vehicle measures and take into account the interdependence between them when it comes to the design of both alignment measures and rolling stock procurement (Lai et al. 2011).

Kang et al. finally follow a smaller-scale design approach. By quantifying land acquisition costs, riding time, and construction costs, an optimisation of a whole railway line in terms of all three parameters can be carried out. The approach is fine-grained enough to model at-grade street crossings, station layout, and even minor structures, which all can be created directly from the terrain model. Finally, a fine-grained alignment is obtained that best follows both riding time requirements and construction costs (Kang et al. 2011).

3.2.5 Inverse Capacity Determination

Wieczorek introduces the concept of *inverse capacity determination*. From a target timetable, a *design operational concept* is constructed. Then, the *parallel and sequential requirements* of an infrastructure are derived, such that an infrastructure, notably the *bottleneck elements*, can be systematically constructed from these requirements. Just like a standard engineering dimensioning task, the target load is set up, the infrastructure is dimensioned and then checked for adequate accommodation of the target operational concept (Bopp 2004, Wieczorek 2006: 60).

Further aspects of this work will be presented in section 4.2 and 5.5, where parts of this methodology will be developed further.

3.3 The Integrated Timetable

The ITF is an idea to develop a railway infrastructure to meet the timetable needs, while the timetable is constructed along strict, but simple design rules.

Also called *Integrated Clock-Faced Schedule* (Wardman et al. 2004), *Integrated Timed Transfer* (Clever 1997, Maxwell 1999), *Integrated Fixed-Interval Timetable* (Liebchen 2006) or simply with the German term *Taktfahrplan* (Johnson et al. 2006, Maxwell 1999), it needs no extensive introduction in this context.

3.3.1 Concept

The idea behind the ITF is to provide a timetable that yields *target riding times*, which are then passed over to the infrastructure. The infrastructure elements are then to be upgraded until the target riding times can be met.

Transfer stations arens to be spaced with $n \cdot {}^{T}\!/_{2}$ and arranged in cycles with a length of $n \cdot T$, either with $n \in \mathbb{N}$. This way *transfer hubs* can be created where all trains meet to make transfers in all directions. As long as all train systems involved run at the same interval and as there is one riding speed per edge[7], the *rule of edges* and the *rule of cycles* suffice to consistently obtain an Integrated Timetable (Lichtenegger 1990).

Lichtenegger, Weis, Kormanyos, and Uttenthaler presented consecutive approaches to upgrade an infrastructure according to a target timetable on a coarse level of infrastructure details (Lichtenegger 1990, Weis 2005, Kormanyos 2007, Uttenthaler 2010) .

Walter and Fellendorf expanded this approach to iteratively develop demand modelling, infrastructure, and timetable construction. Starting from a service intention and a target mobility structure, a set of coarse timetable models without feasibility checks is

[7]Edges with multiple riding speeds are possible only when the riding times differ by T.

developed which allow for extremal analyses of timetable concepts. These then undergo a demand evaluation to find (i) the general orientation of the timetable concept, (ii) the key indicators of successful timetable models, and (iii) the reaction of passenger demand on different parameter variations. From these findings, a feasible timetable is constructed to allow for operability. After another demand check, an iterative process of infrastructure development, demand modeling, and timetable construction is started. The process is stopped when a combination of target infrastructure and a target timetable is found that best fulfils the initially fixed goals (Walter and Fellendorf 2015, Walter 2016).

This approach is presented in depth in section 4.5 and expanded in chapter 5.

Since the installation of an ITF leads to a complete standardisation of all train runs, it is advantageous in several ways for railway networks[8], since (i) the concept of all transfer always working out allows for trips without additional time loss between all points of the network; (ii) the standardisation leads to targeted infrastructure investments that can all be justified with frequent use, while there is no need to build expensive but rarely-used infrastructure, (iii) the infrastructure can be based upon a long-term development strategy, and (iv) operational costs for feeder lines are comparatively low per ride, since all demand is bundled temporally onto a small set of (periodic) rides that carry feeding passengers for all connecting lines (Lichtenegger 1990, Wardman et al. 2004, Weis 2005, Wieczorek 2006, Tzieropoulos et al. 2008, Uttenthaler 2010).

3.3.2 Drawbacks

However, the Integrated Timetable does feature notable drawbacks. Liebchen, R. Möhring, et al. heavily argue against the idea of upgrading railway infrastructure according to a target timetable. The problem with the ITF is that the required infrastructure upgrade measures take long timespans in terms of design, decision, and implementation, while the demand and/or the target service offer change in a comparatively short-handed manner. Furthermore, the strict design rules call for a restrictive set of infrastructure upgrade options with only stepwise, but expensive improvement possibilities. The authors call for a periodic timetable optimisation rather than an infrastructure upgrade (and thus an ITF; Liebchen, R. Möhring, et al. 2004, Liebchen 2005).

At this point, it must be noted that the problem analysis can be supported, especially concerning the inflexibility of the ITF, but the conclusion must be viewed critically: If the only way of coping with riding times infeasible for an ITF is to otherwise optimise the timetable on existing infrastructure, there is no possibility to set a target infrastructure: All current service offers will be designed in such a way that an optimum on existing infrastructure is reached. What is more, infrastructure projects can be justified only in a subjectively defined environment and ranked only with a completely new timetable version per upgrade measure. With the approach of periodic timetable optimisation, it is impossible to spot necessary infrastructure measures by the needs of a timetable.

[8] with intervals of $T = 30\,\text{min}$ and above

However, once a target infrastructure has been set up (with timetable-based infrastructure development methods such as the one presented in this work), periodic timetable optimisation can deliver optimal timetables again.

This work is to both take up the problem analysis concerning the ITF and find ways of enhancing the ITF concept. The drawbacks are to be tackled, giving up neither the design advantages nor the strategic perspective inherent to the ITF. The following sections shall formally describe the significant design drawbacks of the ITF and derive the research demand.

3.3.3 Hub Spread Time

On first sight, an Integrated Timetable hub can be described as the intersection of multiple trajectories. When considered in detail, the situation is more complex.

Figure 3.1 shows a hub E located on two lines, one between A and B and one between C and D.

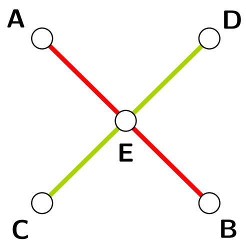
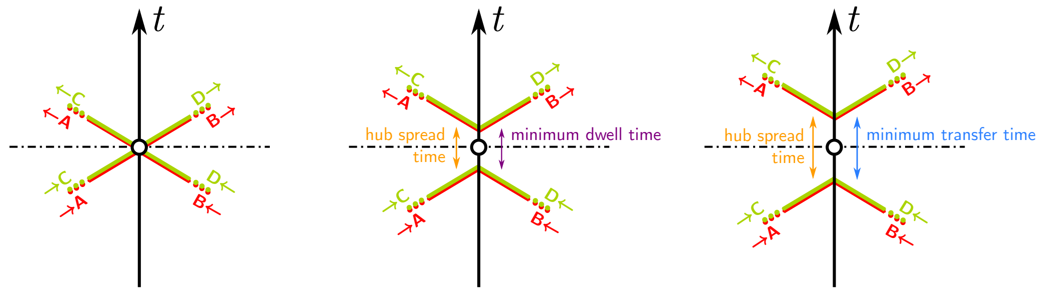


Figure 3.1: Sample hub neighbourhood

Obviously, a mere intersection of trajectories (see figure 3.2a) would imply that none of the trains actually stopped in that hub. The most simple amendment is to include the actual minimum dwell time for each train, as depicted in figure 3.2b. But since we focus on transfer hubs, the minimum transfer time $t_{\mathrm{tr,min}}$ (which is normally at least as long as the dwell time t_{dwell}, see figure 3.2c) will change the stop time t_h again. For this case, this stop time is still equal to the hub spread time t_s, since the two train systems do not interfere with each other, save the transfer time.

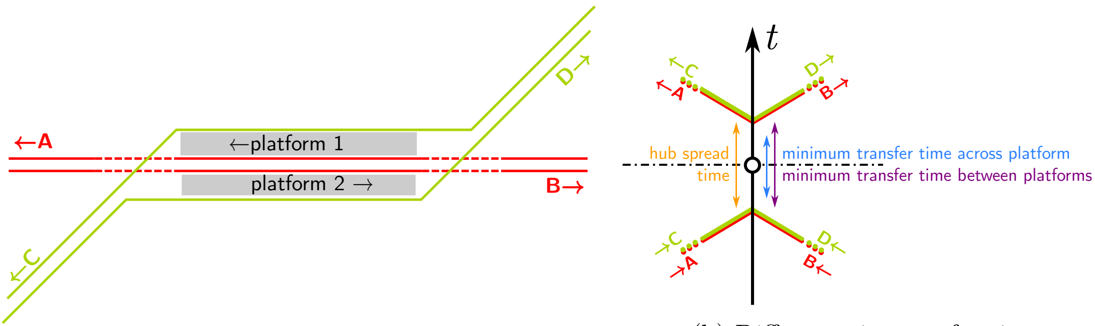
The minimum transfer time, however, is different depending on the infrastructure layout. Figure 3.3a shows a simple two-platform layout for hub E in figure 3.1 with grade-separation at the station gridirons, such that no route conflict will influence the time spent in the hub. However, since the two lines need two platforms, the minimum transfer time across platforms will not suffice for the case of a platform change. The minimum time spent at the station, without any further influences, will therefore be determined by either the minimum dwell time or the minimum transfer times between platforms (see figure 3.3b). Since the dependencies of cross-platform interchange are symmetrical (i. e. the timespans needed for a change between platforms 1 and 2 are equal in all relations)



(a) intersection of trajectories (b) minimum dwell time (c) minimum transfer time

Figure 3.2: Detailed view of a classical ITF hub

and there is no mutual interference save the transfer times, the hub spread time is equal to the cross-platform transfer time.



(a) Grade separated station layout for hub E

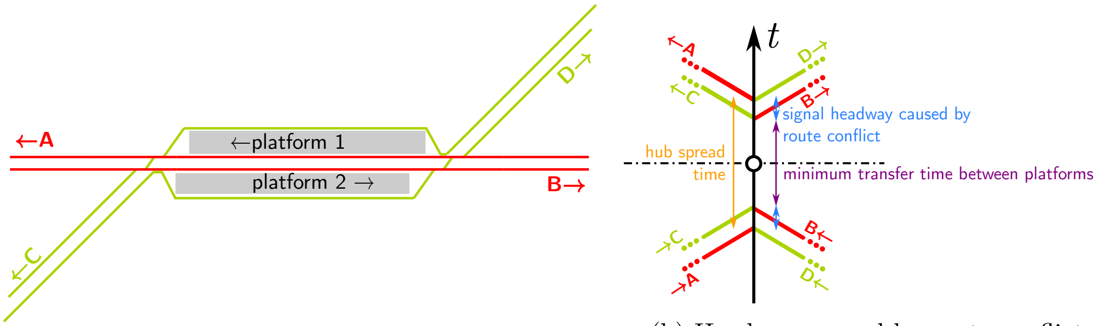
(b) Difference in transfer times

Figure 3.3: Transfer times across platform and between platforms

The case of completely grade separated station gridirons is limited to a small number of heavily used train stations[9]. Therefore, we usually have to deal with route conflicts whenever two lines cross. Figure 3.4a depicts the basic station layout from figure 3.3a with at-grade station gridirons. The necessary offset of one of the trains upon entering and/or leaving the station requires extra stop time (strictly speaking, it only requires extra time waiting for a route, but for passenger convenience, this time will rather be included in the stop time). Figure 3.4b shows a train order such that no train is penalised concerning its stop time, i. e. while trains $A \to B$ and $D \to C$ arrive first, they also leave first and vice versa. The hub spread time is now determined as the timespan between the arrival of the first trains (in this case: trains $A \to B$ and $D \to C$) and the departure of

[9]Note that this statement is to be viewed in the context of railway networks where timetable hubs actually occur. In dense, urban railway networks, there are, of course, many stations with complete grade separation; these are, however, usually not used as timetable hubs. Furthermore, stations that *are* used as timetable hubs and feature grade separation, such as *Utrecht Centraal*, *Zürich HB*, or *Wien Hauptbahnhof*, are mostly so extensive that the hub spread time is large due to (i) long transfer times and (ii) capacity constraints en route rather than within the station itself.

the last trains (in this case: trains $A \to B$ and $D \to C$. Note that this might be arranged differently, depending on the riding time requirements of adjacent routes; it does not change the hub spread time, however.



(a) At-grade station layout for hub E

(b) Headway caused by route conflicts

Figure 3.4: Hub spread time with at-grade station gridirons

One further complication here arises when several train systems are involved, i. e. when there is mixed traffic along the adjacent routes.

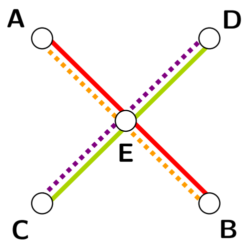


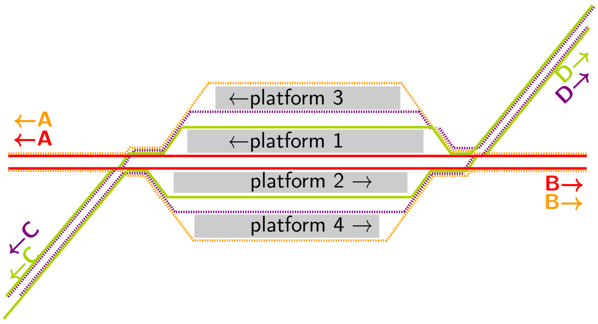
Figure 3.5: Sample hub with additional train systems

Figure 3.5 shows the same network as in the previous example, yet with an additional train system (dashed plot) on each line.

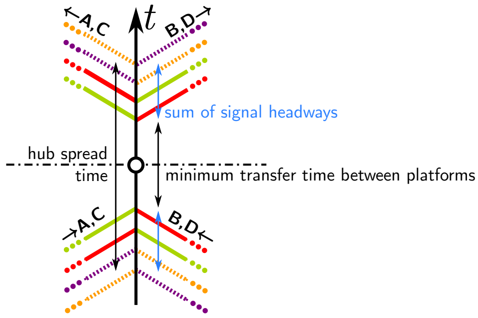
Taking the sample station layout from the previous example, two additional platforms are needed in order to accommodate the extra trains, as depicted in figure 3.6a. The infrastructure restrictions from the at-grade station gridirons remain in place and we allow no counterflow station approaches or exits[10].

Figure 3.6b finally shows the resulting hub spread time. It comprises the minimum cross-platform transfer time and the sum of all signal headways caused from train succession and route conflicts.

[10]Note that counterflow is a useful method of allowing parallel approaches or exits at timetable hubs in general. However, since we are dealing with a long-term planning approach and do not consider freight traffic, it is highly unlikely that the capacity needed to allow for counterflow can be offered once all freight train paths have been scheduled.



(a) Extended hub layout with additional platforms, hub *E*

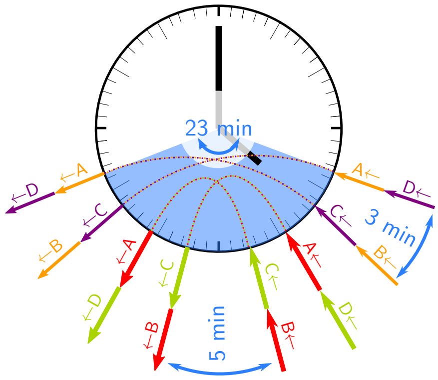


(b) Headway caused by route conflicts

Figure 3.6: Hub spread time for extended station layouts

min. dwell time, hub station	2 min
min. cross-platform transfer time	**5 min**
signal headway, each	3 min
signal headway, sum	**18 min**
hub spread time	**23 min**

(a) Typical values



(b) Hub clock

Figure 3.7: Typical hub spread time

If we set typical values for the aforementioned timespans and add them up, we get a rough estimate of hub spread times without grade-separation, as summarised in table 3.7a.

In other words, with only four train systems on two lines, the first trains will arrive 11.5 minutes before and the last trains depart 11.5 minutes past the axis of symmetry. For an hourly interval, this means that the hub spread time takes up more than one third of the interval and the longest occurring stop times take up exactly one third of the interval (see figure 3.7b).

As long as we require symmetrical transfer conditions (from all trains to all trains), the minimum time between the last arrival of one line and the first departure of another equals the minimum transfer time. This means that only one line can be prioritised to obtain reduced stop times (down to the actual dwell time, see figure 3.8), while the other lines need to be spread further to still offer enough transfer time. This allows for a shorter hub spread time, since the arrivals and departures of the prioritised line will move closer together. This results in a more compact stop time distribution. However, the line *not* prioritised receives extra stop time of double the minimum transfer time minus the minimum dwell time, in this example $2 \cdot 5 - 2 = 8$ minutes.

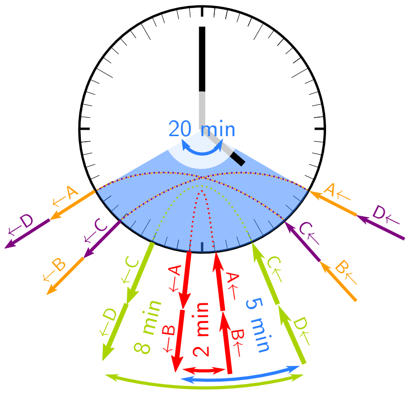


Figure 3.8: Modified hub clock with prioritisation for red line $A \leftrightarrow B$

If we take a look at the other lines, in this case the "local" purple and orange lines, these will always have a considerable hub stop time. This is mainly due to hierarchical arrivals and departures: the lower a train hierarchy (usually the slower a train), the earlier it arrives and the later it departs. Without multiple track layouts, this is strictly necessary so as to prevent faster trains from catching up to slower ones. Therefore, the target gross riding times for slower trains will differ considerably from the theoretical edge riding time.

Of course, multiple tracks, bypass tracks and a modified switch layout can ease these dependencies a little and allow for tweaks to the hub spread time. However, the minimum transfer time will always be a limiting factor. More complex or extensive station layouts extend this minimum transfer even further.

The stated values for hub spread times are, in practice, not uncommon. Table 3.1 shows some practical examples of timetable hubs in Germany, Austria an Switzerland. As can be seen, *St. Pölten* and *Bern* feature hub spread times that amount to almost an interval's time. Therefore, the advantage of planned transfers is heavily jeopardised by transfer times.

From the above thoughts we can derive the following issues to be covered when the Integrated Timetable is to be expanded:

I Timetable hubs trigger hub spread times proportional to the number of train systems serving a hub at a time.

II Even comparatively short hub spread times jeopardise the idea of the Integrated Timetable to ensure fast transfers.

Station	hub time	first arr.–last dep.	hub spread time
St. Pölten (AT)	.00	.44–.12	28 min
Bern (CH)	.00	.47–.13	26 min
	.30	.17–.43	26 min
Wiener Neustadt (AT)	.30	.18–.41	23 min
	.00	.49–.11	22 min
Chemnitz (DE)	.00	.49–.10	21 min
Bruck/Mur (AT)	.00	.51–.08	17 min
Chur (CH)	.00	.51–.08	17 min
Graz (AT)	.00	.51–.08	17 min
Chemnitz (DE)	.30	.20–.36	16 min
St. Pölten (AT)	.30	.22–.37	15 min
Amstetten (AT)	.00	.53–.07	14 min

Table 3.1: Practical hub spread times in full timetable hubs, sorted by hub spread time, 2016 timetables.

III Hierarchical train arrivals and departures lead to considerable losses of potential edge riding time for (usually local) trains with lower hierarchy.

3.3.4 Riding Time

Since the ITF derives its utility from the construction of universal transfer hubs, riding time calculation turns out to be the soft spot of the ITF. Since the edge riding time requirements are strict and only allow for large leaps in riding time changes, the resulting riding times for passengers can jeopardise the advantages gained from transfer time improvement: (i) the hub spread time, as described beforehand, directly influences the riding time negatively for most train types. In terms of riding time, we can find the hub spread time in each train system again in terms of the proportional hub stop time; (ii) the principal concentration of allowing *all* transfers in a hub potentially disadvantages strong passenger flows in favour of less important links. And (iii) the recovery time w for Integrated Timetables is considerably longer than in non-ITF environments to account for the high grade of interdependencies between links and the resulting propagation of potential delays across the network (Uttenthaler 2010: 36).

This means that, as soon as transfer relations in a hub become strongly weighted towards singular OD pairs or a strong flow to the hub itself, a railway line needs to feature, by trend, a considerably lower technical running time than in a conventional timetable environment in order to achieve a similar impact on passengers.

Consider a setting with two timetable hubs and one enclosed edge with $t_{r,\text{net}} = 45\,\text{min}$. As shown in figure 3.9, on either side of the edge there are transfer connections, most of which, in either case, we consider to be through passengers and passengers bound for the hubs themselves. If both hubs are served as full timetable hubs and all trains run at $T = 60\,\text{min}$, the edge riding time amounts to $t_{r,\text{edge}} = \{30, 60, 90, \ldots\}\,\text{min}$. Even if we

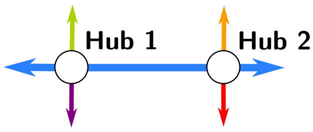
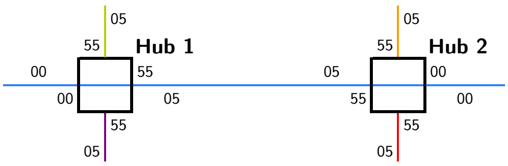
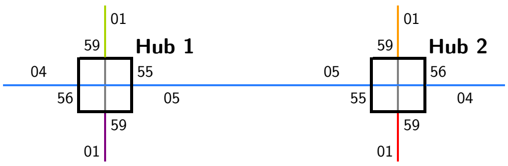


Figure 3.9: Example network with central edge

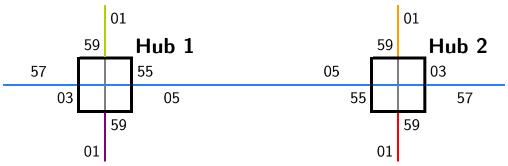
can ensure $t_{h,\text{prop}} = 1\,\text{min}$ only, this results in $t_{r,\text{gr}} = \{28, 58, 88, \ldots\}\,\text{min}$ and, considering 10 % recovery time, $t_{r,\text{net}} = \{25.5, 52.7, 80, \ldots\}\,\text{min}$. Apart from almost halving $t_{r,\text{net}}$, i. e. almost doubling the riding speed, the straight-forward solution in the Integrated Timetable is to set $t_{t,\text{edge}} = 60\,\text{min}$. Therefore, $t_{r,\text{gr}}$ needs to be stretched from 50 to 58 min, be it by incorporating more stops, a lower riding speed or by increasing $t_{h,\text{prop}}$.



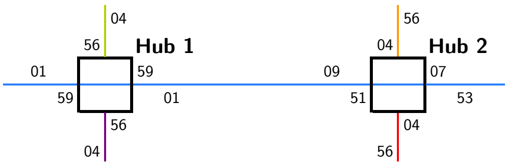
(a) Classical ITF solution, prioritisation of through passengers



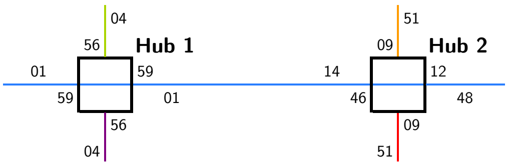
(b) Classical ITF solution, prioritisation of transfer passengers



(c) Negligence of minor transfer relations



(d) ITF hub at *Hub 1* only



(e) ITF hub at *Hub 1* only, reduced running time

Figure 3.10: Options for constructing the timetable hub structure for the example network

Figure 3.10 shows the possibilities for this network in an Integrated Timetable. We consider the option of increasing $t_{r,\text{edge}}$ by increasing $t_{h,\text{prop}}$. This way, the passengers bound for the hubs are not disadvantaged. This makes, at $w = 10$ %, for $t_{r,\text{gr}} = 50$ min. For the use in an Integrated Timetable, one of the settings as depicted in figures 3.10a and 3.10b can be obtained. In the former example, the branches of the main line depart outwards as early as possible to make up for the lost time in the hub. Nevertheless, the through passengers lose a total of 10 minutes in both hubs compared to a ride without extended stops. In the latter example, the branch lines are combined to one through train each, such that the transfers to and from the main line can be improved, but this leads to an additional 4 minutes lost in each hub for the through passengers of the main line.

Figures 3.10c and 3.10d show possibilities of giving up those parts of the hub with minor importance. While the first possibility spreads out the shorter riding time amongst the two hubs, the second one gives up one full hub and retains the other.

Finally, 3.10e shows the effect of a minor running time improvement. With a full hub on either side (not shown here), this merely effects the passengers bound for the hub itself, leaving the rest of the hub structure unchanged, therefore the benefit is only minor. When one hub is dropped, the travel time improvement is passed on through this hub; the benefit is directly proportional to the riding time decrease.

This elaboration can be used for the list of issues for further research:

IV Denser intervals call for less adherence to full timetable hubs.

V The adherence to edge riding times compatible with full ITF hubs can be justified only when transfer passengers outweigh hub-bound and through passengers. Otherwise a full ITF hub leads to an increase of total travel time.

VI Full ITF hubs are best used in medium-sized cities where transfer, through and hub-bound passengers are balanced and where the number of train systems is low enough to allow for a slim hub spread time.

3.3.5 Demand Orientation

The basic idea of Integrated Timetables is to allow for *all* transfers to *always* work out. In theory, this lowers the total riding time, since any passenger will make his journey without losing time due to lost transfers. In practice, however, as shown before, this is only true when transfer relations within a hub are spread as evenly as possible.

Figure 3.11 shows node flows for the example sketched before. As can be seen, the dominant node flows are through passengers on the main line from beyond *Hub 1* to beyond *Hub 2*. Transfers from the branch lines run predominantly towards the central edge, while the flows towards the outer edges are minor.

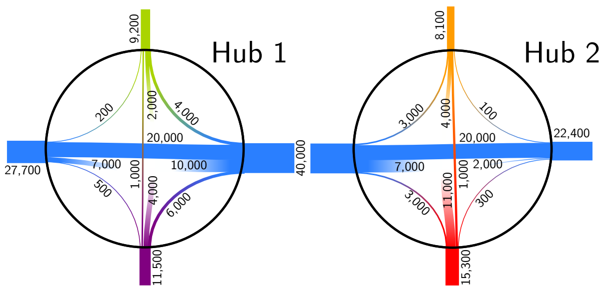


Figure 3.11: Node flows for the example network

If the effects of the five variants sketched are quantified by simply multiplying the time lost at transfers/hub stops by the amount of passengers, it can be seen that in this setting, a strict adherence to the Integrated Timetable causes significantly less benefit than a shift towards a more demand-oriented timetable construction approach. Figure 3.12 shows the sum of passenger minutes lost per timetable version. As can be seen, the three possibilities[11] of dropping the ITF principles perform significantly better in terms of lost passenger minutes than the versions with two full hubs, even though two to four transfers are dropped completely.

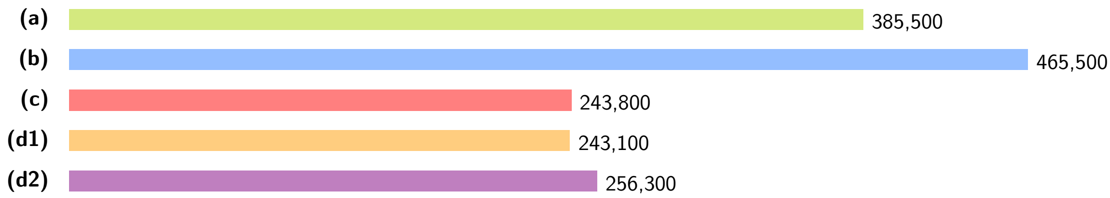


Figure 3.12: Additional travel time compared to all-direct trips per timetable version in passenger minutes

It must be noted again that this is to roughly quantify the advantages and drawbacks of the Integrated Timetable[12]. Of course, the more evenly the transfer relations are spread across the node flows, the more important an adherence to a full timetable hub becomes. Without constantly taking into account the demand, however, the result is bound to perform worse than the non-ITF alternatives.

The list of issues can now be expanded as follows:

VII Junctions with a large set of feeder lines and denser intervals can be served better when the transfers are split up amongst several hub events.

[11]Note that (d1) and (d2) as denote the versions of (d) with mirrored dropping of the timetable hub. (d1) features an ITF hub at *Hub 1* only, (d2) at *Hub 2* only.

[12]Note that if we were to quantify the node flows as sketched in section 2.1 and the main line was a long-distance line, the difference in lost weighted passenger minutes would be even more distinct.

VIII The detailed timetable layout of a hub must be adjusted according to the actual passenger flows, so as to appropriately prioritise strong passenger flows and neglect strong ones in case of overall disadvantages.

3.3.6 Hub Location

One key feature of the Integrated Timetable is the location of transfer hubs. As noted beforehand, the distance between two hubs must be an integer that is a multiple of half the interval. Upon determination of hub locations, the further development of a station into a timetable hub can be carried out along standardised design principles which incorporate the number of tracks, the platform layout, and the parallel and sequential service requirements (Wieczorek 2006: 54ff. Pachl 2011: 201ff.).

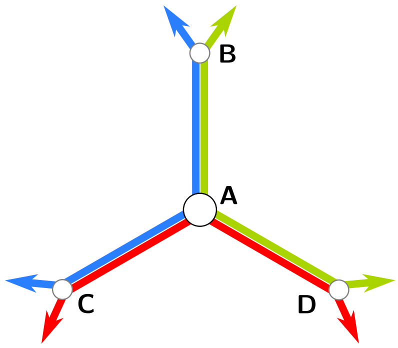
However, the location of timetable hubs requires far-reaching decisions at an early stage of the design process. Considering the long decision, design, jurisdiction, and implementation cycles of infrastructure development and the comparatively fast changes in passenger demand, the decision for a hub location can lead to a potentially unfavourable location decision that is hard to change midway.

Additionally, a gradual growth of timetable hubs over time without an evaluation of the hub structure can lead to excessive hub spread times as described beforehand, whereas a design from scratch would, at identical service intention, probably yield a looser arrangement of transfer relations.

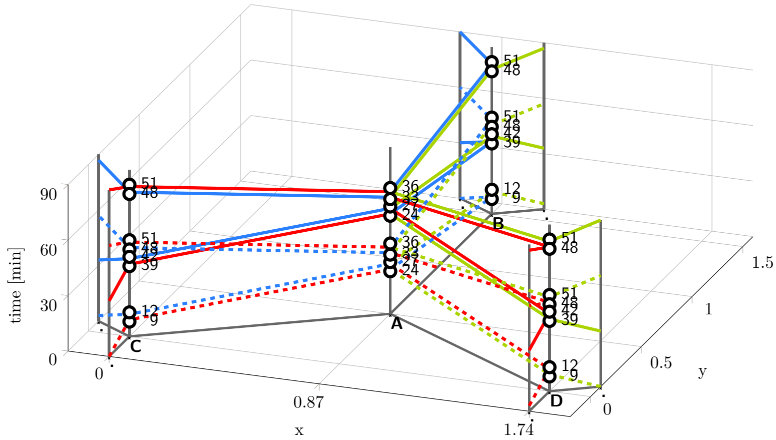
Consider a setting with one central station designed to work as a full timetable hub with three branches on the immediate perimeter and six branches in total. With an hourly interval, the straight-forward solution is a full timetable hub at the centre to allow for all transfers in all directions to work out indiscriminate of the actual connection of branches into lines. Figure 3.13a shows such a layout. This situation requires a central hub with at least six tracks, while the remote stations can be simple two-track branching stations. All the branches are served by two lines each and therefore the signal headway has to be taken into account. Therefore, the minimum hub stop time per train is increased by the signal headway. With the typical values already used in section 3.3.3, this makes for a minimum hub stop time of 8 minutes and a hub spread time of 12 minutes.

When a situation like this is to be upgraded to a denser interval, the basic timetable structure can remain unchanged in an Integrated Timetable, since the only change is in the interval. Figure 3.13b shows the basic, hourly interval in continuous lines and the additional trains for the half-hourly interval in dashed lines.

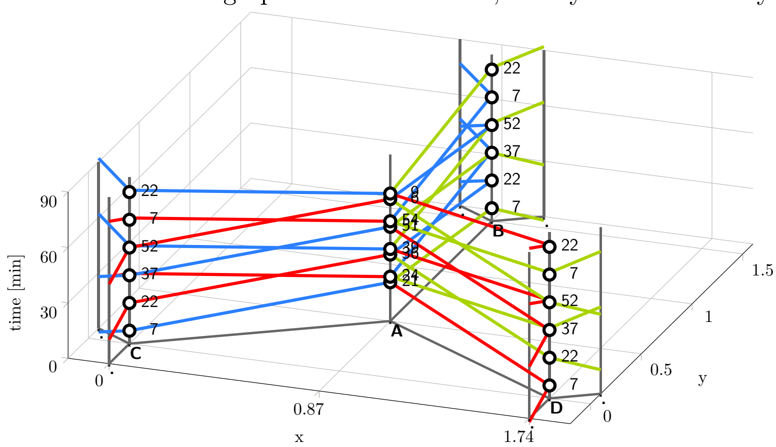
Additionally, if the distances between the central hub and the branching stations are adequate (i. e. an odd integer that is a multiple of half the denser interval; for details see section 5.1.2), the branching stations may also serve as transfer hubs between the remote branches. However, as can be seen, the hub spread time now almost equals half



(a) Network arrangement



(b) Network and timetable graph for a central hub, hourly and half-hourly interval.



(c) Network and timetable graph for dissolved hubs, half-hourly interval.

Figure 3.13: Hub location with decreasing intervals in Integrated Timetables

the interval and the time spent in the station amounts to 30 % of the riding time between the branching stations.

Had the network with a half-hourly interval been designed from scratch, a situation with a dissolved hub structure would yield a different situation for both infrastructure design and passengers. Figure 3.13c shows the arrangement with three dissolved hubs rather than a central one. As can be seen, the denser interval allows for the service of all stations without extended dwell times, with both the transfers between the branches and in the central hubs retained. Since the transfer in the central hub incorporates only three trains at a time, it can even be accomplished all at one platform (by the use of a central bay platform). The transfer along common edges (i. e. between those trains running at signal headway in the ITF solution) is, at first sight, lengthened to 15 minutes, but since it amounted to the full hub spread time in the strict ITF solution, this only accounts for 3 additional minutes each and only affects three transfer relations as opposed to six relations in the ITF solution.

However, such a timetable requires different infrastructure layout, especially around the central hub, since only three platform edges, but a complex system of bypasses at the station gridirons is required, while the more classical approach can be operated with classical station gridirons, but requires six platform edges. Finally, the solution with dissolved hubs requires a different set of target edge riding times for three of the six branches, since a different kind of hub is reached at the branching stations. A shift from the full hub to dissolved hubs would therefore require both abandoning already-built structures and building completely new ones.

This adds two last issues to the list:

IX Larger cities can be better served with a larger set of geographically dispersed hubs that each take over parts of the transfer relations.

X Changes in service offer need to be treated like new concepts rather than just adding or subtracting trips within an unchanged service pattern. If forseen, the infrastructure is to be designed to accommodate either concept with moderate modifications.

3.4 Research Demand

Summing together all properties, advantages and shortcomings of the Integrated Timetable as covered in this section, the following research demand for the expansion of the Integrated Timetable can be derived:

I Hub concepts must favour slim hub spread times.

II Hierarchical arrivals and departures must be replaced by time-optimal travel chains within a hub.

III Lines of denser intervals should serve hubs more loosely in order to reduce hub stop time.

IV Travel demand must be incorporated directly into the timetable construction process.

V The approach must feature both spatial and temporal hub dissolution.

VI Demand changes and their impacts on the timetable structure must be carefully evaluated and quantified beforehand to be able to react with the appropriate timetable structure.

4 Railway Infrastructure Design Process

At first sight, it might seem awkward to elaborate on the Railway Infrastructure Design Process, since most publications in this field cover it right at the beginning. However, this work aims to create a target timetable to be provided to an infrastructure manager, who is to provide adequate infrastructure to accommodate the target timetable. As soon as we start to position this task within *any* of the currently described design processes, we quickly reach their limits, leading to comprehensive extensions of existent models, while core insufficiencies will persist.

Therefore, this chapter is to analyse existing descriptions of the Railway Infrastructure Design Process, retrieve relevant advantages and drawbacks, and finally create one new process description. This process will then serve as the framework for the methodology presented in this work.

Herrigel-Wiedersheim (2015) describes the four stakeholders involved in timetabling: Public Authorities (Principal), Customers (both freight and passenger), Train operators (Agent), and Infrastructure Managers, as depicted in figure 4.1. We need to keep these stakeholders in mind for the further discussion.

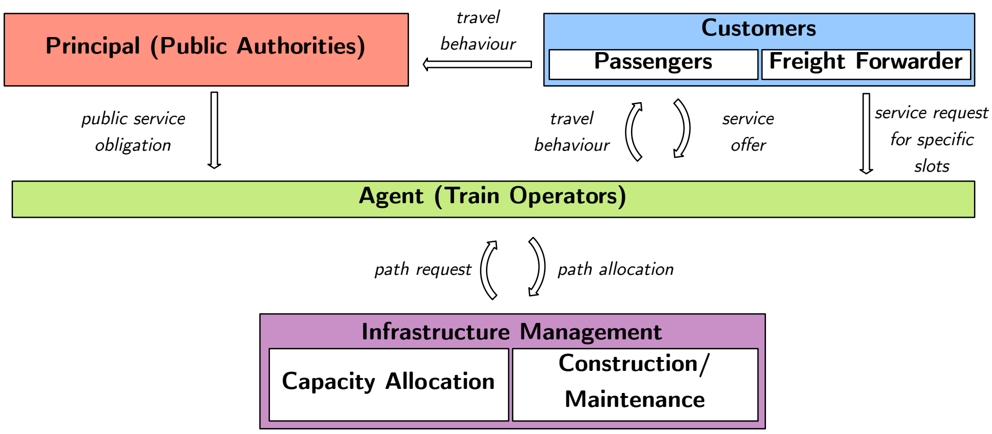


Figure 4.1: The four main stakeholders in timetabling (adapted from Herrigel-Wiedersheim 2015 and updated to 2016 knowledge)

For the further considerations, the following wording shall apply:

- a *Task* shall describe every single planning step, such as timetable design, network design and the like.
- a *Phase* shall denote a set of *Tasks* grouped together to form a unit of similar tasks, common iterations, or common strategic focus.
- a *Process* is the combination of *Phases* to describe the whole procedure from the initiation to the final railway service.

4.1 Sequential Design Process

Classically, the design process of railway services, including the scope from strategic to operational planning, is modelled as a sequential design process or *Waterfall Model* (described in Liebchen and R. H. Möhring 2007: 118). Design results from upper planning tasks will trickle down to the respective next tasks, while insights from lower ones will flow back to their respective predecessors. Figure 4.2 depicts such a waterfall model for the railway design process. *Network Design* and *Line Planning* would classically be part of the *Strategic Planning Phase*, while *Vehicle Scheduling*, *Duty Scheduling*, and *Crew Rostering* make up most of the *Operative Planning Phase*. *Timetabling* is located on the interface between strategic and operational planning.

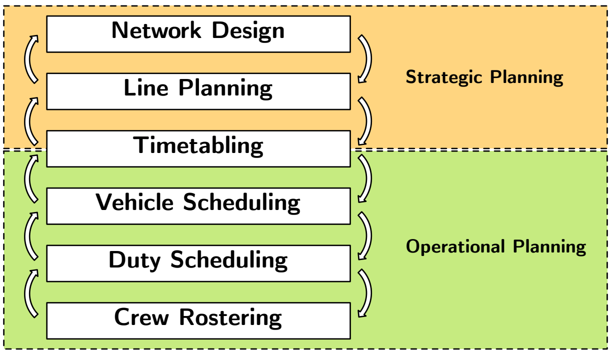


Figure 4.2: The Sequential Railway Design Process according to Liebchen and R. H. Möhring (2007: 118) and Herrigel-Wiedersheim (2015: 10)

If we take a closer look at just the *Timetabling* task (as carried out by Herrigel-Wiedersheim (2015) and Bickel et al. (2010), see table 4.1), we can see that something like the timetable design task actually spans all the way from the beginning of strategic planning (time horizon > 10 years) down to (real-time) online operations.

Several tasks in this depiction of the timetabling process run in parallel with preceding or successive tasks from figure 4.2 or even across several phases, as shown in figure 4.3.

	Strategic Planning	**Long-Term Planning**	**Mid-Term Planning**	**Short-Term Planning**	**Online Planning**	**Online Operation**
Time horizon	>10 years	5–10 years	years–months	months-days	hours	minutes-seconds
Duties of the Infrastructure Manager	strategic network development	infrastructure planning	capacity and works planning	capacity allocation	traffic management	traffic control
Duties of the Train Operator	concept planning	service planning	timetable planning	timetable production	rolling stock and staff dispatching	train operation

Table 4.1: Detailed view on timetable design (Bickel et al. 2010: 203)

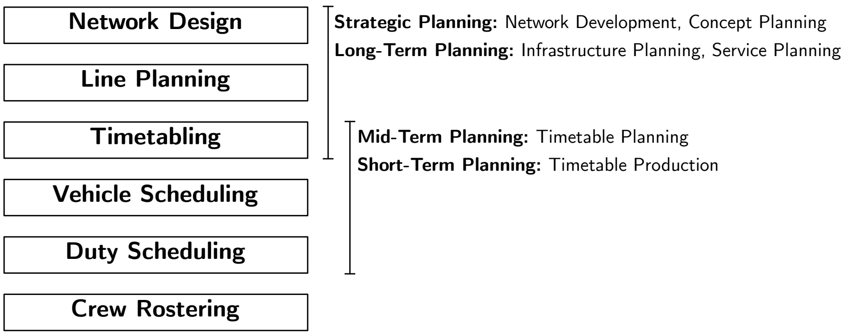


Figure 4.3: Timetabling tasks spanning across several tasks of the classical sequential design process.

Even more, the structure as depicted in figure 4.2 fails to describe even a standard situation in railway infrastructure design:

Consider the following setting in a classical design process: A public authority requests a railway network to be (re-)designed. The infrastructure manager is provided a demand prediction and the train operator is supplied with a service intention. The infrastructure manager will then construct a network with an infrastructure from scratch following rough demand predictions. Afterwards, the train operator will design a service concept on this network and try to install the Service Intention on the given infrastructure. Finally, a timetable will be constructed and handed back to the infrastructure manager, who is to fit all path requests onto the infrastructure.

First, there is a feedback loop between *Timetabling* and *Network Design* as soon as the infrastructure needs to accommodate several lines (designed in *Line Planning*) on one stretch. Otherwise, either the infrastructure needs to be overdimensioned in the first place (not knowing which kind of network two parallel lines might actually require) or the timetable might not reach the target Service Intention (when an a priori underdimensioned infrastructure fails to accommodate all requested paths to fulfil the Service Intention). If the feedback went back to line planning only, the objective is to redesign the line concept rather than to adapt the network—leaving a potentially inferior service of the demand prediction than initially intended (Weigand 2005: 8, Wieczorek 2006: 27).

As long as the Service Intention covers direct trips without transfers only, this process might still be somewhat functional as is. However, as soon as passengers need transfers, the quality of the latter becomes relevant. A thorough investigation on the impact of the service offer on the travel behaviour is impossible without in-depth knowledge about timetable and hub layout. This even installs a feedback loop from *Timetabling* outside the classical design process back to the public authority, who is to reassess the demand predictions and, possibly, redimension the Service Intention.

Within the *Operational Planning Phase*, classically, there is a similar situation:

Timetable slots assigned by the infrastructure operator will lead to a timetable, which is then used to set up vehicle circulations. Then, each vehicle is assigned personnel duties which lead to crew rostering.

If vehicle circulations turn out to be infeasible within the timetable, a step back to timetabling can trigger a timetable redesign. However, upon duty scheduling, a timetable infeasibility can appear without any effect on the vehicle circulation: Minor changes to the timetable might, though, help duty scheduling. The feedback loop to vehicle scheduling would, again, only change vehicle circulation without information about possible better solutions during timetabling.

Figure 4.3 depicts these spans of the *Timetabling* task only – but the tasks of *Network Design* and *Line Planning*, just to mention those of the strategic phase, span just as much across several tasks or phases.

The classical sequential design process has been modified by Peeters, Michaelis et al., and Rittner et al. Some of the above stated problems are eased by these modifications, but still, the problem of a strictly step-wise work persists (Peeters 2003: 23 ff. Michaelis et al. 2009: 212, Rittner et al. 2009: 6ff.).

Caimi set the task of *Timetabling* all along the span from *Strategic Planning* to *Rescheduling* (here: *live operational rescheduling*). *Infrastructure Planning* and *Line Planning* occupy the *Strategic* and the *Tactical Planning* levels. As opposed to the previously shown approaches, the different planning tasks and the different time horizons are arranged perpendicularly (see figure 4.4), which shows their great interdependence. Wieczorek did likewise, albeit in coarser granularity (Caimi 2009: 14, Wieczorek 2006: 16).

Summing up, we can retrieve the following requirements for the creation of a comprehensive design model:

I Full-scale iterations need to take place at least between timetable, infrastructure, and network design.

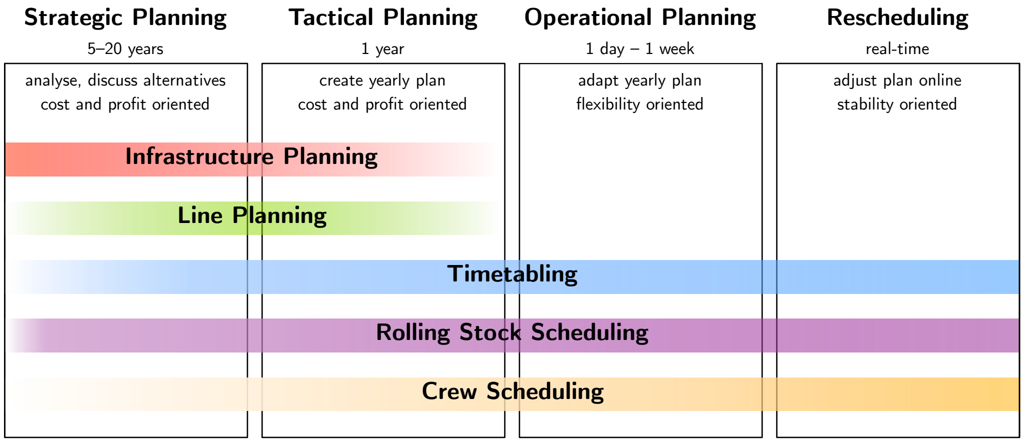


Figure 4.4: Perpendicular arrangement of planning tasks and time horizon (Caimi 2009: 14)

II Feedback loops will, at least, lead outside this process and touch service intention and demand.

4.2 Timetable-Oriented Design Process

Approaches such as the ITF rely heavily upon an infrastructure that is fit to accommodate timetable requirements. Large-scale infrastructure upgrade projects such as *Bahn 2000* and *ZEB* in Switzerland or *NAT91* and *Zielnetz 2025+* in Austria are based on an Integrated Timetable (Caimi 2009: 15, BAV 2016, ÖROK 1992, ÖBB 2011). Infrastructure measures in the target network are derived from the requirements of the target timetable. Figure 4.5 shows this kind of process for the *Bahn 2000* project[13]. This principle basically inverts the *Strategic Planning* dimension of the classical design approach. The idea behind this approach is to construct a timetable model first and then derive infrastructure from it. This model not only puts timetabling strictly in the strategic planning phase, but also requires all actions on the infrastructure to be taken as a consequence of timetable decisions.

Uttenthaler expanded the *Infrastructure Planning* and *Timetabling* tasks of this process to account for the more iterative nature of this process, as shown in figure 4.6.

What we can find in this design process first are three feedback loops:

1. The Network Modification Loop from *Target Riding Time* back to *Target Riding Time* via *Network Context*.

[13]For explanations on the ideas of the Integrated Timetable refer to chapter 3.3; for an explanation of the term *Target Edge Riding Time* refer to chapter 2.2.3

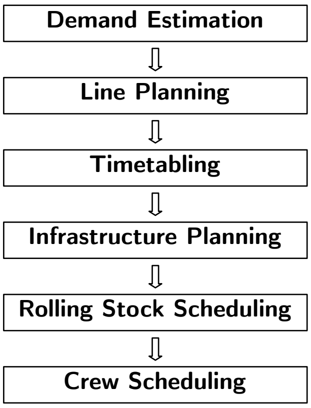


Figure 4.5: Timetable-Oriented Design Process for *Bahn 2000* in Switzerland (Caimi 2009: 15)

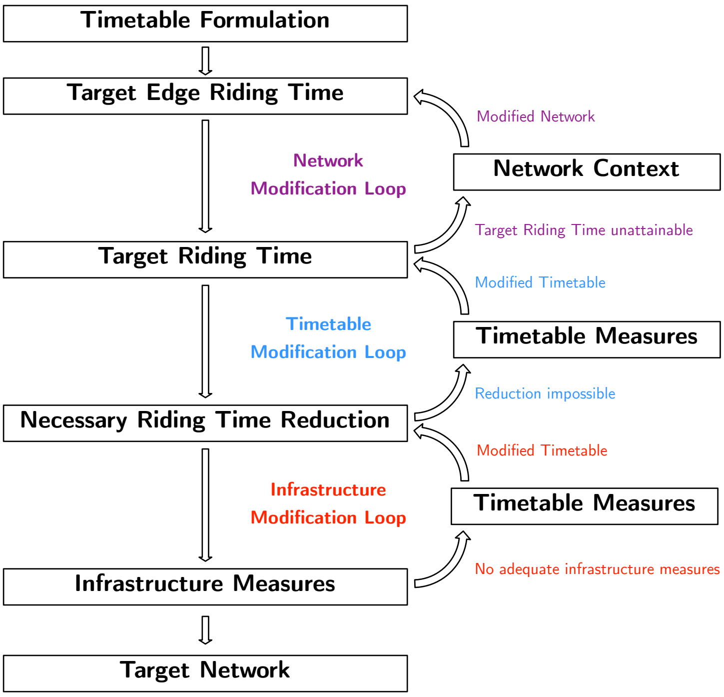


Figure 4.6: Infrastructure planning and timetabling in the timetable-oriented design process, modified from Uttenthaler (2010: 257)

This loop describes the inderdependency between *egde* and *cycle*[14] riding time, since an ITF only works if either are compatible with the network. Within this feedback loop, the geographical location of timetable hubs within the network can be set and modified to meet the topological requirements—or, inversely, fixed geographical locations are connected topologically with new edges that fulfil the requirements.

2. The Timetable Modification Loop from *Necessary Riding Time Reduction* back to *Target Riding Time* via *Timetable Measures*.

   Within this loop, minor timetable measures can be taken to relieve riding time inadequacies when adjacent or partially parallel edges face considerably different riding time requirements. With the mentioned minor timetable measures, tight riding time requirements can be shifted across the network.

3. The Infrastructure Modification Loop from *Infrastructure Measures* back to *Necessary Riding Time Reduction* via *Timetable Measures*.

   This loop is perhaps the most heavily used one, since it directly incorporates timetable and infrastructure measures. Necessary riding time reductions from the (then already modified) timetable formulation need to be put in place in terms of infrastructure measures, i. e. modifications of the current infrastructure[15]. However, there might be required riding time reductions that are simply unattainable by infrastructure measures, such that the timetable needs to be modified again for a relaxation of the requirements.

Note that all three feedback loops imply timetable modifications in varying levels of granularity. Timetabling is thus seen as the driving force behind the infrastructure layout, but will, itself, be modified gradually along the different levels of design.

Wieczorek also follows the timetable-oriented design process, naming it *Inverse Capacity Determination*. The generic approach allows for a track-fine modelling of infrastructure according to a given timetable. By a discrete scan line approach, sequential and parallel infrastructure requirements are derived from the timetable, resulting in the required number of tracks and conflict-free paths for a given timetable (Wieczorek 2006: 95ff).

For the creation of a comprehensive design process, we can take the following requirements from the timetable-oriented design approach:

III Railway Infrastructure is to render possible a timetable which aims to fulfil a service intention. Infrastructure is thus the result, rather than the boundary condition, of a design process.

[14] For details on riding time requirements in the ITF, refer to chapter 3.3.

[15] Certain operational and vehicle measures such as acceleration and deceleration, uncompensated lateral acceleration, station stop time and the like might also be incorporated here, since they can be treated like infrastructure measures in terms of their effect on the timetable. This topic will be covered in chapter 6.

IV Since the timetable is altered several times, a deadlock must be prevented where a timetable is changed back and forth.

4.3 Design Models outside Railway Engineering

The discussion on the applicability of a waterfall model happened in Software Engineering about 40 years ago (Royce 1987: 329ff.), so we might draw upon some of these findings for our thoughts. The problems with a sequential design process gave way to Iterative and Incremental Development (IID), which is a multitude of processes that include many small, incremental iterations along the whole development process rather than one big process with specifications at the beginning and implementation at the end.

The problem field of changing specifications, circumstances, or strategies occurs in Railway Design simply due to the long service lives, design, and implementation procedures. This puts sequential models in danger of still following a long-term strategy, while in parallel the outside world develops differently than predicted. But the sequential model itself also has drawbacks: Genuine feedback loops can happen outside the model only (Mills 1976: 266).

Swartout et al. expanded this problem field with the conclusion that *Specification* and *Implementation* are, in Software Development, not different from each other, but merely subsequent successors of each other. The resulting implementation of one stage determines the specifications of the next, so only their position in the current design stage makes them distinguishable. They recommend the joint development of specifications along an incremental design process (Swartout et al. 1982: 438ff.). For our purposes, this is true for *Demand* and the *Timetable and Infrastructure* concept: Within a design phase, the demand estimation defines the service intention, thus the timetable and the infrastructure, but the resulting mobility patterns form a new iteration of the demand. If we considered fixed demand only, we would ignore this link.

We can translate the following key aspects from Software Engineering to the railway design process:

V When a design process involves several tasks that overlap, iterations need to incorporate all these tasks and not just adjacent ones.

VI No matter how intensive our prior investigations, we will almost certainly stumble across unforeseen issues that require all involved tasks to be modified.

Throughout the discussions about shortcomings of the waterfall model in Software Engineering, IID had emerged around critical defence and space design processes (Larman et al. 2003: 47ff.). There are many different approaches involving *iterative* or *incremental* elements, or a combination of both.

In their guide, *Managing Design and Construction Using Systems Engineering*, the U.S. Department of Energy translated IID, naming it *Evolutionary Development*, to infrastructure projects. IID (named *Spiral* and *Incremental Development*) is recommended against sequential design processes when

- the desired project outcome can be stated but associated requirements cannot be defined;
- the requirements associated with the outcome can be defined but do not appear immediately achievable because of technology, engineering, or funding constraints;
- having an operational project that partially satisfies owner and stakeholder expectations is more desirable from a cost/benefit standpoint than not having or delaying the project until the necessary capabilities become available;
- the project is specifically designed with adequate flexibility to allow future upgrades.

(U.S. Department of Energy 2008: 23f.)

Obviously, *all* of these prerequisites are present throughout the Railway Design Process.

While there is no single distinct method applicable, the guide recommends, amongst others, several *Sublevel Strategies* for an IID approach that we can make use of:

- *Broader Based Integrated Project Teams* to account for interdisciplinarity;
- *Consequence and Scenario Based Planning* to master uncertainties and unknowns;
- *Sensitivity Analyses* for a better understanding of interdependencies;
- *Set-Based Design* to make a postponement of yet uncertain decisions possible without endangering the whole project;
- *Modularity* to account for long service lives with interleaved implementations.

(U.S. Department of Energy 2008: 25ff.)

These sublevel strategies shall then be integrated into an iterative and incremental design approach. As can be seen easily, none of these strategies would actually work in a strictly sequential design model, since all approaches actively jolt through several levels of design.

This way, we can add one more requirement for a comprehensive design process model:

VII Adequate interdisciplinary sublevel design strategies call for a full iteration between several design tasks and several fields of engineering.

4.4 Spiral Model

The best known iterative approach in software design is the *Spiral Model* as proposed by Boehm (1988). The principle behind this model is to gradually enhance a fully-working *Prototype* by repeating cycles of development. The end of each phase is the beginning of the respective next phase, so the experiences gained in each phase can be used as input for the next.

We cannot directly take over this model for the Railway Design Process. However, we can adapt this kind of cyclic nature. Scheidt adapted the Spiral Model to respect the peculiarities of the Railway Design Process, as shown in figure 4.7. The design process is split into *Tasks* (encircled and named Ⓐ to Ⓕ) and *Transitions* (inscribed in diamonds and named ◇0 to ◇7).

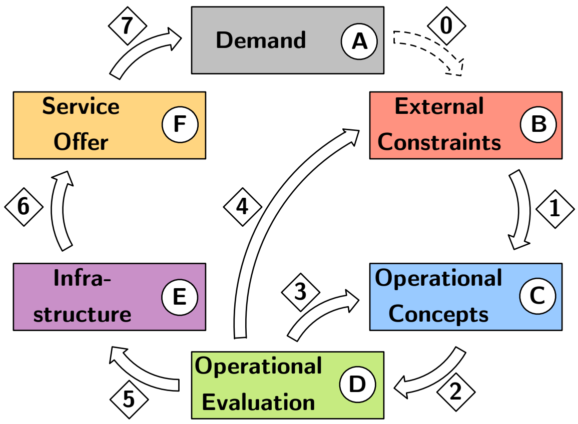


Figure 4.7: Adapted Spiral Model for the Railway Infrastructure Design Process (Scheidt 2016: 62)

In this model, a *Demand* Ⓐ is interpreted to form rough *Decision Guidelines* towards the creation of a *Service Intention*. *External Constraints* Ⓑ (financial restrictions, political decisions, geography) channelise these guidelines to form *Specifications* ◇1 for infrastructure operators. This then triggers the development of an *Operational Concept* Ⓒ.

Depending upon the desired depth of data provided, this concept triggers a first iteration, where *Input Parameters* ◇2 are passed on to an *Operational Evaluation*, where this concept is verified for feasibility and, if necessary, hands infeasible concepts back for redesign ◇3.

Within this evaluation, a great amount of required input parameters cannot be retrieved from the operational concept directly and thus needs to be obtained by an *Educated Guess* which is presumably sound, but relies heavily upon the experience of the planning personnel involved.

If this cycle is unsuccessful, the concept may be handed back for a reconsideration of the Service Intention, redesign of the infrastructure, or a shift to a different mode of operation ⟨4⟩[16].

When the evaluation identifies an operational concept as feasible, its parameters are handed over as *Design Criteria* ⟨5⟩ for the corresponding *Infrastructure* Ⓔ[17]. Due to the cyclic nature of the whole design process, modification steps render an infrastructure that is feasible only for a limited period of time.

On this infrastructure, a *Service Offer* Ⓕ to serve customers can be constructed ⟨6⟩. Additionally, wear and tear to the infrastructure might also require modifications over time.

The customers, for their part, react to the service offer ⟨7⟩ by means of a modified *Demand* Ⓐ (Scheidt 2016).

Scheidt does note that often the link from *Demand* Ⓐ to the rest of the process is missing, i. e. transition ⟨0⟩ remains undone. Furthermore, the cycles to arrive at a modified demand are too long to properly respond to reactions or changes in demand. Finally, Scheidt notes that binding targets or even clear formulations within the service intention rarely ever exist.

We can use these findings for two more requirements that a suitable railway design process should feature:

VIII A design process is to be oriented as close to the demand as possible.

IX The railway design process fully relies on clear, measurable targets.

4.5 Mixed Sequential-Iterative Design Process

Collecting all requirements, we need to develop a design process model with the following features (in order of appearance in the preceding analysis):

I Iterations between timetable, infrastructure, and demand.

II Iterations between service intention, network design, and demand.

III Definite commitment to a detailed infrastructure concept at the latest possible point in time.

IV Breakpoints to fix intermediate results and clearly defined scopes in each planning phase.

[16] Scheidt names a shift from railway operation to Light Rail (LRT) operation, but all shifts in mode of operation (signalled, direct traffic control, line-of-sight, etc.) are thinkable here.

[17] Scheidt considers only (comparatively microscopic) track layout infrastructure measures in this context and regards (macroscopic) open track measures to be part of external constraints, but the considerations also hold without this distinction.

V Interdisciplinary iteration across all fields of engineering involved in a design phase.

VI The ability to react on intermediate specification changes within the design process.

VII Broad sublevel design strategies to account for the incremental and iterative nature of the design process.

VIII A guidance of the whole process close to the demand.

IX Clear, measurable targets rather than interval, line, or route specifications as starting values.

Despite the drawbacks of a sequential design model and the clear preference for an iterative approach, we must not forget one key advantage of a sequential approach: Simplicity.

Within the discussion about the feasibility of the Waterfall Model, Larman et al. lead over to the difference of an engineering process and a political decision process:

> It's simple to explain and recall. 'Do the requirements, then design, and then implement.' IID is more complex to understand and describe. [...] It gives the illusion of an orderly, accountable, and measurable process, with simple [...] milestones [...]. (Larman et al. 2003: 55)

Bussieck expanded these findings to Railway Engineering:

> The disadvantages of this top-down approach are obvious, because the optimal output of a subtask which serves as the input of a subsequent task, will, in general, not result in an overall optimal solution. Nevertheless, this hierarchy decomposes the planning process in manageable segments and reflects the current internal structure of the railroad companies. Furthermore, it provides an integration into the classical temporal division consisting of strategic, tactical, and operational procedures [...] (Bussieck 1998: 6)

In these quotes, we can spot the attractiveness of a simple design process with clear milestones when viewed from the outside. In a Railway Design Process, we need to make use of this simplicity. We do actually have two completely different requirements concerning the process layout: (i) we need an iterative process on the engineering level. However, (ii) the design process is also a political decision process[18]. It will be bound to fail if policymakers—usually not railway experts—face this large, intermingled field where in-depth knowledge of all fields of expertise is required to fully understand the interdependencies.

Policymakers in the field of Railway Design might dislike the idea of spending money to see engineers seemingly handle a problem by "trial and error", which is essentially what an iterative process looks like from the outside (Curtis et al. 1987: 97). They need to be

[18] This becomes obvious if we recall that *Public Authorities* form an important stakeholder in figure 4.1.

offered some kind of waterfall model to fix decisions and enable engineers to proceed in the design process.

Furthermore, there is no such thing as *one* public authority. Railway infrastructure and railway operation are subject to several authorities, sometimes even jointly responsible for the various subsystems of railway infrastructure and operation. Figure 4.8 shows the public authorities involved in a regional railway operator with less than 100 km of tracks—but it sketches the complexity of having many political stakeholders. Having not only one, but *several* policymakers increases the need for a “simple” process significantly.

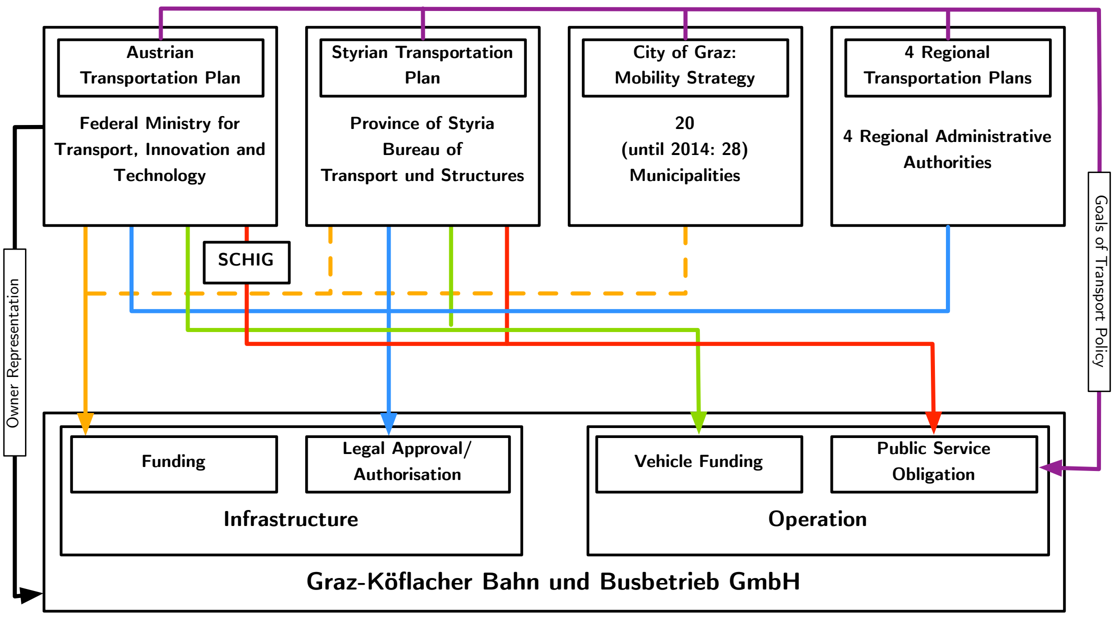


Figure 4.8: Example of a traditional stakeholder structure on the political level: Graz-Köflacher Eisenbahn (Walter 2016: 79)

We can summarise these findings in two conclusions about a railway design process:

on the engineering level, we need as little sequence and as much iteration as possible.

on the political level, we need as much sequence as possible and should avoid iteration completely.

Walter and Fellendorf tackled this by introducing a mixed sequential-iterative design process for regional railways. It includes several of the *Sublevel Strategies* as recommended by the U.S. Department of Energy and several iterative steps between demand, infrastructure, and timetable. But nevertheless, political decisions are made at distinct interfaces between design steps, named *Milestones* just as in waterfall models. Whatever happens between these steps is designed to accommodate as much interdisciplinary iteration as possible, but at milestones, distinct political decisions are jointly asked for (Walter and Fellendorf 2015: 39ff.).

We adapt this model for our purposes in an abstract way first, but we will refine it within the design methodology of this work. Figure 4.9 shows the key elements of this process. All political decisions take place between the *Phases*, which, for their part, include *Tasks* of all fields of engineering involved. Policymakers are supplied with distinct intermediate results which they can decide upon. The interdisciplinary, iterative loops are to settle the mutual dependencies and coordinations between the fields of engineering involved. Results can therefore, after the phases, be packed into distinct alternatives as the basis of decisions without the need to revise the whole decision process.

The process shown in figure 4.9 is composed of the following phases and tasks:

1. Target Definition

2. Service Intention Phase

3. Target Timetable Phase

4. Feasible Timetable Phase

5. Stage Development Phase

In each Phase, the level of detail increases as the level of abstraction decreases. The high level of abstraction concerning the infrastructure in the first phases is required to retain an overview of the systematic correlations within timetable and demand. By the end of the design process, however, all infrastructure elements need to be known in detail, in order to allow for a thorough measure planning and evaluation.

Figure 4.10 shows the relations between the level of detail, the individual Design Phases, and the possibilities of influence. As can be seen, as the level of detail knowledge increases, the possibility of influence from outside, i. e. usually traffic policy modifications, decreases.

The items listed for *Extertion of Influence* are, with the exception of those listed above the Service Intention Phase, *not* desired in this design process; as can be seen, the items listed are to stem from the design process rather than from the outside. However, as noted before, when dealing with policymakers, boundary conditions are prone to midway changes. The best a design process can deliver in this context is a coping strategy. Therefore, the influences listed *can* be incorporated in the corresponding Phases, but are *desired* to come from within the design process.

However, this figure also implies that policy changes that occur *after* the point stated in this list will considerably affect the implementation timeline and provoke stranded investments.

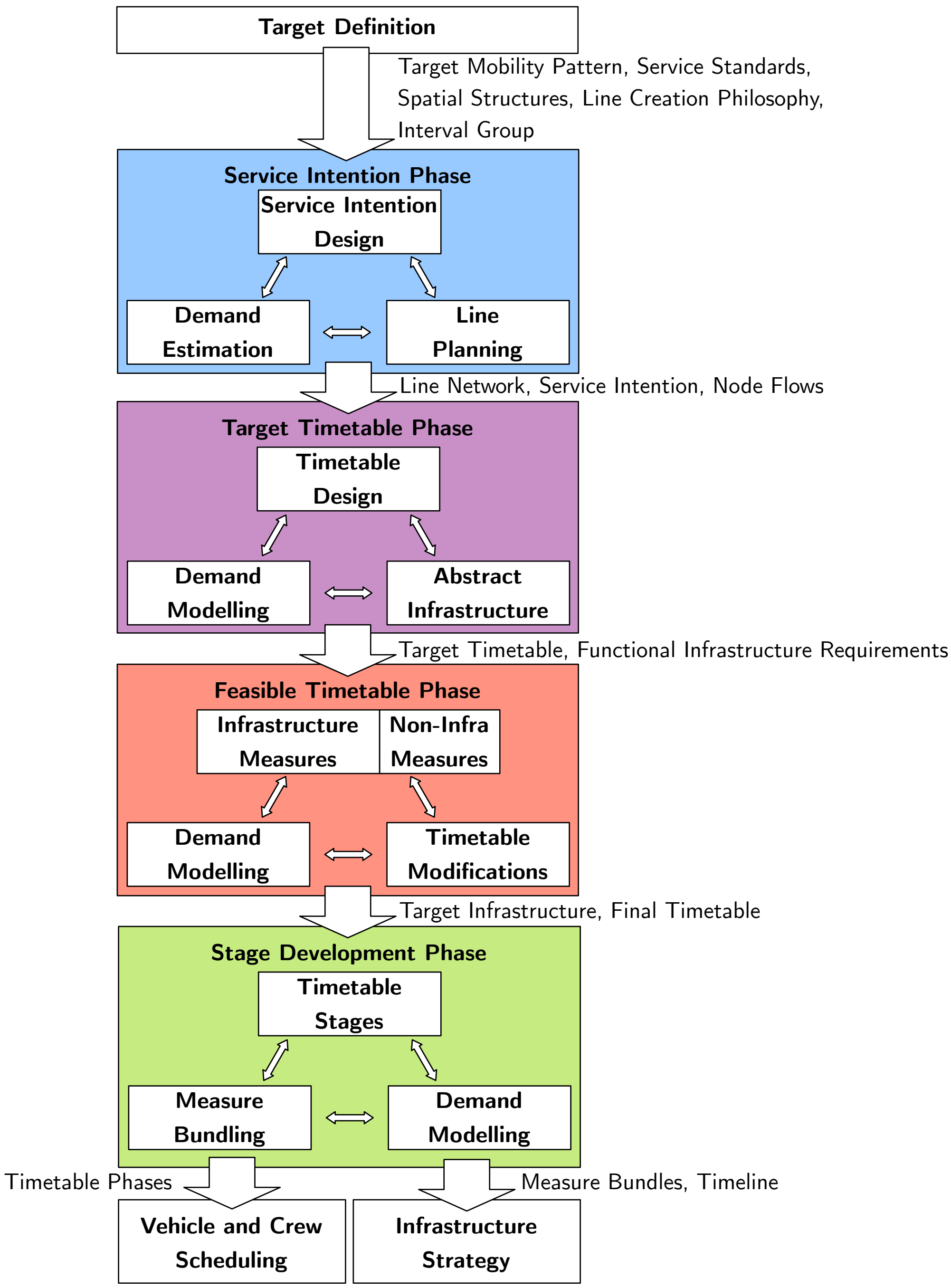


Figure 4.9: Mixed Sequential-Iterative Design Model

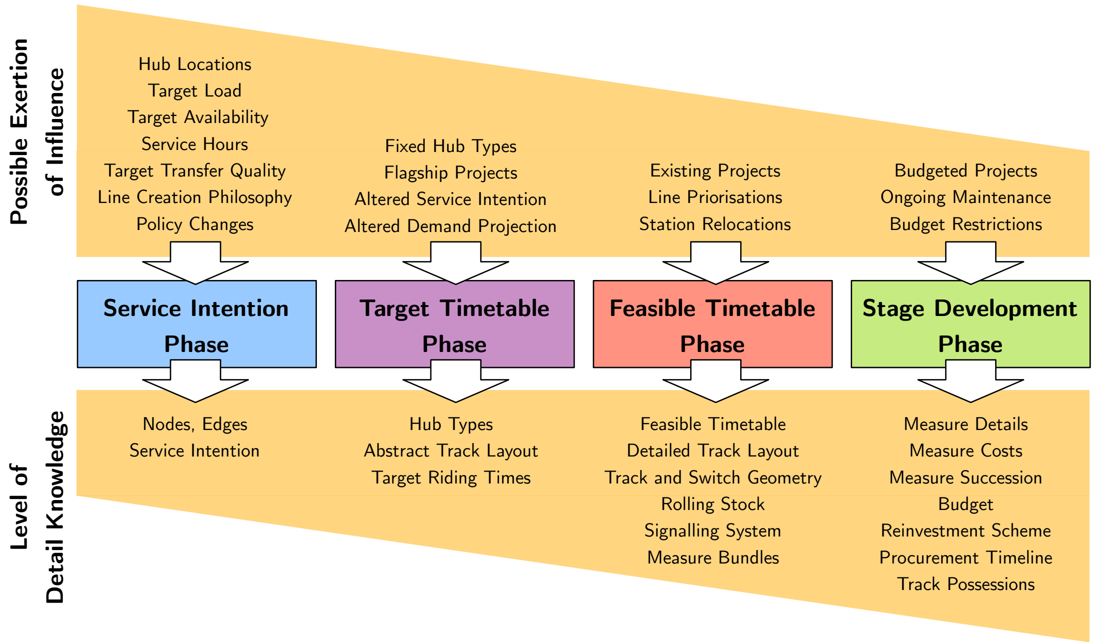


Figure 4.10: Possibility of outside influence and level of detail in the mixed sequential-iterative design process.

Target Definition

Input: none
Output: Target Mobility Pattern, Service Standards, Spatial Structures, Line Creation Philosophy, Interval Group

Before a design process can start, targets must be set. Usually, some kind of *Target Mobility Pattern*, such as target modal split, target ridership, target passenger kilometres and the like will be present as part of the *Service Standards.*

However, these *Service Standards* can and should go further, so as to put the goals of line planning and service intention beyond discussion: When target load, target area of service, service hours, and target connection quality are defined politically beforehand, line planning and a definition of the service intention are considerably easier (Walter 2010: 22ff.).

Spatial Structures need to be passed on to line planning and service intention design. These include (i) the basic layout of settlements in a network context, (ii) the location of local supply centres, and (iii) the spatial pattern of urban areas (Smoliner et al. 2015).

Key aspects of line planning and service intention will yield completely different results, even on the same data base, wenn the *Line Creation Philosophy* is different. There is no such thing as an optimal design philosophy, because the various approaches feature different advantages and drawbacks, which can, at best, be ranked subjectively from case to case (Walter 2010: 25ff.).

Finally, the *Interval Group* is a necessary parameter to be passed over at the beginning, since it defines the further line planning. It makes a distinct difference whether the network is based upon multiples of 15 minutes or multiples of 10 minutes[19], since the discrete steps of intervals result in significantly different demand reactions and capacities (Sparmann 2006: 12, Walter 2010: 42).

Service Intention Phase

Input: Target Mobility Pattern, Service Standards, Spacial Structures, Line Creation Philosophy, Interval Group
Output: Line Network, Service Intention, Node Flows

This phase is to fix lines and their respective intervals. As elaborated in section 4.1, *Line Planning* and *Service Intention Design*, i.e. interval planning, are tightly connected to each other. Either need to be evaluated by *Demand Modelling*, such that the mutual effects of line and interval planning can be quantified. At this stage, however, it is sufficient for a demand model to carry out interval-based traffic assignments.

It must be noted at this point that demand modelling, where applicable, is to be carried out as thoroughly as necessary at any step. Within this work, it is sufficient to subdivide demand modelling in (i) whether it is *dynamic*, i. e. mode choice is recalculated at each step, or *static*, i. e. OD matrices are invariant; (ii) whether the whole *network* or just *node flows* are considered; and (iii) whether the demand assignment is carried out *headway based* or *timetable based*. Table 4.2 shows the respective ways of demand modelling per Design Phase.

	Service Intention	**Target Timetable**	**Feasible Timetable**	**Stage Development**
OD matrices	dynamic	static	dynamic	dynamic
spatial scope	network	nodes	network	network
assignment	headway	–	timetable	timetable

Table 4.2: Demand modelling in the Design Phases

The two parameters to be handed over to the next phase are the *Line Network* and the corresponding *Service Intention*. All lines should be known with their corresponding

[19] not to mention multiples of 12, which are ruled out here for their unattractiveness (Walter 2010: 42f.)

train type (see section 2.2.4) and interval. Furthermore, we need the knowledge about (i) where lines are to part each other's intervals, (ii) where trains are to be split and coupled, (iii) which transfer relations are to be prioritised, and the like.

This phase requires several sensitivity analyses to be taken. These come into question when several competing solutions for a problem are possible, such as (i) whether to skip smaller stops completely and run faster (Walter and Fellendorf 2015: 39ff), (ii) whether to offer a homogenous service with denser intervals or a diversified one with less dense intervals per train type (Walter 2016: 77ff), (iii) whether to change a single hub into a hub pair (see section 5.1.4), and the like. All of these questions require constant communication between service intention, line planning, and demand estimation.

There might, however, be several competing service concepts with individual advantages and drawbacks that are to be decided upon politically—after the completion of this engineering phase, in the political discussion.

For a more detailed discussion of this Design Phase, refer to Walter 2010.

Target Timetable Phase

> *Input: Line Network, Service Intention, Node Flows*
> *Output: Target Timetable, Functional Infrastructure Requirements*

This phase is the first complete iteration between demand, timetable, and infrastructure. From the *Line Network* and the *Service Intention* we first create a model integrated timetable (*Timetable Design*). We then examine this timetable's implications in the *Demand Modelling*, this time in terms of static node flows. Then, we need to derive a feasible, yet still *Abstract Infrastructure* for this timetable, such that we can accommodate this timetable. We will, at this stage, already face timetable infeasibilities, especially when different train systems are to be fit onto one infrastructure or when certain edges between hubs are too short or too long to allow for a proper timetable design. This requires a different timetable model, which in turn makes for a different demand reaction.

This phase will be discussed elaborately in chapter 5.

The result of this phase is a *Target Timetable*, i. e. a timetable that fulfils the service intention best and that *probably* can be run on the abstract infrastructure constructed. For the infrastructure, we hand over *Functional Infrastructure Requirements*, such as (i) section riding times, (ii) approximate situation of double-tracked sections, (iii) location of tracks for parallel entry and exit, (iv) approximate location of overtaking facilities, and the like.

Feasible Timetable Phase

> *Input: Target Timetable, Functional Infrastructure Requirements*
> *Output: Target Infrastructure, Final Timetable*

This phase features the most intense interaction and iteration between all fields of engineering involved. The *Target Timetable* handed over from the last phase is the starting point for all *Measures*. Traditionally, these will be infrastructure measures, but there is a range of possibilities to allow for (i) riding time reductions or (ii) capacity enhancements that reduce the need for riding time reductions elsewhere (Walter and Fellendorf 2015: 39ff.). All measures at that point need to be elaborated up to timetable relevance and a cost estimate. Measures can then be packed into bundles that, summed up, allow for the desired timetable to work.

But again, external circumstances will render some timetable features impossible, so we need *Timetable Modifications* so as to keep as many features of the target timetable working. Some of these modifications might go as far as to require a reworking of a part of the network. In any case, any timetable relevant change needs to be investigated by *Demand Modelling* again—restarting the iteration until both timetable and infrastructure match.

This phase will be elaborated on in detail in chapter 6.

Stage Development Phase

> *Input: Target Infrastructure, Final Timetable*
> *Output: Timetable Phases to Vehicle and Crew Scheduling,*
> *Measure Bundles, Timeline to Infrastructure Strategy*

The last phase is to schedule the stages to finally reach the *Target Infrastructure*. Since railway infrastructure modifications feature long service lives and long design phases, a set of functional, intermediate combinations of timetable and infrastructure is desirable. Therefore, the measures from the first step need to be clustered in bundles that allow for these functional steps. Since the requested target infrastructure is not in place at that point, the *Timetable Stages* can only feature a reduced service compared to the feasible timetable (and possibly also to the service intention). To achieve the best intermediate benefit, the *Measure Bundles* are to be selected in a way that the completion of each bundle allows for the implementation of the next timetable stage. The bundles' order and the importance is to be assessed by *Demand Modelling* again, which will likely change the order and bundling again.

Finally, the *Timetable Phases* are handed over to Vehicle and Crew Scheduling well ahead of their implementation. The *Measure Bundles* and the *Timeline* together form the technical part of the *Infrastructure Strategy*, which can be used for financial negotiations and legal procedures.

5 Target Timetable

5.1 Hub Types

Within any public transport network, the purpose of a timetable hub is to ensure transfers. The approach found in the Integrated Timetable makes use of a basic property of any periodic timetable: Any two trips along a line in opposite directions will cross each other at the axes of symmetry, which are found twice as often as the interval of the line, at intervals of $T/2$. If all lines within a network run at the same interval T and if we can align the timetables such that these crossings occur at transfer hubs, we can ensure transfers from all directions in all directions at any hub. The sample network in figure 5.1 features three lines, all with the same interval T(in this case $T = 60$ minutes). Since the edge riding times between stations A, B, and C, respectively, are multiples of $T/2$, all trains will meet at these three points jointly, allowing for good transfer conditions in any direction.

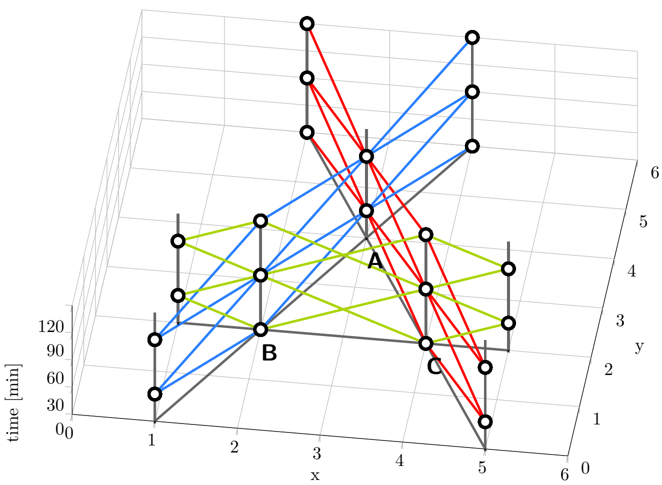


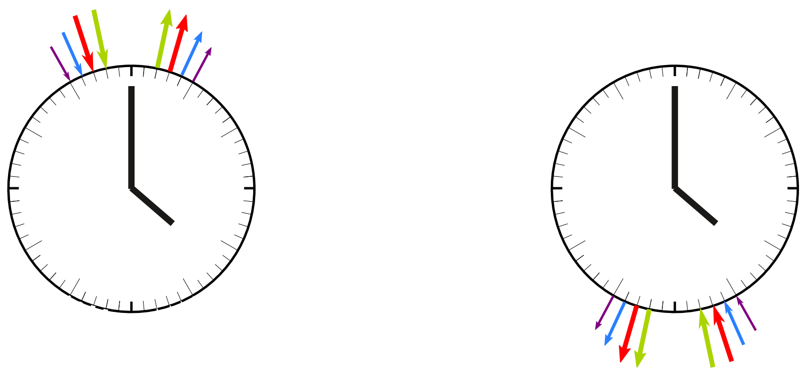
Figure 5.1: Sample network with Integrated Timetable hubs at stations A, B, and C

The prerequisites of the Integrated Timetable deal at large with *edge* properties (i. e. target riding times). However, when dealing with mixed traffic, there is no such thing as *one* target riding time per edge anymore. Rather we face a manifold of different riding times and a manifold of intervals per edge. We shall therefore focus on the transfer hubs first and deduct functional edge requirements from there.

When rigidly adhering to the comparatively strict requirements of the Integrated Timetable, a timetable can be constructed where (i) at all hubs, all transfers (ii) from any direction (iii) in any direction will work out. However, not all transfers at all hubs are actually relevant in a network view; this gives way to a larger set of possible hub constructions. The following sections deal with several approaches to relax these dependencies within a hub without considerably worsening the transfer conditions. From there, we will derive a variety of alternative hub constructions.

5.1.1 Full Hubs

A full hub shall be described here just for the sake of completeness. A hub of this kind, as omnipresent and marked with black circles in figure 5.1, is characterised by being reached by all train systems from all directions at the same time, thus allowing for transfers from every train system to every train system. Since a full hub can occur every half an interval, there are two instances of a full hub: one where all events happen at minute .00 and one after half an interval. Figures 5.2a and 5.2b show the respective hub clocks of such full hubs for hourly intervals, i. e. at minutes .00 and .30.



(a) Hub events at minute .00 (b) Hub events at minute .30

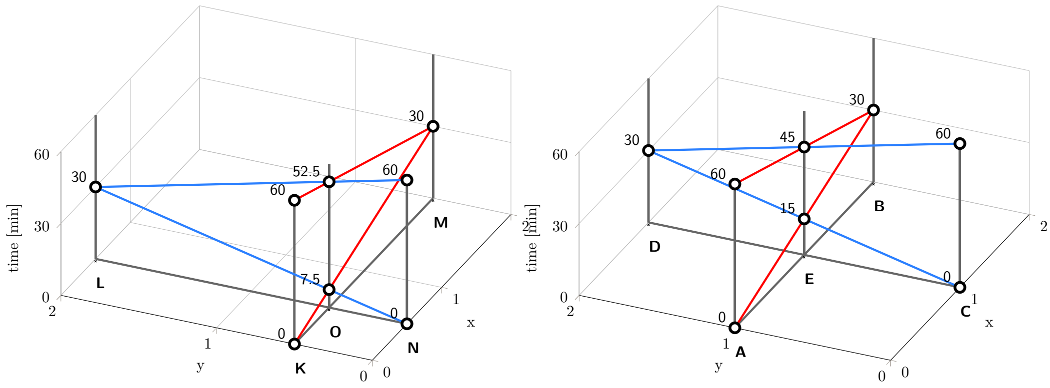
Figure 5.2: Hub clocks of full hub, hourly interval

5.1.2 Directional Transfers and Semi Hubs

Whenever a full timetable hub is not possible due to riding time or infrastructure restrictions, there is still the possibility of directional transfers. Since we require any line to obey the axis of symmetry (see chapter 2.3.3), two crossing lines will also serve a crossing station symmetrically. Any event in one direction will be mirrored across the axis of symmetry to form the corresponding event in the other direction.

If we align the crossing train trajectories to allow for a transfer between the lines in one direction, the transfer will therefore also be possible in the other.

Figure 5.3a shows a directional transfer in station O. Since transfers from station N to station M and from station K to station L are possible, so are the other directions due to the axis of symmetry. Likewise, the relations $N \leftrightarrow K$ and $M \leftrightarrow L$ are impossible[20] in either direction.



(a) Directional transfer (b) Semi hub

Figure 5.3: Transfer relations at directional transfers

Figure 5.3b shows a special case where the lines intersect each other right in the middle of their journeys, i. e. after a quarter of their intervals. We will refer to this situation as a *semi hub*.

A semi hub describes a situation on a line where departure and arrival events appear in equally spaced intervals, but in only one direction at a time. Topologically, this happens right in the middle of hub occurrences, meaning every half an interval ($T/2$), but shifted by a quarter of an interval ($T/4$), as shown in figure 5.4.

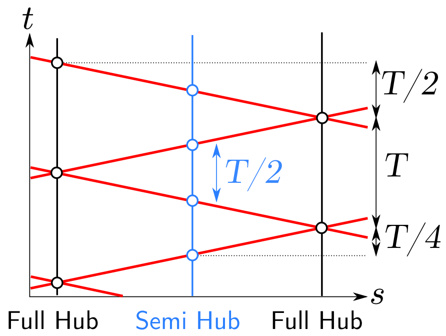


Figure 5.4: Time relations in a semi hub

Semi refers to the fact that only half the connections meet, and only half the transfers are possible in the hub, while transfers between all trains are possible in a full hub.

[20] i. e. require a significantly longer transfer time

Transfers for up to two lines in a semi hub can be offered across the platform[21], so the obvious advantage is that the hub spread time (see section 3.3.3) can be reduced. Even if route conflicts prevent a simultaneous arrival and departure, the necessary stop time overlap for transfers can be reduced to only accompany a cross-platform transfer, see figure 5.5.

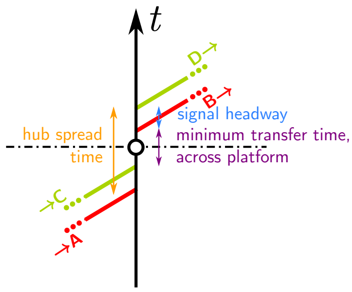


Figure 5.5: Reduced hub spread time in semi hubs

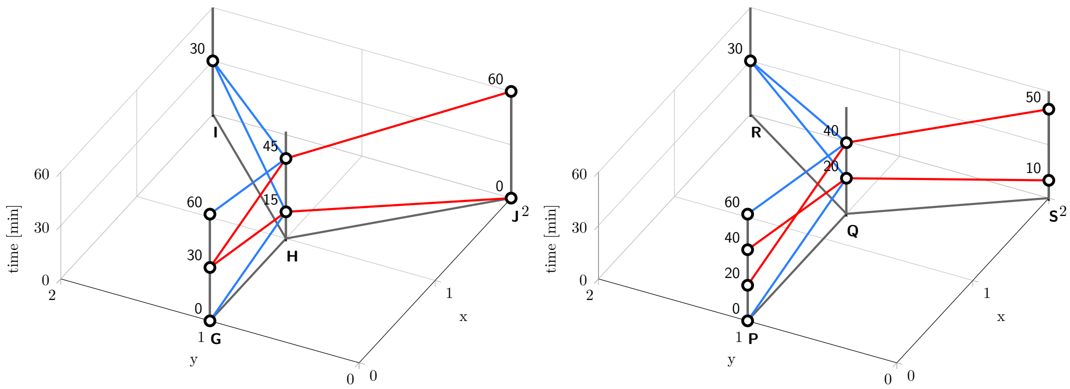
Semi Hubs at Line Branchings

A more special version of a semi hub can be used at line branchings. If two of the four branches from figure 5.3b overlap, they halve each other's intervals on a combined stretch. In this case, semi hubs serve as a more attractive transfer station than full hubs: Since the two lines share their interval on one stretch, they will be shifted by half the interval at any station. If the station at the branching were a full hub, this would mean that a transfer between the branches also takes half an interval. But if the station is a semi hub, this means that a train directed to one branch will meet there with a train coming from the other branch and a transfer is rendered possible. An ideal transfer between the branches will occur only where the trajectories of one line cross those of the other.

In figure 5.6a, the two lines coming from station G (full hub at minutes .00 and .30) will approach the branching station H at minutes .15 and .45. The train directed to station I will therefore meet the train coming from station J in station H, so a transfer from station I to station J and vice versa is possible.

As long as the lines share an equally spaced interval on the common stretch, the trajectories must also be equally spaced within the semi hub. When the semi hub is shifted along the line, transfers between the branches can only be accomplished if the interval on the common stretch is also distributed unevenly, see figure 5.6b. This is essentially a general form of a directional transfer (as in figure 5.3a) where two of four branches are folded into one.

[21] When using bay platforms, three or four lines can be accommodated on one platform. This, however, requires two lines to change direction in the station.



(a) Semi hub at line branchings

(b) Directional transfer at branchings with unevenly distributed interval on common stretch

Figure 5.6: Transfer relations at line branchings

The situation of a semi hub at line branchings can also be viewed from the perspective of the joint stretch rather than from the two lines: Due to half the interval there, this hub is actually a full hub on this stretch and a semi hub only when viewed from the perspective of the line branches.

When a service around a semi hub is overlain with a service at half the interval, a semi hub can serve as a full hub for this denser service; obviously, this works only when the departure and arrival events happen at equally spaced intervals.

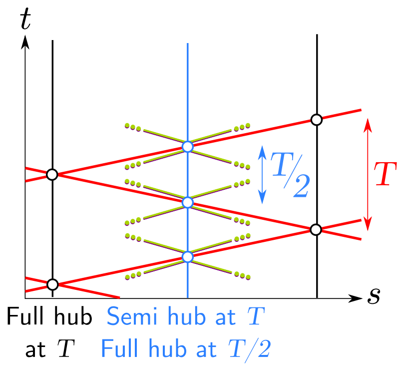


Figure 5.7: Semi hub at interval T, full hub at interval $T/2$

Figure 5.7 shows a semi hub for one line with interval T which also serves as a full hub for two other lines with interval $T/2$. Since no transfers between the directions of the former line are necessary, there is no loss of transfer attractiveness compared to a full hub for either train system.

Combined Semi Hubs and Full Hubs

However, when more lines serve the station as a semi hub only, the reduced transfer possibilities will remain amongst these lines. Figure 5.8 shows an example of a more complex hub structure. The lines *L1*↔*L2* and *L3*↔*L4* meet at station X at a quarter of their intervals ($T = 60\,\text{min}$, so they meet at minutes 15 and 45) to serve the station as a semi hub.

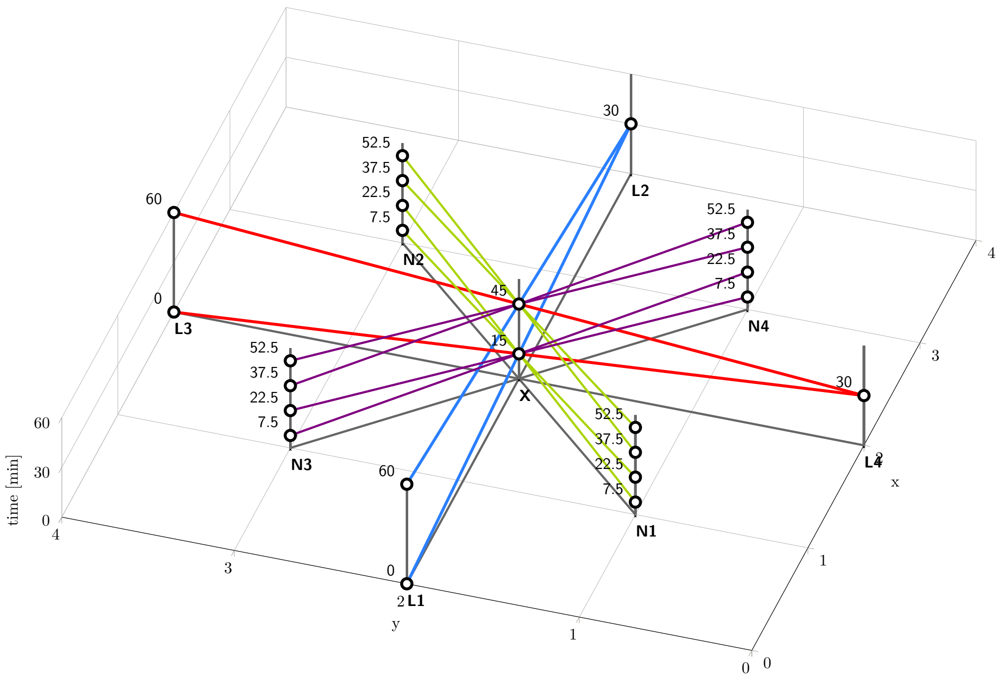


Figure 5.8: Complex hub with hub and semi hub function

As already noted in figure 5.6a, only transfers *L1*↔*L4* and *L2*↔*L3* can be offered with minimum transfer time. However, lines *N1*↔*N2* and *N3*↔*N4* run at half the interval ($T/2 = 30\,\text{min}$) and consequently meet at X at minutes 15 and 45 to serve a full hub. Any transfers between the two train system groups are possible without restrictions, as are transfers within the “N”-system. Restrictions only apply for the aforementioned “L”-system.

Directional Transfer at Utrecht

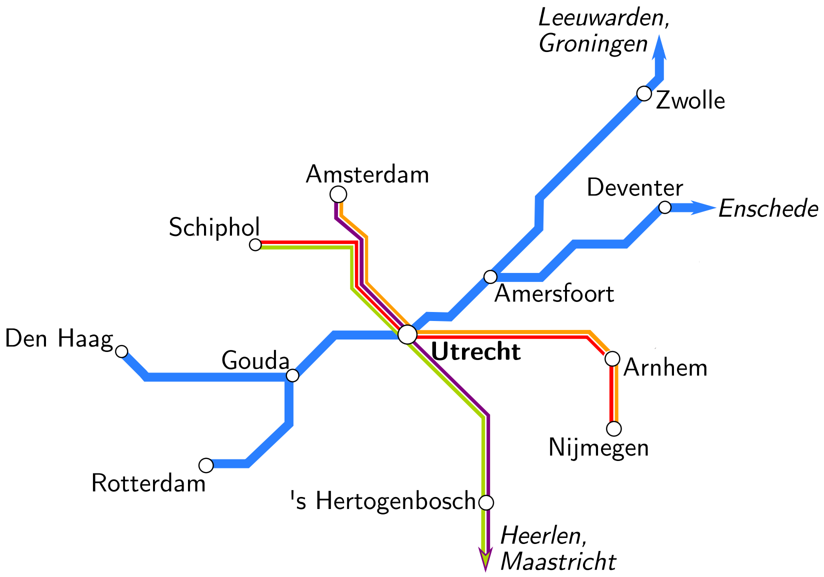
The city of Utrecht is the central hub of the Dutch railway system. Much like the Swiss system, the Dutch IC train network features dense intervals (half-hourly in most of the country and quarter-hourly within the *Randstad*-area around the major cities of *Amsterdam, Den Haag, Rotterdam, and Utrecht*). However, compared to Switzerland, the approach to an Integrated Timetable is different, featuring mostly directional transfers rather than full hubs.

Utrecht is located on the intersection of the *Intercity* lines from *Den Haag/Rotterdam to Enschede/Groningen/Leeuwarden* and from *Schipol/Amsterdam to Nijmegen/Heerlen/Maastricht*, respectively. Figure 5.9a shows the geographical relations and figure 5.9b shows the IC hub clock (i. e. long-distance trains only) of Utrecht. Colours used resemble the line colours of figure 5.9a, while opacity distinguishes the two respective riding directions. While the hub clock seems relatively dense and unsystematic at first sight, Utrecht is actually a set of overlapping directional transfer hubs.

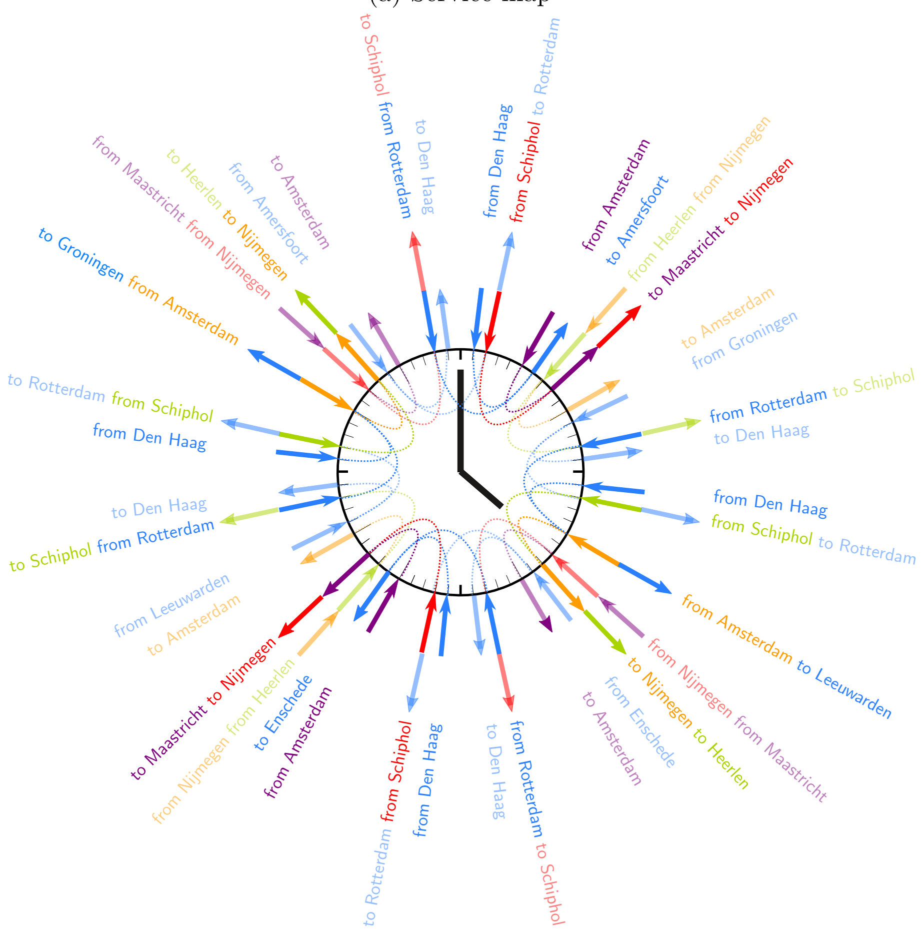
If we disintegrate the hub clock into (i) directions and (ii) lines, we can recognise more patterns. Figure 5.10a highlights only the eastbound line *Den Haag/Rotterdam–Enschede/Groningen/Leeuwarden*. We can see that trains from *Rotterdam* arrive first. Trains from *Den Haag* follow with signal headway, as both trains share their tracks between *Gouda* and *Utrecht*. They are coupled on the platform and continue their journeys jointly to *Amersfoort, Leeuwarden, Enschede*, or *Groningen*.

Figure 5.10b shows the same disintegration for the other four lines eastbound: *Schiphol–Nijmegen*, *Schiphol–Heerlen*, *Amsterdam–Nijmegen*, and *Amsterdam–Maastricht*. Trains from *Schipol* and *Amsterdam* share their tracks between *Bijlmer* and *Utrecht*, so the trains from *Schipol* arrive first and from *Amsterdam* second. These two trains, however, arrive at two adjacent edges of one platform to allow for transfers across the platform and leave the station simultaneously for *Nijmegen, Heerlen*, or *Maastricht*. The branches west and east of Utrecht are permuted, so there are four different origin-destination pairs, i. e. lines. Due to the platform transfer, all four relations are served at quarter-hourly intervals[22], while every other trip includes a transfer at *Utrecht*. Either of these transfers is a directional transfer in itself.

[22] The southern branch splits at *Sittard* to *Heerlen* and *Maastricht*, from which there are only half-hourly intervals. Since this is on the far south of the line, a further distinction is omitted here.

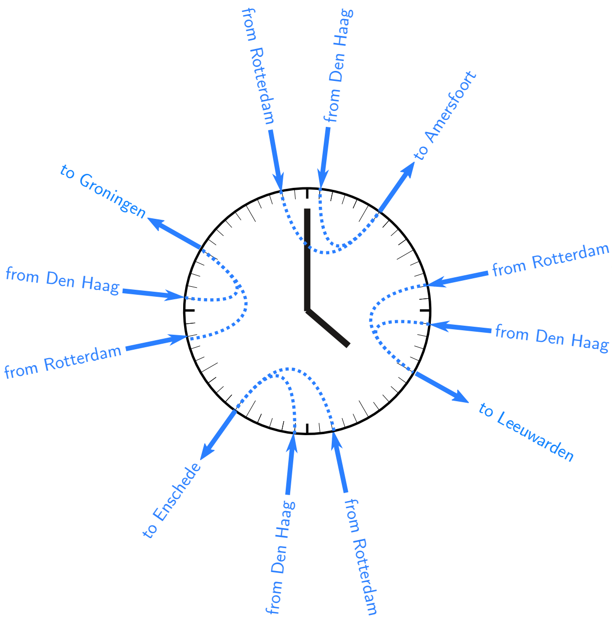


(a) Service map

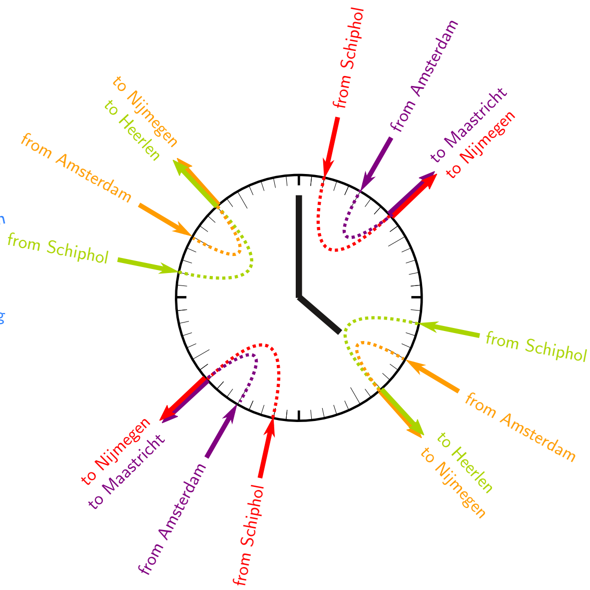


(b) Hub clock

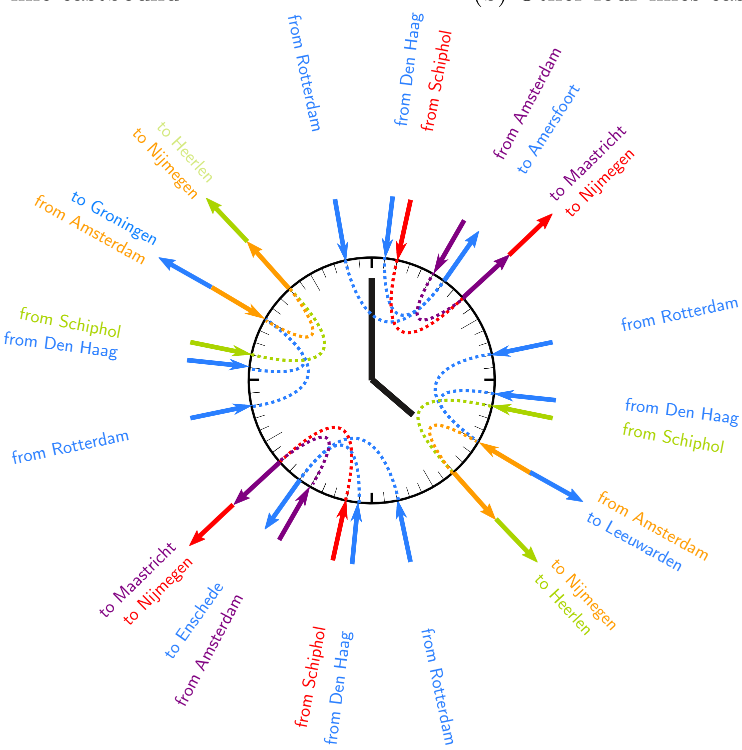
Figure 5.9: *Intercity* services in Utrecht, 2015 timetable.



(a) Blue line eastbound



(b) Other four lines eastbound



(c) All five lines eastbound

Figure 5.10: Disintegrated Utrecht hub clock, 2015 timetable.

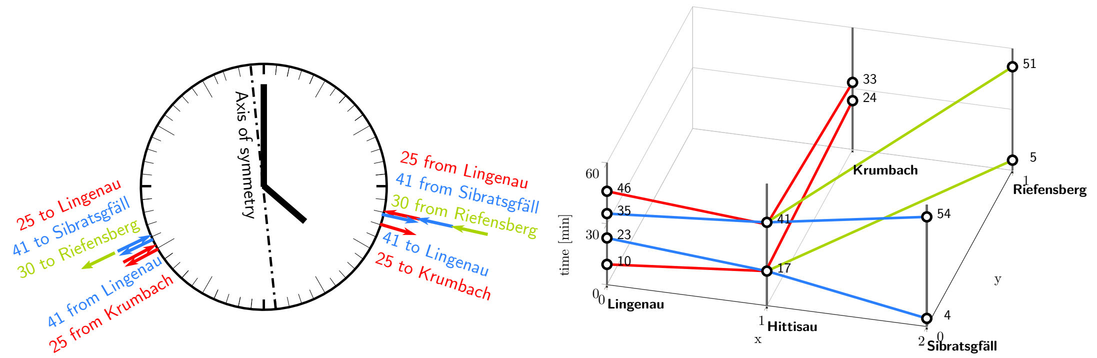
Figure 5.10c finally shows the complete extent of a directional transfer hub at Utrecht: If we jointly highlight the eastbound lines, we can spot a directional transfer hub between all lines[23, 24].

Directional Transfer at Hittisau

The *Wälderbus* concept in the Austrian province of Vorarlberg has been one of the first bus concepts with an Integrated Timetable. It connects a comparatively sparsely settled area with bus lines at hourly or half-hourly intervals. Among several variations of timetable hubs, we can also find a set of directional transfers of various types. One of these is *Hittisau*, as shown in figure 5.11b. It is mainly served by lines 25 on the route (*Bregenz*–)*Krumbach*–*Hittisau*–*Lingenau*(–*Egg*) and 41 on the route (*Dornbirn*–) *Lingenau*–*Hittisau*–*Sibratsgfäll*. Furthermore, line 30 from *Hittisau* to *Riefensberg* also connects to that hub.

The hub of *Hittisau* is constructed as a directional transfer at the branching of 25 and 41. Since the stretch between *Lingenau* and *Hittisau* is served by two lines, a transfer can be offered there both at minutes .17 and .41. Therefore, the only transfer not possible is between *Sibratsgfäll* and *Riefensberg*, because lines 41 and 30 both attach to the same directional hub at *Hittisau*.

As can be seen, the interval between *Lingenau* and *Hittisau* is distributed slightly unevenly to account for unevenly distributed riding times on the branches served by lines 25 and 41 on their own.



(a) Hub clock

(b) Line and timetable model

Figure 5.11: Directional transfer at *Hittisau*, 2015 timetable.

[23] The minimum transfer time between platforms in Utrecht has been defined as 3 minutes, and 2 minutes across a platform. Trains do not wait for transfers and the interval is dense enough to allow for this tighter approach.

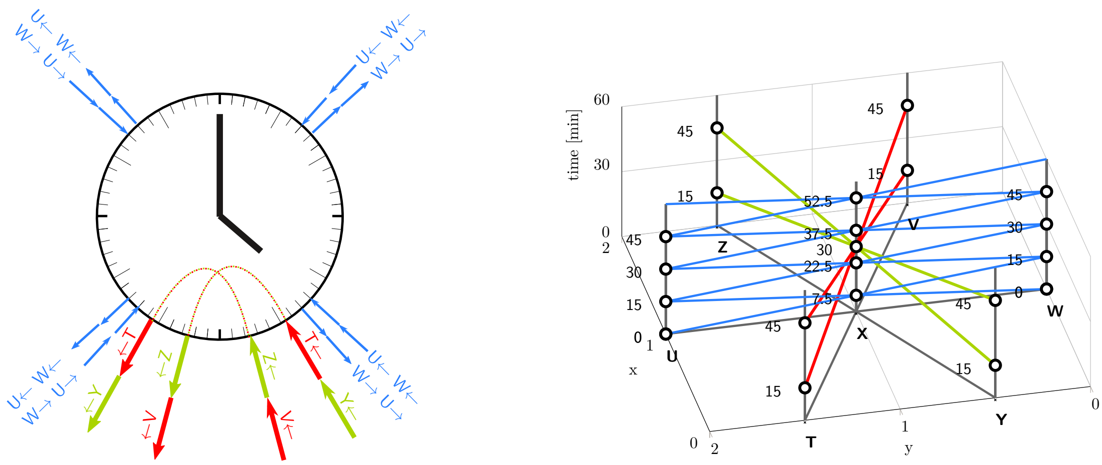
[24] The departures to Amersfoort/Enschede and Groningen/Leeuwarden are spread slightly unevenly, so there is no transfer from Amsterdam to Amersfoort/Enschede. But since these relations can be travelled faster via the direct link Amsterdam–Amersfoort, this is of no disadvantage.

5.1.3 Selectively Served Hubs

With the term *Selectively served hub*, one more hub type shall be introduced. *Selectively* refers to the fact that certain train systems do *not* serve the hub classically, but shifted by half their respective intervals.

As noted in section 3.3.3, a conventional service of a hub, i. e. arriving before and departing after the hub time, requires the less prioritised train systems to spend additional time at the hub. For train systems with relatively dense intervals (20 minutes and below for the example in figure 3.8 in section 3.3.3), this hub stop time amounts to values close to the interval, which means that the departure of one train will be close to the arrival of the next. For customers aboard that train, this means a riding time increase by almost one interval. Moreover, this hub service has significant drawbacks for both railway undertaking and infrastructure owner: (i) at least one trainset is always bound stopping at a hub, thus remaining unproductive and (ii) these unproductive trainsets continuously block platform tracks which are usually desperately needed in transfer hubs. Either drawback holds for bus, tram or other public transport services analogously.

Since the interval is close to this waiting time, the train system might as well *not* serve the hub traditionally (by waiting), but arrive *and* depart before the hub time with one train and arrive *and* depart with the subsequent train right after the hub time without an influence on the transfer time within the hub (see figure 5.12a).



(a) Hub clock (b) Network and timetable graph

Figure 5.12: Selectively served hub

We initially stated that an alternative service of the hub should *not* result in an aggravation of transfer times compared to the regular waiting time in the hub. Therefore, (i) the axis of symmetry needs to halve the train's interval, i. e. the hub spread time is to be equally spread around the hub clock, and (ii) the interval is to be short enough so as to serve the hub as tightly as possible.

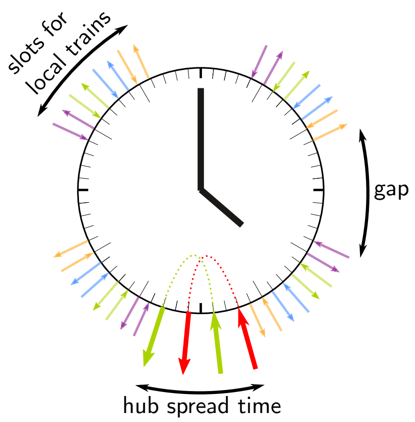


Figure 5.13: Densest setting for a selectively served hub

From a theoretical point of view, the densest possible slot allocation for the dense interval T_{dense} (i. e. the highest possible number of local train systems n_{Tr}) at selectively served hubs can be calculated as

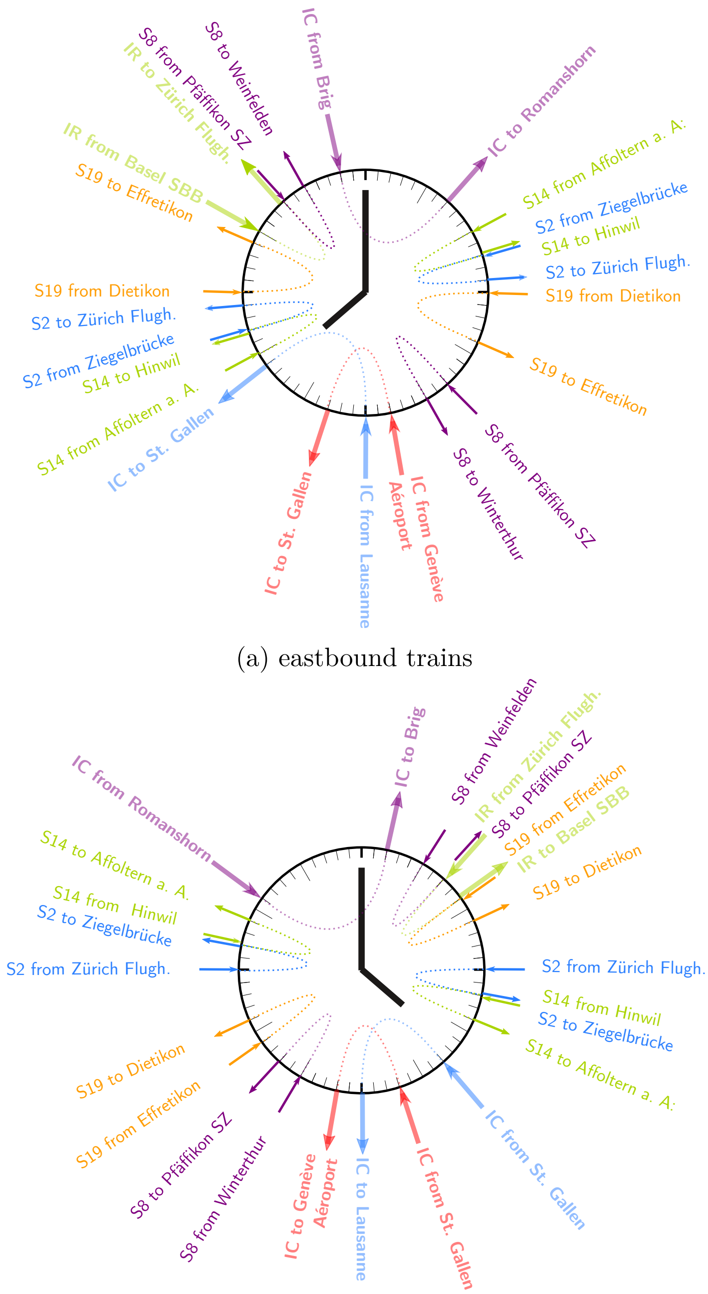
$$n_{\text{Tr}} = \lfloor \frac{T_{\text{dense}} - t_s}{t_{\text{signal headway}}} \rfloor \tag{5.1}$$

so for a 15-minute headway and six minutes hub spread time (as shown in figure 5.13), this amounts to $n_{\text{Tr}} = \lfloor (15-6)/2 \rfloor = 4$. Note that the interval of the long-distance trains does not matter in this view, since the gap created by one transfer hub repeats every interval of the local trains anyway.

Selectively Served Hub at Zürich

The *Durchmesserlinie* project in *Zürich* aims to directly connect the main train station *Zürich HB* with *Oerlikon*, *Zürich Flughafen*, and eastern Switzerland (*Winterthur, Romanshorn, St. Gallen*). The line is served by both long-distance trains and S-Bahn trains, the former of which serve it as a full timetable hub, while the latter serve it selectively. Figures 5.14a and 5.14b show the hub clocks of *Bahnhof Löwenstrasse*, the part of *Zürich HB* that serves the *Durchmesserlinie*.

In addition to the theoretical example given beforehand, the S-Bahn in *Zürich* offers a dense interval on the commonly served stretch, which interferes with the long-distance trains occupying the station during the hub time. Therefore, all S-Bahn departures have to take place in the slot between the full hubs (minutes .07–.28 and .39–.58 eastbound and minutes .02–.23 and .32–.51 westbound). This leads to an uneven distribution of departures (and no exact 15-minute headways) on the S-Bahn lines, such that the interval between *Zürich* and *Effretikon* (lines S8 and S19) is 4/26 (westbound) and 6/24 (eastbound) minutes and the interval between *Zürich* and*Pfäffikon SZ* (lines S2 and S8) is 10/20 (westbound) and 11/19 (eastbound) minutes. This phenomenon of packed departures and large timetable gaps on local trains is a direct result of the design of an



(a) eastbound trains

(b) westbound trains

Figure 5.14: Hub clocks of *Zürich HB*, *Bahnhof Löwenstrasse* part, 2016 timetable.

Integrated Timetable, if one trunk line has to simultaneously serve as a long-distance timetable hub and a densely served local traffic through station. The only way to improve the interval spacing of the S-Bahn trains in this case would be to lift the timetable hub for long-distance trains completely and serve it as a through station, as is the case with the IR from Basel to *Zürich Flughafen* already.

5.1.4 Hub Pair

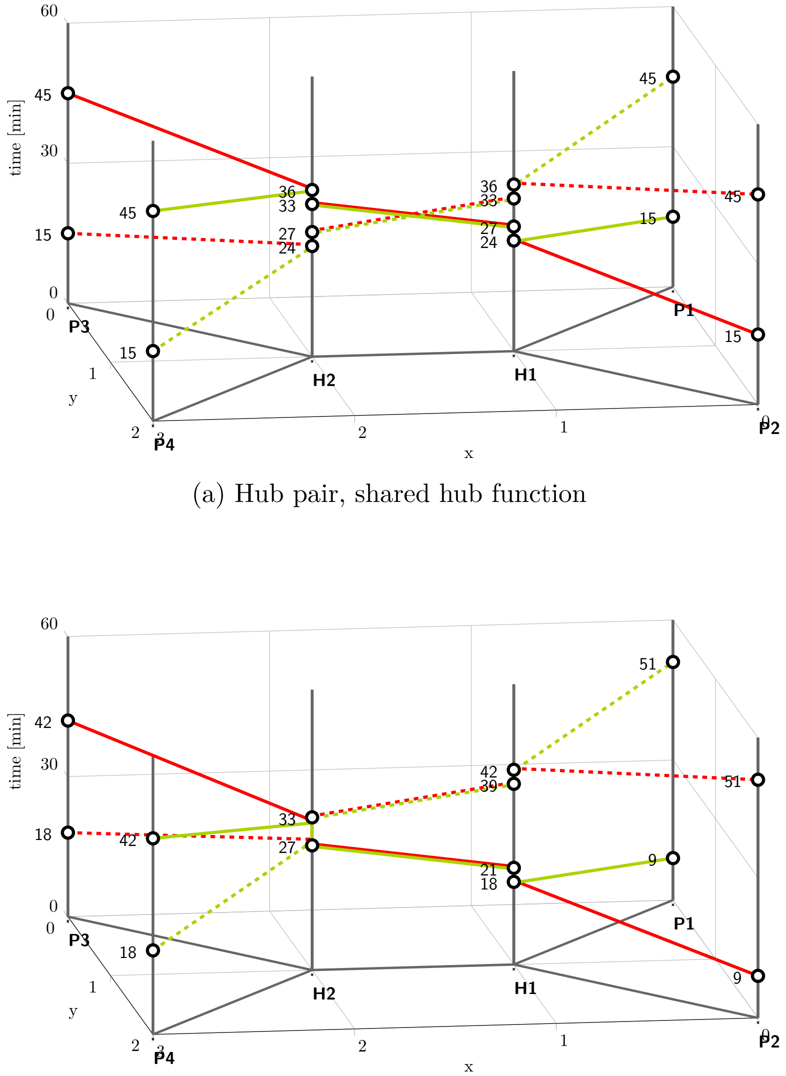
Rather than concentrating the transfer function of a hub in one station only, a *Hub Pair* shall denote a situation where the hub function is carried out together by two neighbouring stations. This means that a great portion of the hub stop time can be shifted to the itinerary between the two stations.

Figure 5.15a shows a hub pair with two lines. For easier understanding we assume completely grade-separated tracks with common platforms for lines in the same direction. In this system, trains from stations *P3* and *P4* arrive *and* depart before the hub time at hub *H2* before calling at hub *H1* after the hub time and continue to stations *P2* and *P1*, respectively. Transfers for onward travel can be done during the normal dwell time across the platform, while platform changes for remote transfers (*P1*↔*P2* and *P3*↔*P4*) can be accomplished while the trains are on the way to the other hub station. Topologically, the hub now lies in the centre of the stretch *H1*–*H2*.

One might also construct the timetable in such a way as to use only one of the stations as a timetable hub, rendering the other an adjacent station without distinctive function, as depicted in figure 5.15b. This does result in a clear and simple timetable design situation, whilst several disadvantages occur: (i) the hub stop times at the hub station increase to a minimum as shown in section 3.3.3. Furthermore, (ii) the overall riding time increases, since the dwell time at the second station will not be considerably shorter than the transfer time across the platform. Finally, (iii) while the remote transfer at one hub (hub *H2* in the given example) decreases by half the riding time between the two stations, it increases by half the riding time *plus* the additional transfer time at the other. This results in an overall increase of transfer time, provided the transfer passenger flows are similar.

Hub Pair at Wien

The main train station (*Hauptbahnhof*) of *Wien* serves as such a hub pair together with the neighboring station of *Meidling*. In long distance traffic (see figure 5.16), the hub pair is served by the ICE from *Frankfurt* to *Wien*, the RJ from *Zürich/München* via *Wien* to *Budapest/Flughafen Wien*, the IC from *Salzburg* via *Wien* to *Flughafen Wien*, the RJ from *Graz* via *Wien* to *Praha*, and the RJ from *Villach* to *Wien*. Generally speaking, trains from the west continue to the east, while trains from the south continue to the north. Upon completion of the *Pottendorfer Linie* rail link, the station will feature full



(a) Hub pair, shared hub function

(b) Hub with adjacent station

Figure 5.15: Possibilities of serving a hub pair

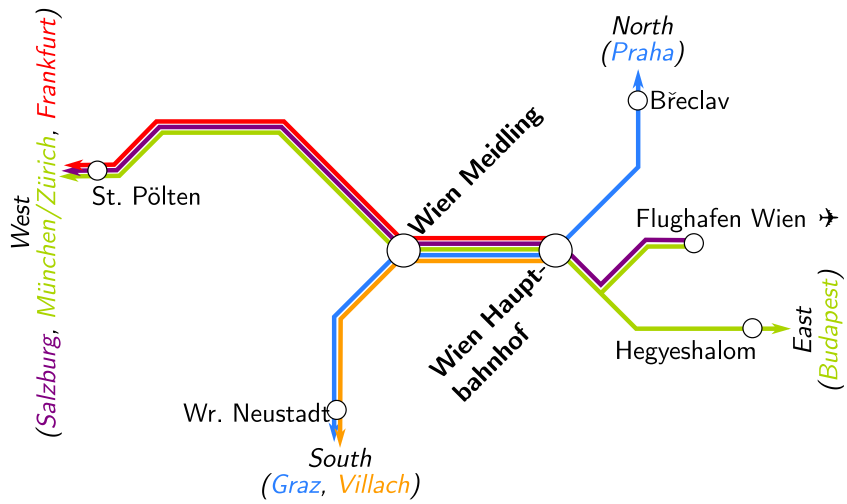


Figure 5.16: Long distance railway map around *Wien*

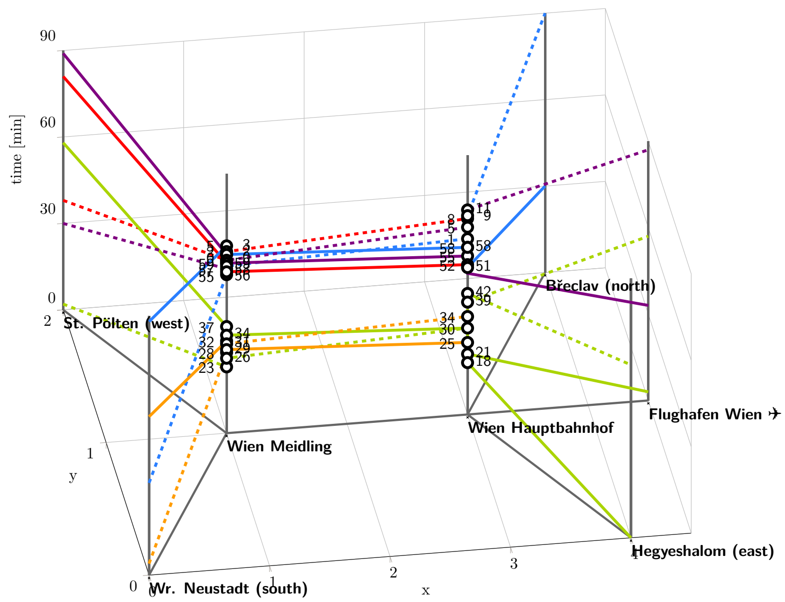


Figure 5.17: Network and timetable graph for the hub pair at *Wien*, 2016 timetable

grade separation. Up to then, there are still route conflicts between the western and the southern branch at *Meidling*, thwarting the hub pair advantages a little for the time being.

Figure 5.17 shows the line and timetable model of the hub pair. As can be seen, the trains follow each other at signal headway between the two stations and offer a transfer across the platform at *Hauptbahnhof*. Upon completion of the *Pottendorfer Linie* link, a service, as shown in figure 5.15a, will be possible.

Hub Pair at Egg/Müselbach

In the *Wälderbus* concept, a hub pair exists at *Müselbach* and *Egg*. The latter is the central village of the region and the main hub of the bus concept, but the hub service is slightly shifted to allow for improved transfer conditions at both stations. *Egg* is served by lines 25 (*Bregenz–*)*Lingenau–Egg*, 35 (*Bregenz–*)*Alberschwende–Müselbach–Egg–Schwarzenberg–Bersbuch*(*–Bezau*), 37 (*Bregenz–*)*Alberschwende–Müselbach–Egg–Bersbuch*(*–Mellau*), and 40 (*Dornbirn–*)*Alberschwende–Müselbach–Egg–Bersbuch*(*–Schoppernau*). In *Müselbach*, line 41 (*Dornbirn–*) *Alberschwende–Müselbach–Lingenau* (*–Sibratsgfäll*) branches off the main line. Lines 40/41 and 35/37 each jointly offer a half-hourly service between *Bregenz* and *Dornbirn*, respectively, and *Egg*. Figure 5.18 shows the hub pair *Egg–Müselbach*. Lines 35, 37, and 40 have no extra transfer time at *Müselbach* and continue for *Egg* to serve it as the other part of the hub pair. Line 41 spends an extra 2 minutes at *Müselbach* to make a transfer with line 37 coming from *Egg* and therefore renders the transfer *Egg–Müselbach–Lingenau* possible, which is useful since line 25 runs half an hour shifted from line 37.

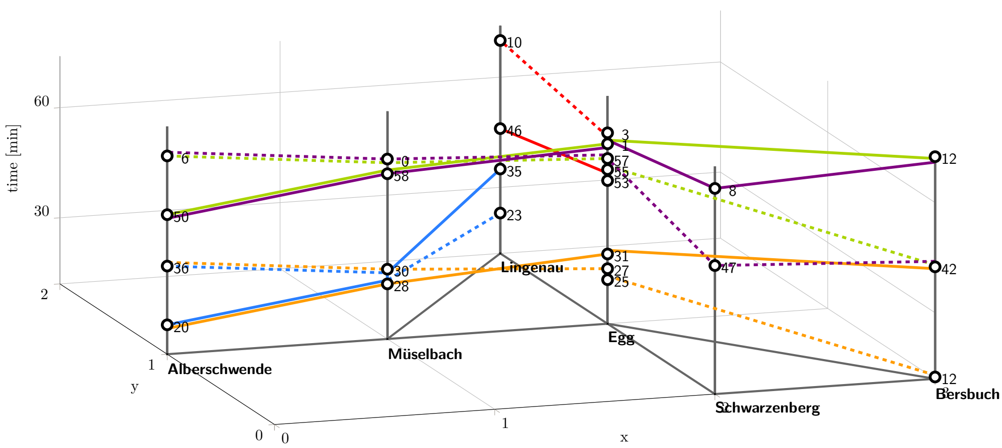
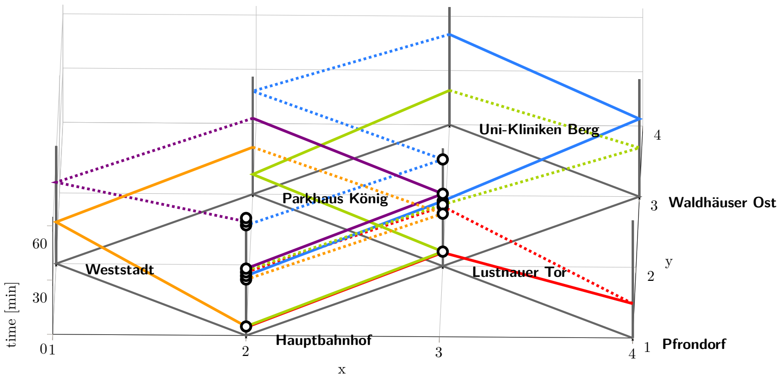


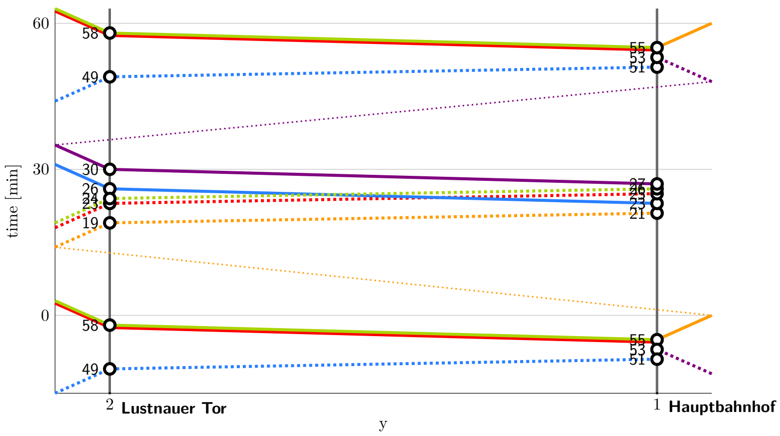
Figure 5.18: Hub pair at Egg and Müselbach, 2015 timetable

Hub Pair at Tübingen

One more application for hub pairs is for inadequate riding times on network parts. In *Tübingen*, the night bus system features loops negotiated within 60 minutes each. The main transfer hub is at *Hauptbahnhof* at minute .55. Loops of 30 minutes riding time are combined to 60-minutes circulations via *Hauptbahnhof*. Therefore, there is a secondary hub shifted by 30 minutes. Figure 5.19a shows part of the night bus network. Lines N91, N94, and N98 start at the main hub *Hauptbahnhof* at minute .55 and are supposed to end there 30 minutes later. Lines N91 and N94 exceed the required time of less than 30 minutes, ending at *Hauptbahnhof* at minutes .25 and .26. The secondary hub at .25 (see figure 5.19b) is stretched to form a hub pair between *Hauptbahnhof* and *Lustnauer Tor*. Line N97 can leave *Hauptbahnhof* at minute .30 to make a transfer there, while line N93 also exceeds the required riding time. Therefore, it starts at minute .23 already and makes the transfer with N91 and N94 at *Lustnauer Tor*.



(a) Night bus network overview



(b) Detail of Hauptbahnhof/Lustnauer Tor hub pair

Figure 5.19: Hub pair at Tübingen, 2010 timetable

5.2 Line Service

The approach presented in this work bases mainly on hub classification. However, there are several possibilities to actually serve an edge. Note that, in any case, the trajectory simplification as described in section 2.3.2 shall hold to retain the ability to process trajectories as described in this methodology.

5.2.1 Interval Parting

Interval parting, as already stated in the problem statement in section 2.3.4, describes a situation where n_{Tr} different train systems of interval T_i jointly serve a section in such a way that they split each other's intervals to T_{joint}, i. e. a joint interval $T_{\mathrm{joint}} = {}^{T_i}/_{n_{\mathrm{Tr}}}$ is created.

Halved intervals

Apart from inner city trunk lines (touched upon in the next section), the most common application of interval parting is two train systems *halving* each other's intervals.

Figure 5.20 shows such a situation. As can be seen, the green and the red train system each run at their respective intervals T_i on the outer branches A–X, B–X, C–Y, and D–Y, while they are shifted by ${}^{T_i}/_2$ on the common stretch X–Y, thus serving this part with the interval ${}^{T_i}/_2$.

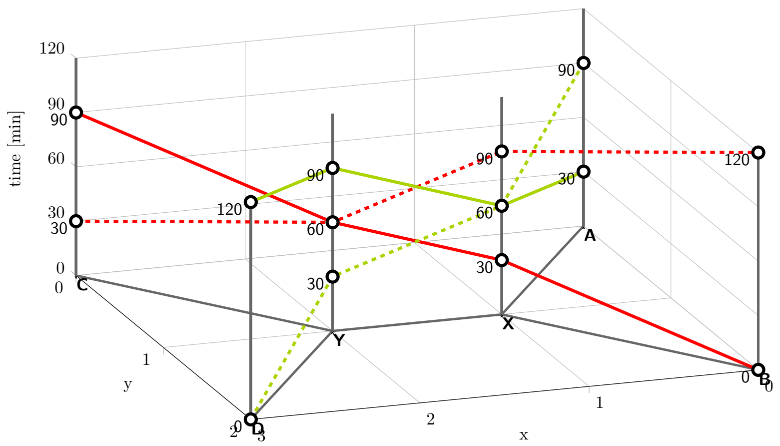


Figure 5.20: Interval parting on common edge

Topologically, the common stretch can also be viewed as a line with a denser interval, attaching lines with less dense intervals at the hubs at either end, be it as transfers or direct connections.

Halved Intervals South of Ulm

Interval partings can also be used for more complex arrangements of networks. Wieczorek and Bopp describe a target timetable solution for the area around *Ulm* where complex interval parting takes place on several levels (Bopp 2004 and Wieczorek 2006: 83). Figure 5.21 shows the network graph of this timetable.

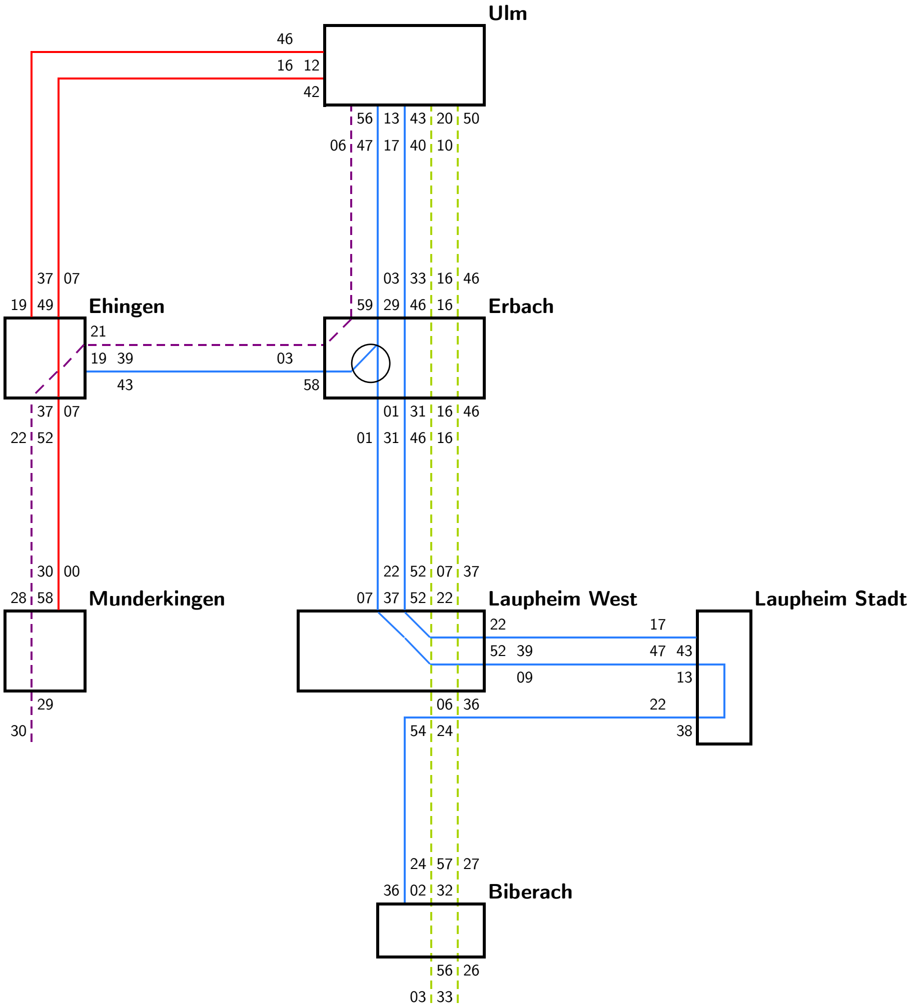


Figure 5.21: Network graph of a model target timetable as used by Bopp 2004 and Wieczorek 2006: 83

Figure 5.22 shows a network and timetable graph for the stretch between *Ulm* and *Biberach*.

As can be seen, the local train from *Ulm* to *Laupheim Stadt* parts its half-hourly interval with the express train between *Ulm* and *Biberach* on the edge between *Erbach* and *Laupheim West*. The express trains are faster on the stretches between *Ulm* and *Erbach*

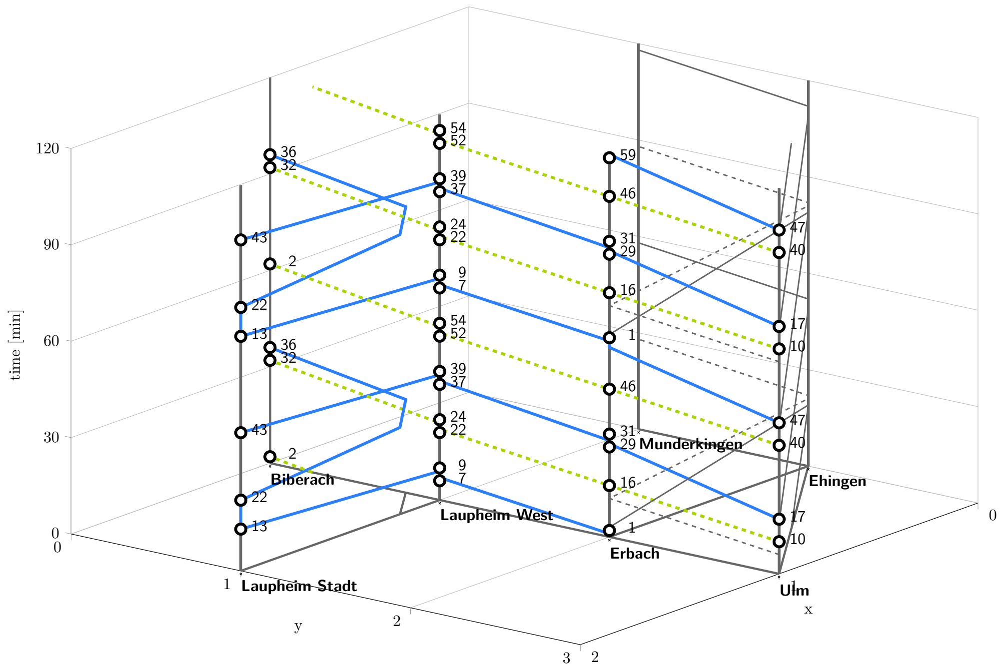


Figure 5.22: Network and timetable graph, trains on the stretch *Ulm–Biberach* highlighted in colour, as used by Bopp 2004 and Wieczorek 2006: 83

to allow for a connection with the full hub at *Ulm*, which they would not reach if they were as fast as the regional trains. They are also faster between *Laupheim West* and *Biberach*, while half of the local trains that split off at *Laupheim West*, turn at *Laupheim Stadt* and continue to *Biberach*, where they meet the express trains again to jointly serve a—albeit slightly shifted—full hub there. In total, both the hubs at *Ulm* and *Biberach* are served by the express trains, while the local trains serve *Biberach* only and double the interval on the aforementioned joint stretch[25].

Figure 5.23 shows the part between *Ulm* and *Munderkingen*. The local trains between *Ulm* and *Ehingen/Munderkingen* serve the same hub at minutes .15/.45 as the local trains to *Laupheim*, and the express trains serve the full hub. However, the express trains gain enough time before *Ehingen* to achieve an interval parting with the local trains on the stretch between *Ehingen* and *Munderkingen*, again serving a full hub there. The branch line (*Ulm–*)*Erbach–Ehingen* that also serves the .15/.45-hub at *Ulm* reaches *Ehingen* just right to make a connection with the express train. In total, passengers can profit from a denser interval on the common stretch, while the medium-distance passengers directed for the full hub at *Ulm* can take advantage of the lower riding time of the express trains.

[25]Note that there is, additionally, a semi hub at the line branching at *Laupheim West*: As can be seen in the network graph in figure 5.21, express trains from *Biberach* make connections to the local trains to *Laupheim Stadt* and vice versa.

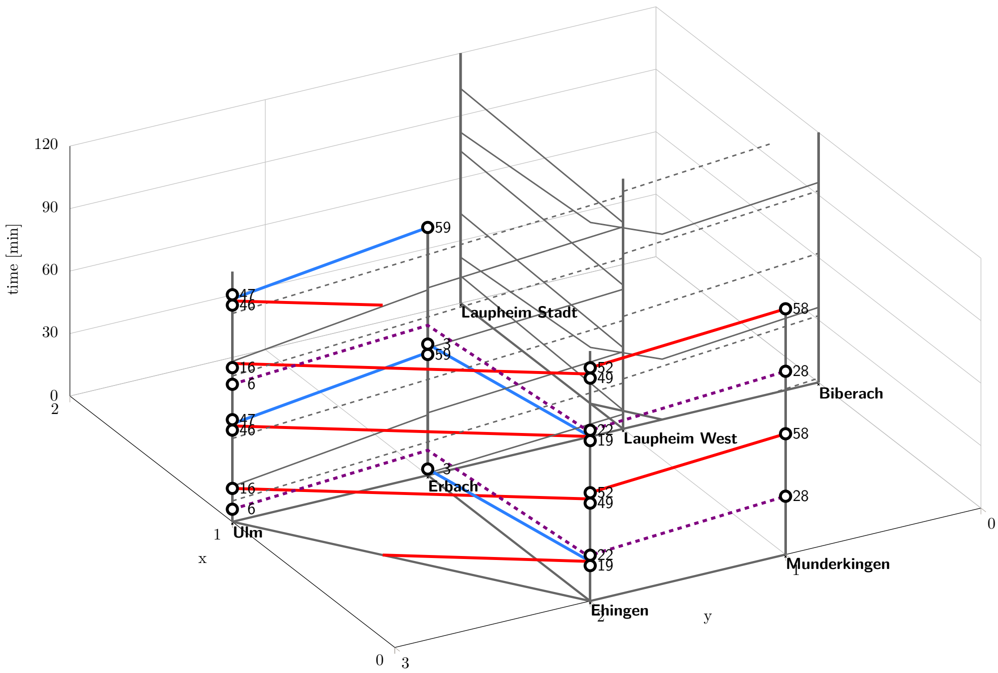


Figure 5.23: Network and timetable graph, trains on the stretch *Ulm–Munderkingen* highlighted in colour, as used by Bopp 2004 and Wieczorek 2006: 83

Inner City Trunk Lines

A more sophisticated version of an interval parting can be found on many inner city trunk lines of *S-Bahn* and similar systems where regional railway lines jointly serve a common stretch through the inner city, offering a much denser interval there, while retaining direct trips to the outer branches. [26]

In any case, the design principles of the outer branches differ significantly from those on the central stretch. The latter resembles more a classical metro, with a focus on regular, dense intervals, a need for a high grade of interval stability, and no need for timed transfers or a differentiated train hierarchy. Infrastructural measures focus on route capacity, passenger thoughput, and the avoidance of route conflicts. The outer branches need to be designed, by the nature of the less dense interval, according to classical regional railway principles, i. e. with a focus on organised transfer conditions and riding times. Infrastructural measures will rather focus on crossings and overtakings, station location, and transfer station design.

[26]Note that from a timetable construction point of view, there is no difference in whether the trunk line is considered *one* line with branchings (such as the *RER* lines in greater *Paris*) or as one combined stretch of many *different* lines (such as *S-Bahn*, *S-tog* in *København*, *Pendeltåg* in *Stockholm* and *Göteborg*, *Metropolitano* in *Napoli*, and others).

However, either principles need to be combined in this kind of service concept. Stohler et al. proposed a design from the regional branches moving inwards, thus obtaining a proper hub service there, leaving the inner city service as a result of the outer hubs (Stohler 2008: 108, Stohler et al. 2012: 31). Given that the possibility for variation is comparatively low on the outer branches, this design direction results in less need for feedback loops. However, upon reaching the trunk line, the arrivals and the departures still have to be carefully filed into the timetable to guarantee a regular interval. This, in turn, calls for an evaluation of which feeder branch can offer which amount of riding time modification to allow for a regular interval on the trunk line.

Wien S-Bahn Inner City Trunk Line

Amongst many, the *Stammstrecke* inner city trunk line at *Wien* is an example for the extensive use of interval parting. Several S-Bahn and R lines, each at half-hourly or hourly intervals, join to offer a three-minute headway on a central stretch.

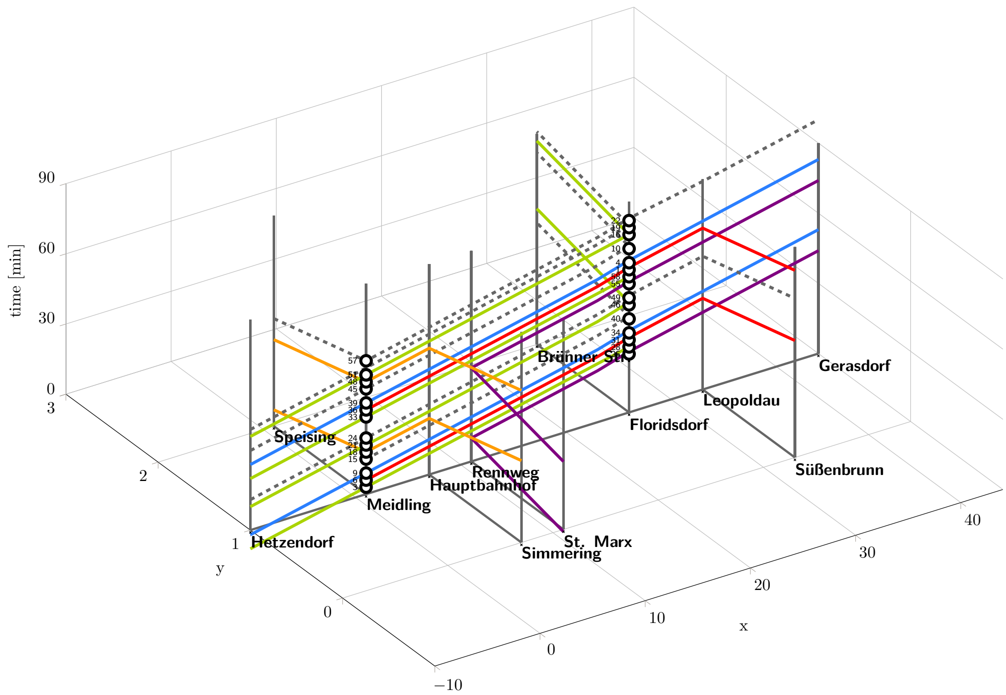


Figure 5.24: Network and timetable graph of the *Wien* inner city trunk line, afternoon peak in the 2016 timetable.

Figure 5.24 shows network and timetable graph of the afternoon peak along the inner city trunk line of *Wien*, which runs from *Meidling* to *Floridsdorf*. It features branches from *Hetzendorf* (lines S2, S3, and S4) and *Speising* (line S80) towards *Meidling*, from *Hauptbahnhof* to *Simmering* (line S80), from *St. Marx* to *Rennweg* (Linie S7), and from

Floridsdorf to *Brünner Straße* (lines S3 and S4) as well as to *Leopoldau* and further to *Gerasdorf* (line S2) and *Süßenbrunn* (line S1). Furthermore, three regional express lines (R) traverse the trunk line along the same, homogenous, trajectory as the S-Bahn lines, calling at all intermediate stops, but continue to the outer branches as express lines. In total, an interval of at least three minutes can be achieved, but not all slots are filled.

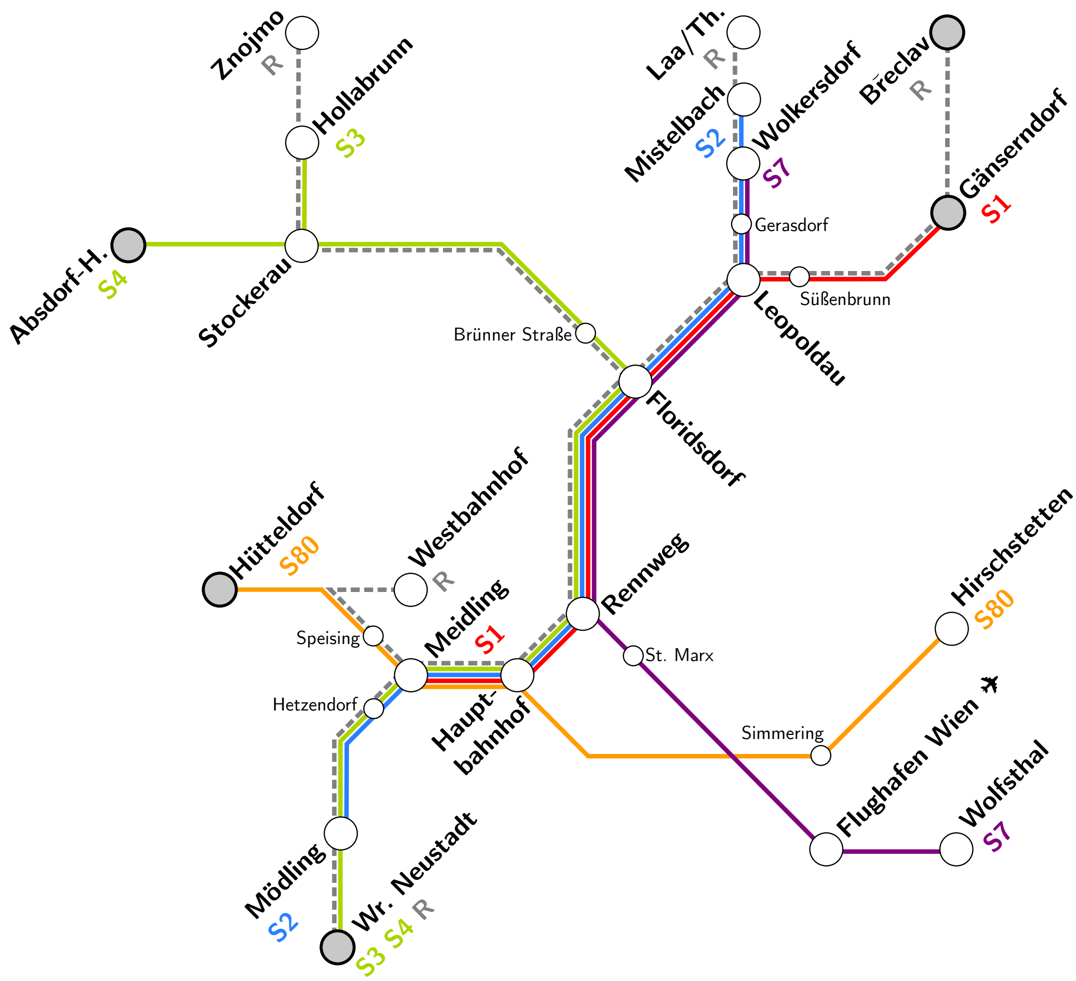


Figure 5.25: Network map of the network around Wien inner city trunk line.

From a network-wide perspective, the dependencies of the regional branches moving inwards become obvious: Figure 5.25 shows the part of the *Wien* S-Bahn network connected to the inner city trunk line. The stations marked in grey (*Absdorf-Hippersdorf* (line S4), *Břeclav* (R), *Gänserndorf* (S1 and R), *Hütteldorf* (S80), and *Wiener Neustadt* (S3, S4, and R) are important hubs of long-distance and regional traffic[27] and are mostly

[27] Of course, *Westbahnhof* is, too; but for the R train system, there is no need for a timetable integration there.

organised as full timetable hubs, so there is little flexibility as to how they are served. S-Bahn lines are designed to serve the hubs and subsequently pass their trajectories inwards, until they are fed into the trunk line. Since not all slots in the three-minute headway framework are filled in the system timetable, some flexibility remains as to which slot is reached, and remaining gaps in the timetable are gradually filled with singular extra journeys until all trajectories are filled in the morning peak hours.

5.2.2 Train Hierarchy

Since this work aims to develop design principles for mixed-traffic passenger traffic, i. e. trains of different riding times and different intervals on one network, the *train hierarchy* needs to be tackled.

Independent of the actual naming of different train types, a hierarchy is useful for constructing a differentiated service offer. Walter proposed the use of at least an internal hierarchy for any network planning task, so as to obtain a guideline to systematically design public transport lines. This includes hours of operation, prioritisation in the design process, variation of the interval, use of rolling stock and treatment in dispatching (Walter 2010: 26f.).

Wien-Wiener Neustadt Train Hierarchy

Figure 5.26 shows the train graph of the *Südbahn* line between *Wiener Neustadt* and *Wien*. Right away, we can spot three train types: REX/D trains, marked in thick, dotted lines; express trains (R/R/R), marked in thin, dashed lines; and S trains, marked in thin, continuous lines. The REX/D trains run four times an hour without an intermediate stop, the S trains run twice an hour from *Wiener Neustadt* to *Wien*, another two times an hour from *Mödling* to *Wien* and another two times an hour from *Wien Liesing* to *Wien*, making a total of six trips per hour arriving at Wien. The express trains (R/R/R) feature three different combinations of riding time and stopping pattern: the slowest R trains only skip 5 of 19 stops and run alternating to the S services (with 63 minutes of riding time compared to 70 of the S trains). The R system features a riding time of 48 minutes and features 8 stops, while finally the R system makes the journey in 45 minutes with 6 stops. In total, passengers are offered six different half-hourly train systems between *Wiener Neustadt* and *Wien*, each with a different service concept.

Within the design process, there is a high level of interdependency between these train systems; if tackled with a hierarchical design process, this dense operational concept would not be possible. Compared to the operational concept throughout the day, the fastest slots are slower by four minutes to allow for more capacity[28]. Also, the stopping patterns of the “faster” express trains (R/R) stem from the riding time requirements rather than the desired stop coverage. However, it can be seen at the stations of *Bad Vöslau*, *Mödling*, and *Wien Liesing*, that the maximum capacity for a conflict-free design of a line with six different riding time patterns has actually been exceeded, since the S and the R systems need to be overtaken *twice* on their journey from *Wiener Neustadt* to *Wien*.

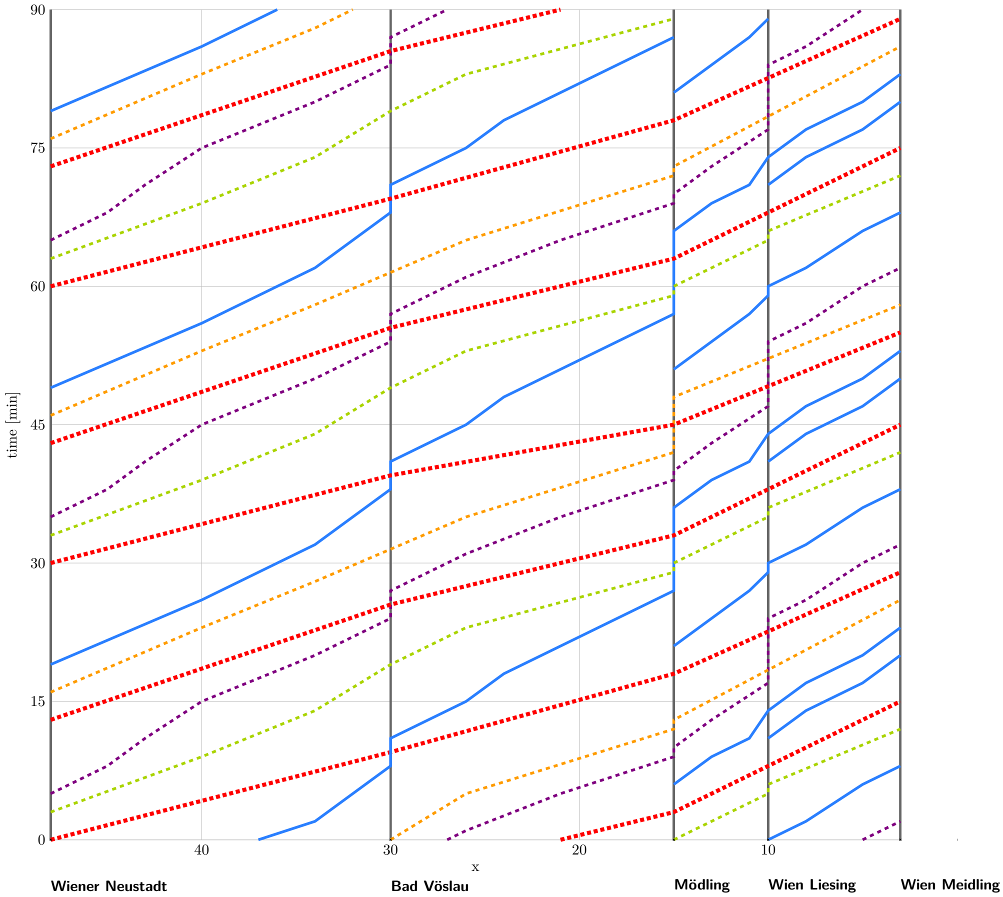


Figure 5.26: Train graph of the *Südbahn* line between Wiener Neustadt and Wien, morning peak, 2016 timetable.

[28]This can be spotted at the D train leaving *Wiener Neustadt* at 6.30 am (minute .30 in the graph). Since it features the regular riding time, it overtakes the R system at Mödling, resulting in a six-minute stop for passengers of this system and a slot switch between *Mödling* and *Wien*. All other REX trajectories are a little slower, thus allowing for a shorter total travel time.

5.2.3 Zoning Trains

The example of the *Südbahn* railway shows a high level of interdependency between different train systems and the immediate result on the track capacity. For centralised structures, *Zoning Trains* can offer a relief: The idea, presented by Borza et al., is to construct a set of train systems of increasing length, such that only the shortest train run serves all stops, while all other train systems pass without stopping. Whenever a train system reaches its terminus, the next shortest train system takes over to serve all stops. *Zoning* refers to the idea that a line is split into *zones* that are each served by a different train system. Figure 5.27 shows such a principle: the first train serves all stops between A (the station with highest grade of centrality) and B and ends there. The second train skips all stops between A and B, makes a transfer with the first train at B and continues to C, calling at all stations. So do the third and the fourth train, such that in the end regional passengers (i. e. passengers on intermediate stops) on the stretch between D and E can reach A directly at almost half the time it would take with a regional train calling at all intermediate stations.

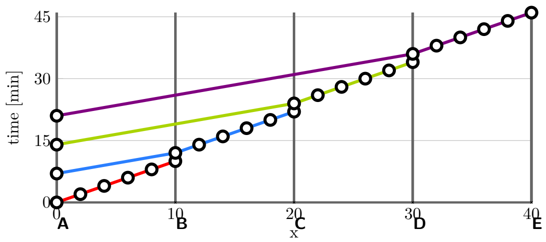


Figure 5.27: Schematic stopping pattern principle of zoning trains

The advantages are that no overtaking is required and all passengers bound for the terminus of high centrality (on the left in the figure) find short riding times *without* the need for a transfers. The drawback is that medium-distance trips along the line need to be covered with transfers. Therefore, the timetable coordination needs to be carried out from the longest train system back to the shortest, arranging the trajectories in such a way that a transfer from the slower to the faster train system is possible. The departures and arrivals at the central terminus follow from the transfers only. A zoning train concept has been successfully applied to many suburban, regional, and even long-distance lines in Hungary, but it is also the key element in the design of the future second inner-city trunk line in *München* (Borza et al. 2005: 57f. Scheller et al. 2015: 17).

Budapest–Szob Zoning Trains

Figure 5.28 shows the timetable graph on the *Budapest–Vác–Szob* line in Hungary, where this service type was first installed in 2004. As can be seen, there is a regular half-hourly service of regional trains between *Budapest* and *Vác*. Just before the next regional train departs from Budapest, a zoning train departs and reaches *Vác* just after the last regional train, so the passengers can transfer to the zoning train. Then, the zoning train changes its train type to serve all stops. There is a second, slightly slower, zoning train alternating with the aforementioned one that makes for a third kind of zoning: Upon reaching *Szob*, this train makes a connection to the EC train that continues to *Štúrovo* (and *Bratislava–Praha–Berlin*). Finally, there is a branch line *Budapest–Veresegyház–Vác* also incorporated in this system. The branch line trains also connect to the zoning trains at *Vác*, which makes *Vác* almost a full timetable hub at minutes .00 and .30[29].

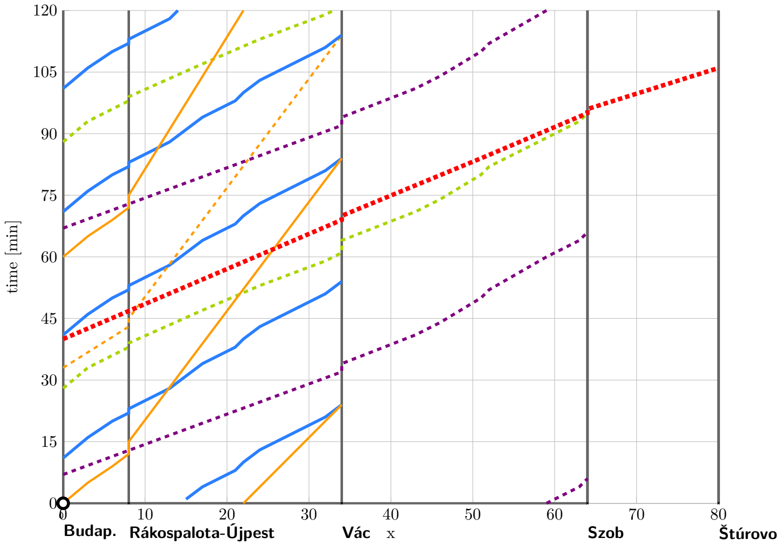


Figure 5.28: Train graph of the line between *Budapest* and *Štúrovo*, afternoon peak, 2015 timetable.

5.3 Trajectory Construction

In the methodology presented here, the main focus of timetable construction is put onto the construction of timetable hubs. Train trajectories are thus considered the result of a hub classification process and therefore need to follow the requirements of hub design, rather than vice versa. However, we do not face infinite possibilities along the edges, but must instead iteratively find a method to both serve the hub requirements and yield valid, i. e. feasible, train trajectories. Therefore, we must deliver a hub structure likely to be feasible at first and then look to adjust the best possible set of trajectories.

[29]Note that the faster trains run through *Vác* shortly after the hub time. However, since all transfers are organised with respect to the trip to and from Budapest, all transfer relations are met regardless.

5.3.1 Hub Classification

For the setup of a hub structure, we need to first carry out a sound hub type classification to serve as *hinges* to attach the train trajectories to.

First, we need to extend the hub types to reflect different intervals. Second, we need to categorise hub types. Figure 5.29 shows the basic types. Note that all of these types are based upon the hub type descriptions carried out in section 5.1.

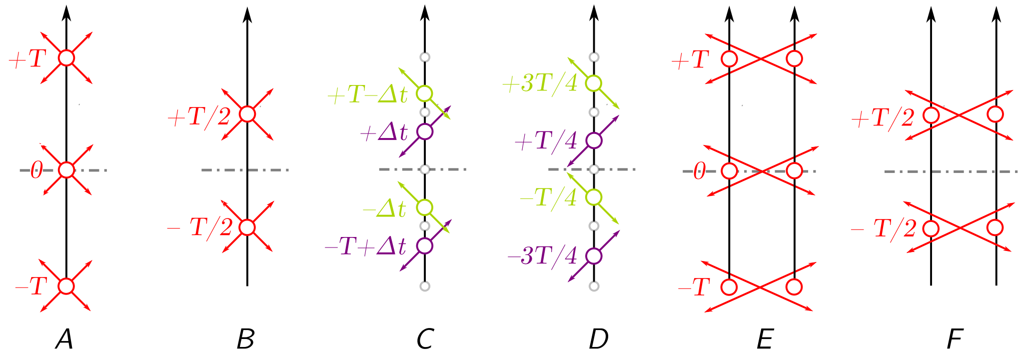


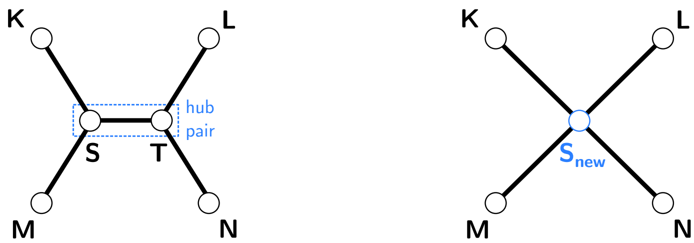
Figure 5.29: Basic hub classification

- *A* and *B* represent classic full timetable hubs (see section 5.1.1) that occur every $^T/_2$. *A* and *B* are two instances of the same type. We differentiate these instances according to whether or not there is a hub right at .00, i. e. at the axis of symmetry[30].
- *C* represents a directional transfer (see section 5.1.2), where Δt denotes the shift towards .00. We do not need to make a distinction of instances as with *A* and *B* here, since the information about transfer compatibility can be judged in a network context only.
- *D* represents a true semi hub (see section 5.1.2), i. e. a directional transfer with equally distributed trajectories. It is a special case of *C* with $\Delta t = {}^T/_4$, but is treated separately to account for its greater importance.
- *E* represents a hub pair (see section 5.1.4), but of topological equivalence to *A*.
- *F* represents a hub pair equivalent to *B*.

Note that hub pairs, i. e. *E* and *F*, only work for full hubs, since their application to a directional transfer (and subsequently to a semi hub) just changes Δt with respect to the hub location, but does not allow for the sophisticated transfer improvements described in section 5.1.4. Hub pairs shall therefore be treated like single hubs in this

[30] In the Integrated Timetable approach (see section 3.3), no differentiation between these two instances is made, since the network-wide hub compatibility is taken over by the rule of cycles. However, we cannot use this rule in a mixed-traffic network, therefore we must implicitly differentiate between these instances.

section. Nevertheless, neither their existence nor their importance shall be neglected. So we keep types E and F, but treat them like types A and B, respectively, during the hub classification process. Figure 5.30a shows at network where stations S and T are to be modelled as a hub pair. Figure 5.30b shows this modified network. Hub S_{new} is now modelled as one single hub, and needs to be expanded to form a hub pair upon the creation of a feasible timetable. Since types E and F cannot work as directional transfers, respectively, we need to restrict the possible hub types for hub pairs to $\{A, B\}$.



(a) Original network layout (b) Modified network layout

Figure 5.30: Modelling of a hub pair

5.3.2 Hub Type Compatibility

Denser intervals, i. e. $\nu_i > 1$[31], render different hub types compatible. This means that increasing values for ν_i increasingly combine hitherto separate hub types. In other words, the denser the interval, the more valid combinations of different basic hub types are possible [32].

In principle, we consider hub types compatible when at least all hub events of the longer interval are also served by the shorter one. Therefore, when there are more than two train systems with more than two different intervals, these intervals need to be mutually compatible, i. e. every ν_i needs to be a common multiple of all lower values.

Figures 5.31, 5.32a, and 5.32b show hub type compatibilities for *even* values of ν_i. Without more elaborated hub constructions, type A requires type A hubs for denser intervals. For even values of ν_i, type B hubs also require type A hubs for denser intervals, since the first interval bisection already leads to a hub at .00. For type D, a hub event occurs every $T/2$,

[31] While not technically necessary, we require $\nu_i \in \mathbb{N}$ since only $T_i|T$ allows for a timetable that repeats every T, which is one core parameter of an integrated timetable. Note that Caimi et al. use $T_i = T \cdot k/\lambda$ to acccount for denser intervals, i. e. $1/\nu_i = k/\lambda$. Since we neglect intervals not fulfilling $T_i|T$, we can use one single parameter rather than two (Caimi et al. 2011: 8).

[32] Note that, while generally independent of the actually offered intervals, it is practical to set T, i. e. the basic interval, to the largest interval in place in a system Typically, $T = 60\,\text{min}$ or $T = 120\,\text{min}$ will be appropriate for railway applications. Theoretically, it is possible to set T to a smaller interval. In this case, full hubs for larger intervals cannot be constructed directly. This setup is only useful if the set of train systems at $T_i > T$ is small and of limited importance. In this case, only every T/T_ith instance of a hub will be served by these train systems.

but shifted towards 0 by $^{T}/_{4}$. Therefore, even values of ν_i first lead to hub type B (since the events are shifted by $^{T_1}/_{4}$, which becomes $^{T_2}/_{2}$ when $T_1 = {}^{T_2}/_{2}$). The amount of hub events per hour remains the same, while all hitherto semi hub events turn into full hub events.

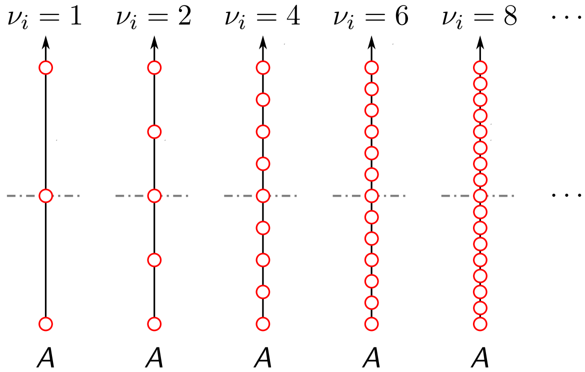
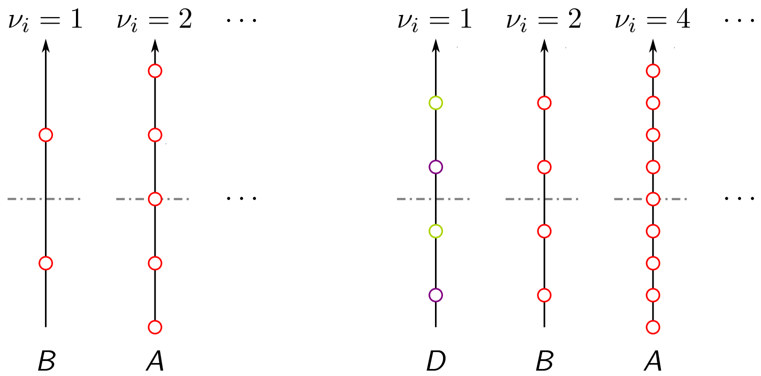


Figure 5.31: Hub type compatibilities for hub type A, even values of $\nu_i > 1$.



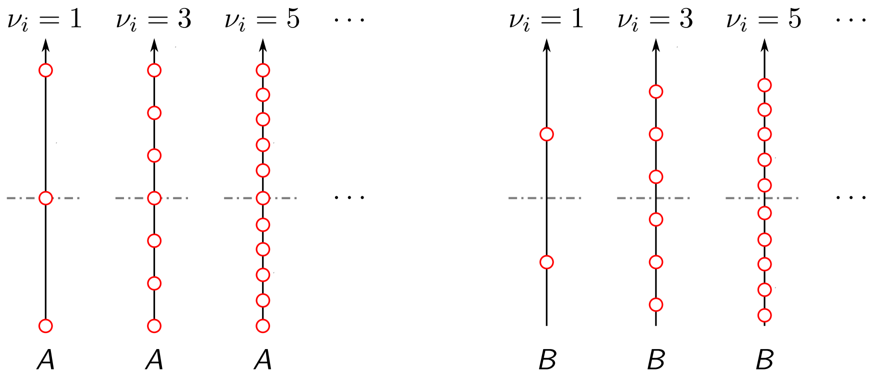
(a) type B

(b) type D

Figure 5.32: Hub type compatibilities for hub type B and D, even values of $\nu_i > 1$.

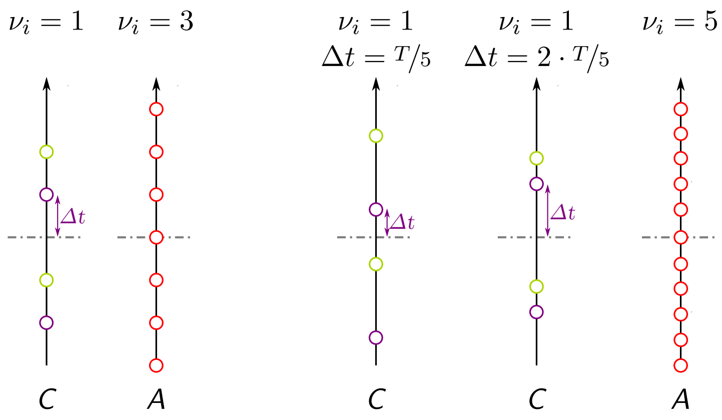
Odd values of ν_i lead to more complicated hub type compatibilities. Since no interval bisection takes place, but a subdivision into an odd number of sections, the differentiation of whether a hub at .00 occurs or not remains in place. Figures 5.33a and 5.33b show the compatibilities for odd values of ν_i.

Type D, i. e. a special case of type C with $\Delta t = {}^{T}/_{4}$, is not compatible with odd values of ν_i, since a semi hub is dependent upon a *bi*section of the interval. However, if we require $\Delta t = {}^{T}/_{\nu_i}$ (and subsequently $\Delta t = {}^{n \cdot T}/_{\nu_i}$), we can achieve compatibility between hub type C and odd values of ν_i. Figures 5.34a and 5.34b show examples for the respective compatibilities. Note that $n \in \mathbb{N}$ and $n < \nu_i$, since $n = \nu_i$ would mean $\Delta t = T$, which essentially changes hub type C to A again.



(a) type A (b) type B

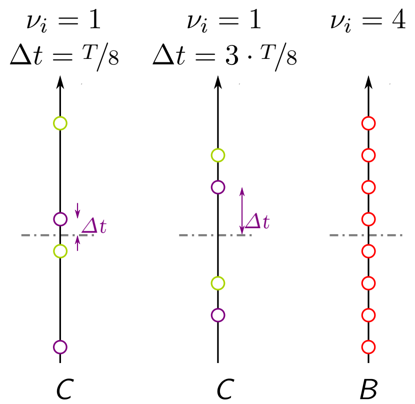
Figure 5.33: Hub type compatibilities for hub type A and B, odd values of $\nu_i > 1$.



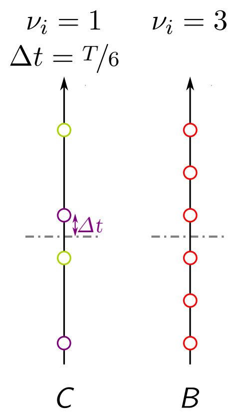
(a) $\nu_i = 3$, $\Delta t = T/3$ (b) $\nu_i = 5$, $\Delta t = n \cdot T/5$

Figure 5.34: Hub type compatibilities for hub type C, odd values of $\nu_i > 1$.

Additionally, type C can serve as a hub type for situations where $T_{i+1} = {}^{T_i}/_n, n > 2$, so e. g. when an hourly train system is to be connected with a train system of a 15-minute interval. In this case, there is no need to consider a compatibility with half-hourly intervals, so a more flexible set of values for Δt is possible. This requires the directional transfer to be shifted by ${}^{\nu_i}/_2$, such that $\Delta t = {}^{(2n+1)\cdot T}/_{2\cdot\nu_{i+1}}, n \in \mathbb{N}^0$ (see figures 5.35a and 5.35b). This way, there is no hub event at 0 and subsequent hubs need to be of type B.



(a) $\nu_i = \{1, 4\}$, $\Delta t = {}^{(2n+1)\cdot T}/_8$



(b) $\nu_i = \{1, 3\}$, $\Delta t = {}^{1\cdot T}/_6$

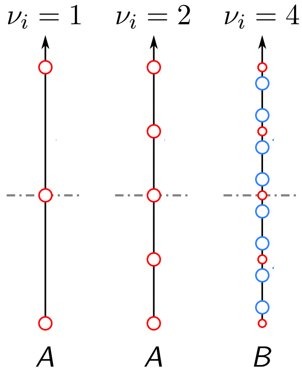
Figure 5.35: Hub type compatibilities for hub type C with $\Delta t = {}^{(2n+1)\cdot T}/_{2\cdot\nu_i}$

Finally, we can generalise the hub type compatibilities by introducing a maximum tolerable transfer time $t_{\text{tr,max}}$. We can make use of this parameter in two ways: (i) train systems with $T_i \le t_{\text{tr, max}}$ can be neglected in the timetable construction process, since we consider transfer improvements at this stage negligible and (ii) train systems with ${}^T/_2 \le t_{\text{tr,max}}$ can serve hubs as *selectively served hubs* as described in chapter 5.1.3. This hub type is but a shift from A to B and vice versa, so we need no additional definition here. This relaxation offers twice the possibilities for hub service for $\nu_i \ge {}^T/_{t_{\text{tr,max}}}$.

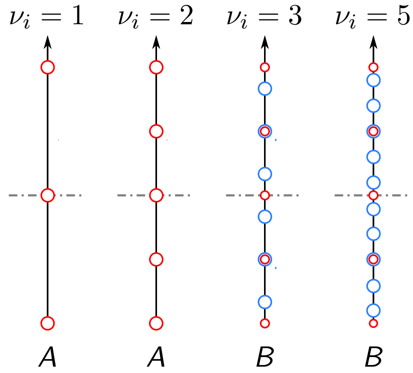
Figure 5.36a shows the compatibilities with $t_{\text{tr,max}} = {}^T/_3$ for even values of ν_i and figure 5.36b for odd values of ν_i. Remarkably, for odd values, the relaxation allows for greater compatibility between different intervals: Hubs with odd ν_i serve either $\{0, T, \ldots\}$ (type A) or $\{{}^T/_2, {}^{3T}/_2, \ldots\}$ (type B). The respective other hub events are missed by ${}^{T_i}/_2$. When ${}^{T_i}/_2 \le t_{\text{tr,max}}$, this missing of a hub turns into a selective service.

With this information, we are already able to generally describe any valid hub type combinations and use this information for a more flexible hub construction.

If we comprehensively cascade these general compatibilities, we receive a high number of possible cumulative hub type combinations. For running time reduction of the algorithm, it is desirable to keep the number of possibilities as low as possible. If we take a closer look at the general description of hub type combinations and compare them to occurrences in reality, the number can be drastically reduced:



(a) Even values of ν_i



(b) Odd values of ν_i

Figure 5.36: Selectively served hubs, $t_{\text{tr,max}} = {}^{T}/_{3}$

1. The number of actual intervals in place is, in reality, limited. The vast majority of public transport networks can be sufficiently modelled with $T_i = \{120, 60, 30, 20, 15, 10, 7.5, 5\}$ min (Sparmann 2006: 12, Walter 2010: 42, Tzieropoulos et al. 2008: 20).
2. Following the conclusion of chapter 3.3.3, we can set $t_{\text{tr,max}} = 10$ min for regional networks, so $T_i = \{120, 60, 30, 20, 15\}$ min.
3. In urban networks, $t_{\text{tr,max}} = 5$ min is more appropriate. However, in these cases it is usually not necessary to simultaneously consider intervals larger than 30 min (Walter 2010: 43).
4. For interval compatibility, we can require $T_i = 20$ min and $T_i = 15$ min *not* to appear within the same hub.
5. For $T = 120$ min, no odd values of ν_i are in existence, since intervals of $T = \{40, 24, \ldots\}$ min are out of scope here.
6. For $T = 120$ min, type C can only occur when $\nu_i \geq 4$, i. e. when *no* $T_i = 60$ min exists (see also figure 5.35a).
7. For $T = 60$ min, the only odd value is $\nu_i = 3$.
8. Types C and D can appear on the highest interval level only.
9. Type C (for $T = 60$ min) exists with $\Delta t = \{20, 10, 22.5, 7.5\}$ min only. The former two require $\nu_i = 3$ and the latter two require $\nu_i = \{2, 4\}$.

   For $T < 60$ min, the corresponding first values of the sequence are cut off.

   All other values of Δt can appear only (i) when one single interval is in place at the hub or (ii) for the special case of interval parting as described in sections 5.2.1 and 5.4.3.
10. Type D is not compatible with any hub with odd values of ν_i.

This leads to the taxative compatibility trees in figures 5.37, 5.38, and 5.39.

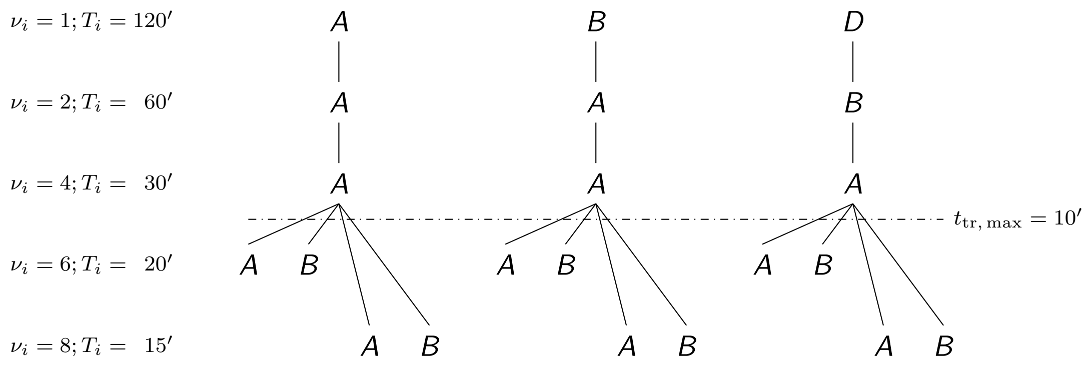


Figure 5.37: Tree of hub type combinations, $T = 120\,\mathrm{min}$, $t_{\mathrm{tr,\,max}} = 10\,\mathrm{min}$

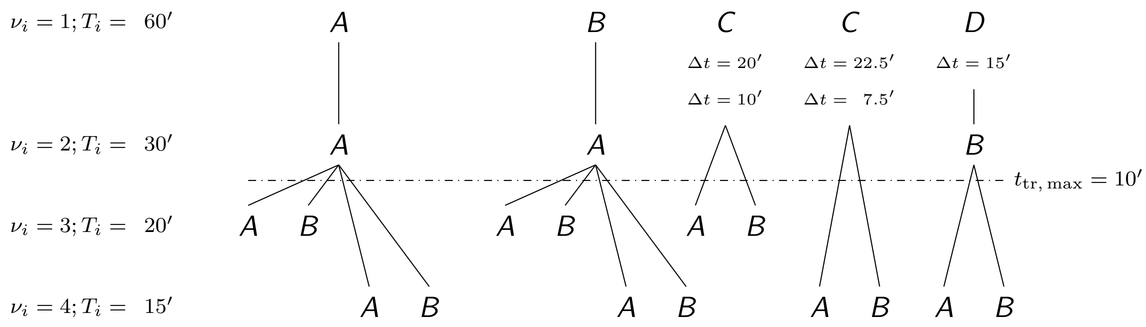


Figure 5.38: Tree of hub type combinations, $T = 60\,\mathrm{min}$, $t_{\mathrm{tr,\,max}} = 10\,\mathrm{min}$

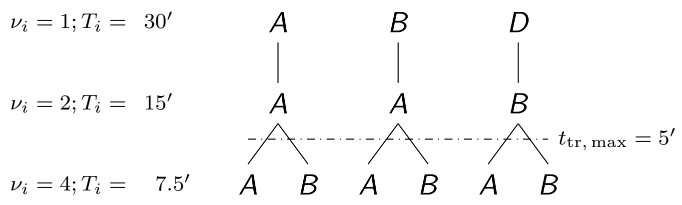


Figure 5.39: Tree of hub type combinations, $T = 30\,\mathrm{min}$, $t_{\mathrm{tr,\,max}} = 5\,\mathrm{min}$

5.3.3 Hub Type Compatibility and Travel Chains

It must be noted at this point that the drawback of an extensively used hub type compatibility is that, in more complex layouts, travel chains along a series of hubs might be broken. Figure 5.40 shows such a setup: While the edges from *Hub 1* to *Hub 2* and from *Hub 3* to *Hub 4* respectively, are each served at interval T_1, the central part between *Hub 2* and *Hub 3* is served at interval $T_2 = {}^{T_1}/_{4}$. As indicated in the figure, one possible solution might render a situation where a transfer is established at both *Hub 2* and *Hub 3*, but there is no direct trajectory between these transfers, since either is served by a different instance of the denser train system. In the worst case shown here and highlighted in grey shading, a potential passenger faces a waiting time of $T_1 - T_2 = {}^{3 \cdot T_1}/_{4}$ at either *Hub 2* or *Hub 3* when going all the way from *Hub 1* to *Hub 4*[33].

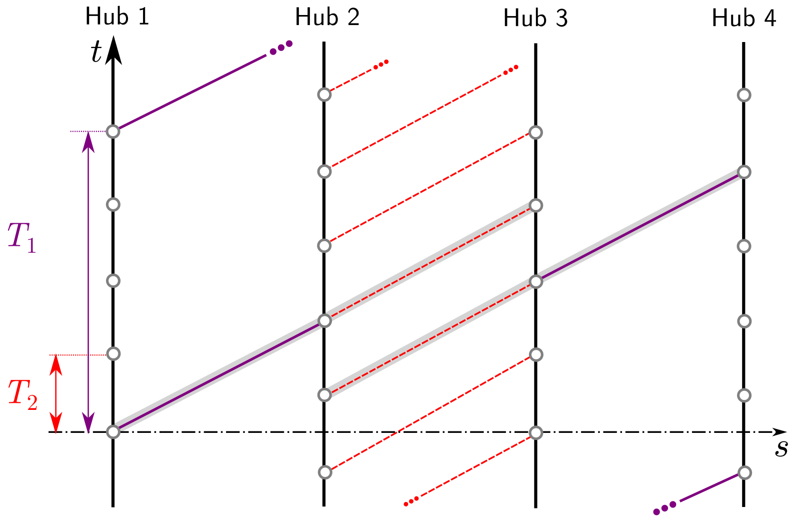


Figure 5.40: Broken travel chain with varying values of T_i

To avoid such a situation, important, longer travel chains, as identified in the Service Intention Phase, can be modelled as continuous lines and the denser stretch can be modelled with *interval parting* as mentioned in section 5.4.3. This way, the transfer conditions within the travel chain can be considered.

Note that an implicit consideration of *all* travel chains is counterproductive in this context: If all possible travel chains across a series of edges were to be fulfilled, this essentially equals a network with *one* interval only, while certain stretches might contain additional journeys without transfer relations. Given this fact, the advantages of more flexible hub types, as described in section 5.1, cannot be applied, neither can mixed traffic be modelled.

Note also that the drawback sketched here applies *only* when an interval changes from a less dense to a denser interval *and back again to a less dense interval*. In practice, such situations imply that there must be some kind of major traffic attractor in the denser section, which will also imply that the travel chains broken in this example will be of minor importance.

[33]Note that any intermediate travel chains, i. e. *Hub 1*–*Hub 3* and *Hub 2*–*Hub 4* can be accomplished without extra waiting time.

5.3.4 Riding Time Calculation

Having classified the hubs, we can now construct train trajectories to connect hub types. In order to obtain a set of valid hub type combinations, we need to find all hub type combinations per edge and per train type. Theoretically, any hub type combination is possible as long as we allow an infinite range of riding speeds. However, we can drastically narrow down the solution space to feasible riding speeds, such that $v_{\min} \leq v_r \leq v_{\max}$. This also aligns with the practical need, since either end of the possible range of riding times has drawbacks if we allowed speeds beyond a feasible range.

1. *Too high* speeds, i. e. too low riding times, would essentially result in unattainable infrastructure and vehicle requirements, especially when acceleration and deceleration at stops put a limit to further speed increases.
2. *Too low* speeds, on the other hand, lead to unattractive riding times and thus render all further timetable discussions useless.

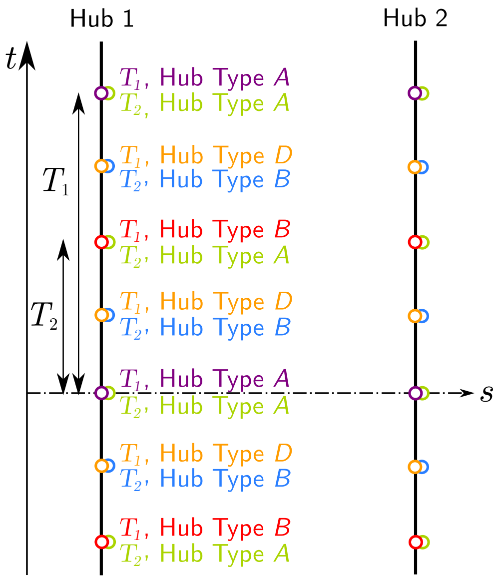
As a starting value, it is a useful approximation to aggregate the current riding times per train system and deduct extreme values from the current timetable. This might require a sensitivity analysis on riding speed feasibility during the hub type conflict resolution process (see section 5.4.5), but shall be considered sufficient for the moment.

Figure 5.41a shows the first step of the riding time calculation: We deduct all existent hub classes from the set of intervals in place and plot them along the time axis of each hub. Then, we pick one of the hub as the starting point and the other as the end point. Since we require the timetable to be symmetrical (see section 2.3.3), it is enough to calculate in one direction only to obtain all valid solutions. Figure 5.41b shows the effects of a symmetric timetable onto such a unidirectional search method: Since any search trajectory is mirrored along the axis of symmetry (depicted by the dash-dotted line), the resulting trajectories (depicted in grey lines) form the same search pattern in the reverse direction again (shown in coloured shading).

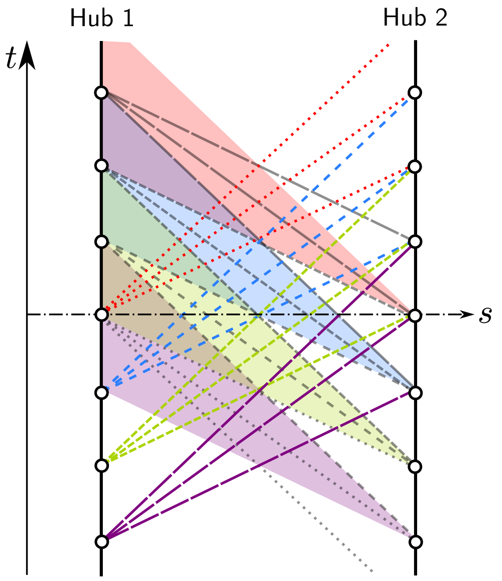
We can define the starting points at *Hub 1* as `hingelist_1`:

$$
\begin{aligned}
\mathsf{A}_0 :&\quad t_{\text{Hub 1}} = && 0 \\
\mathsf{B}_0 :&\quad t_{\text{Hub 1}} = && {}^{T_i}/_2 \\
\mathsf{C}_{\Delta t,0,1} :&\quad t_{\text{Hub 1},\Delta t,1} = && \Delta t \\
\mathsf{C}_{\Delta t,0,2} :&\quad t_{\text{Hub 1},\Delta t,2} = && T_i - \Delta t \\
\mathsf{D}_{0,1} :&\quad t_{\text{Hub 1,1}} = && {}^{T_i}/_4 \\
\mathsf{D}_{0,2} :&\quad t_{\text{Hub 1,2}} = && {}^{3 \cdot T_i}/_4
\end{aligned}
$$

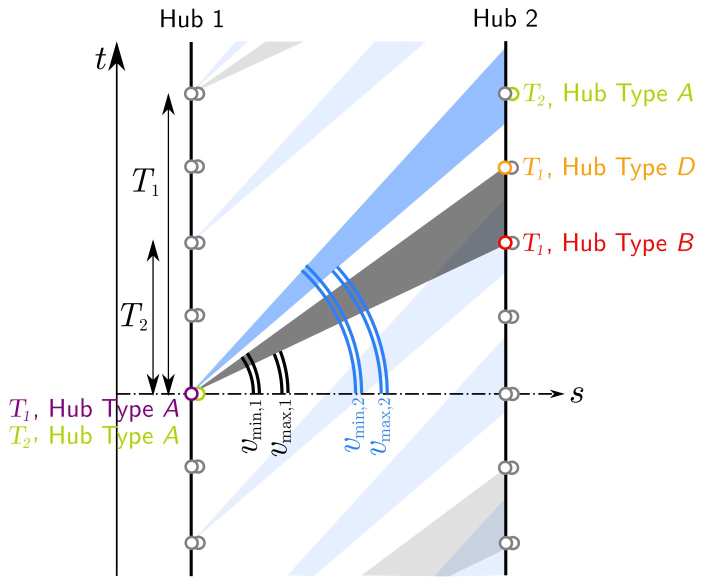
Note that for C_0, there are *two* instances of $t_{\text{Hub 1}}$ for each Δt, and for D_0, there are *two* instances of $t_{\text{Hub 1}}$; in both cases, because the directional nature of the transfer hub creates two hinges per interval.



(a) First step: Identification of hub types per interval.



(b) Validity of unidirectional search due to timetable symmetry.



(c) Second step: Selection of plausible target hubs per train system.

Figure 5.41: Riding time calculation

Second, we need to define the hub hinges on the second hub, i. e. the destination of the edge. Since we cannot predict the set of hubs a priori, we need the full set of possibilities there. While in practice, we do not need an upper limit for the search space (since we handle finite edge lengths and thus finite riding times), we need to algorithmically define an upper limit $n_{\max} \cdot T$ in order to reduce running time. From the evaluation of average riding times and trip lengths combined with the list of typical train type intervals (see section 2.1, Brezina et al. 2014, and section 7.1.3), this limit can be a small integer that is a multiple of T, $n_{\max} = 3$ in this example[34]. Since the latest possible starting point at *Hub 1* is $T - \min(\Delta t)$, we can therefore account for riding times of up to $2 \cdot T + \min(\Delta t)$ for the slowest train system, which amounts to more than 120 minutes per edge in networks with hourly base interval.

We can define the end points at *Hub 2* as `hingelist_2`:

$$\begin{aligned}
\mathsf{A}_j: &\quad t_{\text{Hub 2}} = &&\{j \cdot T_i\}\\
\mathsf{B}_j: &\quad t_{\text{Hub 2}} = &&\{{}^{T_i}\!/_2 + j \cdot T_i\}\\
\mathsf{C}_{\Delta t,j,1}: &\quad t_{\text{Hub 2},\Delta t,1} = &&\{T_i - \Delta t + j \cdot T_i\}\\
\mathsf{C}_{\Delta t,j,2}: &\quad t_{\text{Hub 2},\Delta t,2} = &&\{\Delta t + j \cdot T_i\}\\
\mathsf{D}_{j,1}: &\quad t_{\text{Hub 2},1} = &&\{{}^{3 \cdot T_i}\!/_4 + j \cdot T_i\}\\
\mathsf{D}_{j,2}: &\quad t_{\text{Hub 2},2} = &&\{{}^{T_i}\!/_4 + j \cdot T_i\}
\end{aligned}$$

Note that $t_{\text{Hub 2}}$ for A and B, respectively, are vectors with $n_{\max} \cdot \nu_i - 1$ elements. For D, $t_{\text{Hub 2}}$ is a vector with $2 \cdot n_{\max} \cdot \nu_i - 1$ elements and for C, $t_{\text{Hub 2}}$ is a $(2 \cdot n_{\max} \cdot \nu_i - 1) \times |\Delta t|$ matrix. Note that $C_{\Delta t,j,1}$ and $C_{\Delta t,j,2}$ changed order when compared to the starting points. This is due to the further treatment of directional transfers as described in section 5.3.5.

Figure 5.41c finally shows the process of riding time calculation. For each train system with interval divisor ν_i, $v_{\min}$, and $v_{\max}$ on each edge with length l_{edge}, we need to find a hub type combination of $t_{\text{Hub 1}} \in$ `hingelist_1` and $t_{\text{Hub 2}} \in$ `hingelist_2` such that

$$t_{\text{Hub 1}} + \frac{l_{\text{edge}}}{v_{\min}} \leq t_{\text{Hub 2}} \leq t_{\text{Hub 1}} + \frac{l_{\text{edge}}}{v_{\max}} \tag{5.2}$$

In figure 5.41c, we can spot two train systems:

1. train system 1 with T_1, $v_{\min,1}$, and $v_{\max,1}$, and
2. train system 2 in blue with T_2, $v_{\min,2}$, and $v_{\max,2}$.

In the figure, we can see both train systems start at hub type A at $t = 0$ and span a funnel-shaped search area according to their respective $v_{\min}$ and $v_{\max}$. In the example, train system 1 reaches two hub hinges (type D and B), and train system 2 reaches one

[34] When $T = 60\,\text{min}$ and $n_{\max} = 3$, all solutions in the interval $[0, 180]\,\text{min}$ are considered.

hinge of type *A*. To indicate the influence of the respective intervals, further funnels are depicted in faded colours. As can be seen already, train system 2 would be able to reach the same hub type *B* as train system 1 when we consider one departure earlier. We will make use of this effect by means of *Hub Type Transformation* as described in section 5.3.6.

Of course, there might always be combinations of edge lengths, riding speeds, and intervals where *no* solution can be found. This can be tackled by

1. modifying (usually shortening) the interval,
2. combining one train system with another for mutual interval parting, as described in section 5.4.6,
3. modifying v_{min} and/or v_{max}.

and needs to be covered within the edge preprocessing (see section 7.1) However, for the time being, trains where no valid trajectory was found are moved to the list `nonefound`.

Note that during hub type conflict resolution (see section 5.4.6) at the end of the Target Timetable Phase and during the Feasible Timetable Phase, a much larger set of solutions to infeasible train trajectories can be applied. This is, however, only possible if (i) the network context and (ii) the influences of other train systems are known. Therefore, we must stick to this limited set of variations at this point in time.

The step of riding time calculation needs to be repeated on all edges, resulting in a `hublist` as shown in table 5.1. Note that "Profile No." denotes the running number to identify several possible riding time profiles per train system.

Edge No.	Hub Name 1	Hub Name 2	Train System	ν_i	Hub Type 1	$t_{\text{Hub 1}}$	Hub Type 2	t_{Hub2}	t_r	v_r	Profile No.
1	Hub 1	Hub 2	S11	4	A	00	A	60	60	60	1
1	Hub 1	Hub 2	IC11	1	A	00	D	45	45	80	1
1	Hub 1	Hub 2	IC11	1	D	45	B	90	45	80	2
2	Hub 2	Hub 3	R21	2	B	15	A	60	45	80	1
3	Hub 3	Hub 4	IC31	1	A	00	D	45	45	80	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 5.1: Sample comprehensive `hublist` table for a 60 km stretch, $T = 60$ min.

For a better understanding, we take a closer look at train system IC11 in table 5.1. As can be seen, the first line finds hub type *A* at minute .00 at *Hub 1* and hub type *D* at minute .45 at *Hub 2*. This is calculated as follows: Considering $v_{\text{min}} = 70$ km/h and

$v_{\max} = 90$ km/h for train type *IC*, at 60 km stretch can be negotiated in [40, 52] min. Since $\nu_i = 1$, i. e. IC11 runs hourly, the only hinge when departing from *Hub 1* at minute .00 (type A) is at minute .45 in *Hub 2* (type D). Also, a second instance of train system IC11 is displayed: When starting from *Hub 1* at minute .45 (type D), a hinge at minute .90 (type B) at *Hub 2* can be reached.

5.3.5 Treatment of Directional Transfer Hub Types

While hub types A and B (as well as E and F) can be used straight away for transfer relations, directional transfer hubs, i. e. types C and D, need to be treated further.

Without further treatment, table 5.1 would regard hub types C and D as is, without further information about which instance of the hub is reached. This is a side effect of considering the directional search sufficient, as stated in section 5.3.4. As discussed in section 5.1.2, hub events of directional transfer hubs occur *twice* as often as full hubs, since either occurrence serves one direction of a timetable only. When combined with lines of denser intervals, this is of no importance, since the directional property vanishes when used as a hinge for full hubs (see figure 5.7 in section 5.1.2). However, in cases where two or more lines of the same interval all serve a directional transfer hub, the directional property becomes important.

Figure 5.42 shows this side effect. When searching from *Hub 1* towards *Hub 2*, a search from hub type A at *Hub 1* yields type D at $t_{\text{Hub 3}} = {}^{T}/_{4}$, which implies that transfer (or a continuation of the journey) towards *Hub 2* leads to this hub being of type B. Furthermore, a search from *Hub 2* towards *Hub 3* and *Hub 2* being of type A also yields type D at $t_{\text{Hub 3}} = {}^{T}/_{4}$. However, it can be easily seen that this would *not* result in a valid transfer relation at *Hub 3*. The same applies to hub type C accordingly.

Since we restricted hub types C and D to only occur at the topmost interval present in a hub (see section 5.3.1), the treatment of this peculiarity is comparatively simple, yet needs to be taken into account. For each instance of hub types C and D, we add a *hub type index* stating whether the instance closer or the farther from the axis of symmetry has been found.

However, this index needs to be different whether the *origin* or the *destination* hub of a search is categorised. As can be seen in figure 5.42, the *end*point at $t_{\text{Hub3}} = {}^{T}/_{4}$, when searched from *Hub 1*, is equivalent to the *starting* point at $t_{\text{Hub 3}} = {}^{T}/_{4}$. When searched from *Hub 2*, it is equivalent to the *end*point at $t_{\text{Hub 3}} = {}^{3 \cdot T}/_{4}$, and the *starting* point at $t_{\text{Hub 3}} = {}^{3 \cdot T}/_{4}$.

Therefore, it is not sufficient to just index the points found by their actual respective distance to the axis of symmetry, but depending upon their topological position within the search process. Table 5.2a shows the matrix of indexing conventions depending on the mentioned topological position.

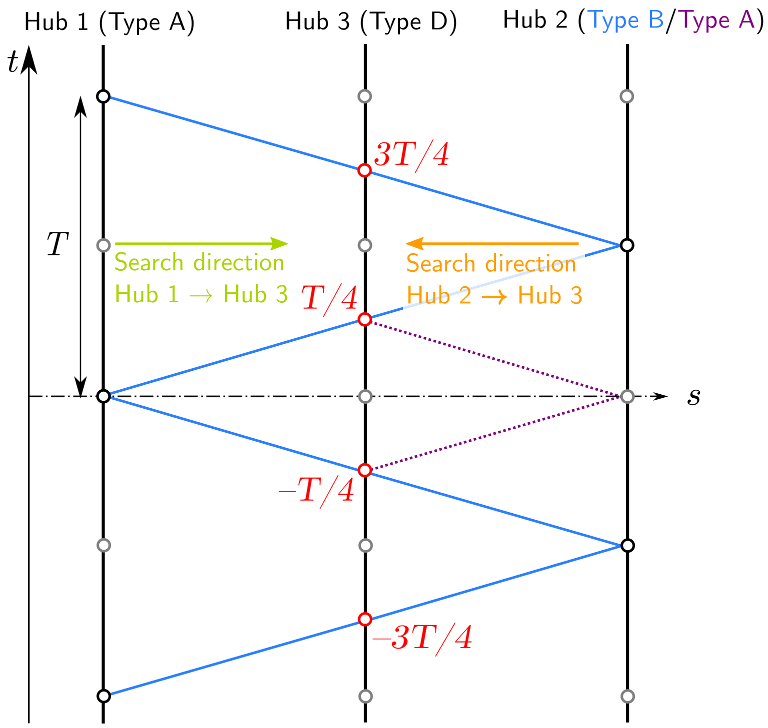


Figure 5.42: Side effect of unidirectional trajectory search in directional transfer hubs

	t_{origin}	$t_{\text{destination}}$
Δt	$C_{\Delta t,2}$	$C_{\Delta t,1}$
$T - \Delta t$	$C_{\Delta t,1}$	$C_{\Delta t,2}$

(a) Type C

	t_{origin}	$t_{\text{destination}}$
$T/4$	D_2	D_1
$3{\cdot}T/4$	D_1	D_2

(b) Type D

Table 5.2: Indexing of directional transfer hubs with respect to their topological position within the search process

At first sight, this does not seem sufficient to guarantee a valid transfer (or continuous journey, respectively). But here, we can take advantage of the unidirectional search: Any transfer that will work geometrically in the three-dimensional view can be considered to work out in practice, as long as we keep track of the direction in which the search has been carried out.

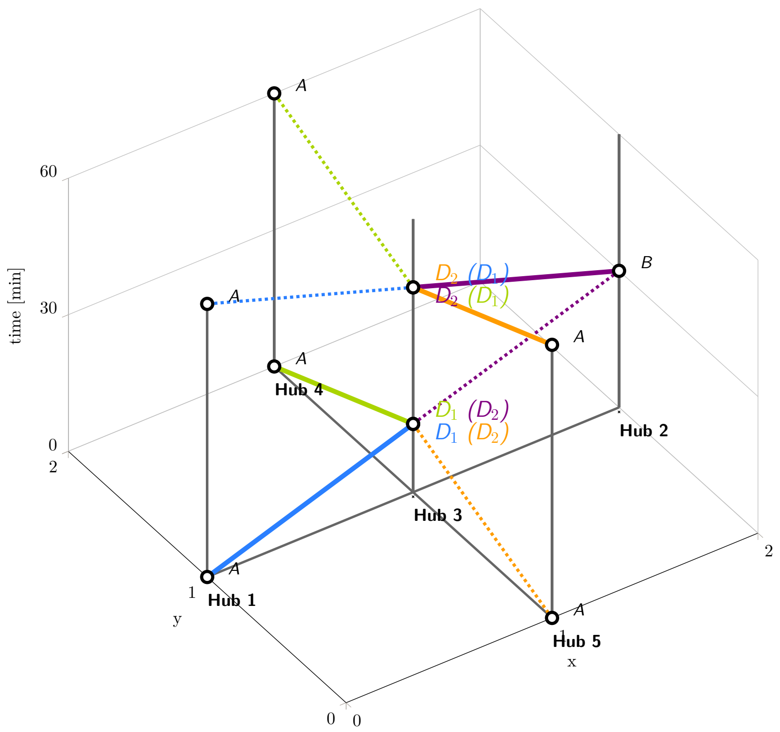


Figure 5.43: Hub type indexing and transfer conditions

Figure 5.43 shows such a situation in the three-dimensional view: The trajectories from *Hub 1* and *Hub 2*, respectively, to *Hub 3* represent those in figure 5.42, with the search direction highlighted in full, thicker lines and the corresponding symmetrical counterparts drawn in dotted lines. *Hub 3* can, when seen from *Hub 1*, be described as type D_1 (marked in the respective colours of the trajectories), since a search from type A in Hub 1 yielded $t_{\text{Hub 3}} = {}^{T}/_{4}$. The same hub, when seen from Hub 2, is of type D_2, since a search from type B in Hub 2 yielded $t_{\text{Hub 3}} = {}^{3 \cdot T}/_{4}$. As can be seen,

1. a transfer or continuous journey is possible when two *end*points with *different* hub type indices meet.

On the other hand, again in figure 5.43, there are trajectories from *Hub 4* to *Hub 3* and from there to *Hub 5*. In the former case, *Hub 3* is of type D_1, since a search from *Hub 4* of type A yielded $t_{\text{Hub 3}} = {}^{T}/_{4}$. In the latter case, *Hub 3* is of type D_2, since the *origin* is at $t_{\text{Hub 3}} = {}^{3 \cdot T}/_{4}$. As opposed to before, in this case the two trajectories head in the same direction, i. e. the indexing of the intermediate hub directly reflects the geometrical impossibility of a transfer or a continuous journey. Therefore,

2. a transfer or continuous journey is possible when one *starting* and one *end*point of *identical* hub type index meet.

Finally, as can be seen by the symmetrical counterparts of the respective trajectories, these two rules also apply for the third possible combination,

3. a transfer or continuous journey is possible when two *starting* points of *different* hub type index meet.

We will, however, *not* make use of directional transfer hub type indexing for the overall hub type compatibility discussion to follow: In figure 5.43, we can see that both indices will always occur at a hub jointly, since any trajectory is cut by the vertical hub axis *twice* per interval (see also figure 5.4 in section 5.1.2). So from a global perspective, there is no need to further subdivide the hub types; we must, however, keep track of the indexing throughout the algorithm for later use in hub type conflict diagnosis.

5.3.6 Hub Type Conversion

We can make use of the hub type compatibilities from section 5.3.1, since we need a manifold of possibilities to construct timetable hubs compatible with several different intervals.

Essentially, the main benefit of this hub classification is that train types with denser intervals can make for more flexibility in the hub type definition of train systems with less dense intervals.

Hub type A with $\nu_i = 4$ can therefore be converted to type A or B with $\nu_i = 2$. The former can be further converted to type A or B and the latter to type D, either with $\nu_i = 1$.

We therefore need to reformulate the compatibility trees to be read upside down, so as to gain knowledge about the conversions with *de*creasing values of ν_i. Figures 5.44 and 5.45 show these relationships.

Then, we take the `hublist` table and extract the information about ν_i, the involved hubs and their respective hub types, resulting in a compacted table such as shown in table 5.3.

For any $\nu_i > 1$, we need to translate the hub type to match the type at the basic interval T. This is essentially an incremental copy of each line, while decreasing the corresponding ν_i in each step, until the final table consists of $\nu_i = 1$ only. For the sample `hublist` in table 5.3, we need to first convert the lines with $\nu_i = 4$ to multiple entries with $\nu_i = 2$, yielding a modified list with $\nu_i = \{1, 2\}$ only, and finally convert these entries, to finally obtain a list with $\nu_i = 1$ only.

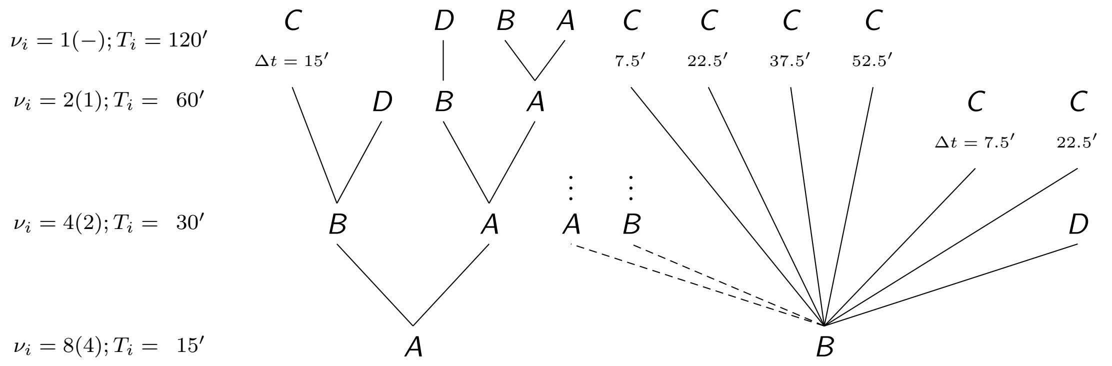


Figure 5.44: Inverse tree for hub type transformation, $T = 120\,\text{min}$, $t_{\text{tr, max}} = 10\,\text{min}$ (values for $T = 60\,\text{min}$ in brackets), starting at $\nu_i = 8(4)$. The dashed branches can be used since $T_i/2 < t_{\text{tr, max}}$ in this case. Values of $\nu_i \leq 4$ are included within the comprehensive trees for $\nu_i = 8$.

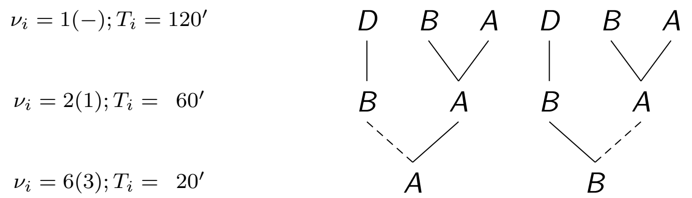


Figure 5.45: Inverse tree for hub type transformation, $T = 120\,\text{min}$, $t_{\text{tr, max}} = 10\,\text{min}$ (values for $T = 60\,\text{min}$ in brackets), $\nu_i = 6(3)$. The dashed branches can be used since $T_i/2 < t_{\text{tr, max}}$ in this case.

ν_i	Train System	Hub Name 1	Hub Type 1	$t_{\text{Hub 1}}$	Hub Name 2	Hub Type 2	$t_{\text{Hub 1}}$	Profile No.
4	S11	Hub 1	A	00	Hub 2	A	60	1
1	IC11	Hub 1	A	00	Hub 2	D_2	45	1
1	IC11	Hub 1	D_1	45	Hub 2	B	90	2
2	R21	Hub 2	B	15	Hub 3	A	60	1
1	IC31	Hub 3	A	00	Hub 4	D_2	45	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 5.3: Sample compacted `hublist` table.

Note that, upon conversion, there is no need to maintain the subdivision of directional transfer hubs. Since the train systems in the conversion process feature intervals of $T_i \leq {}^{T}\!/2$ by definition, all instances of directional hubs turn into full hubs[35].

When converting the list, we need to keep track of the riding times obtained within the riding time calculation step. If we simply followed the compatibility trees for the hub types on either side of an edge, we would obtain a significantly greater number of possible solutions, which in turn would imply infeasible hub type combinations.

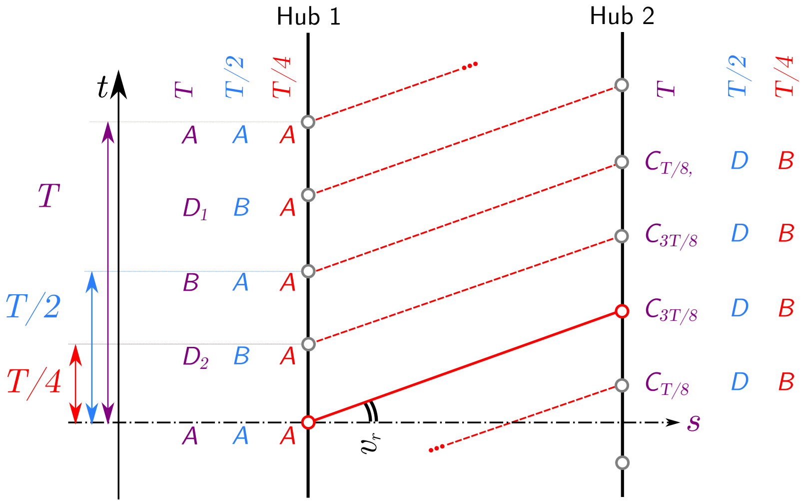


Figure 5.46: Feasible hub type conversions

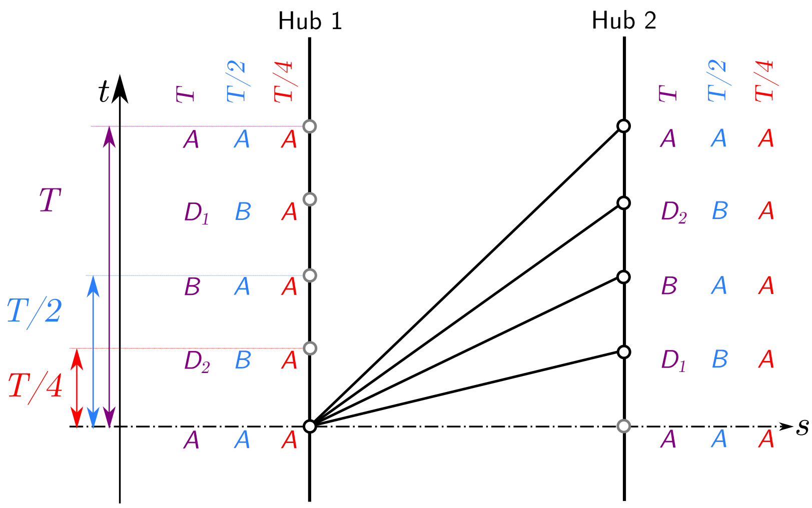


Figure 5.47: Differences in hub type conversion for identical base trajectories

[35]Note that, by the hub type compatibilities elaborated for type C, an interval of $T_i \leq {}^{T}\!/3$ is required, such that these hubs also turn into full hubs.

Figure 5.46 shows a sample hub type combination for $T_i = {}^{T}\!/_{4}$. Upon creation of valid trajectories, the base trajectory (thick solid line), shown here as A–B, is the same for all its copies within $0 \leq t_{\text{Hub 1}} \leq T$ (thin dashed trajectories). Still at Hub 1, we can see the corresponding hub types at each instance of the basic trajectory in separate columns for the respective higher values of T_i, in this case ${}^{T}\!/_{2}$ and T. Focussing only on Hub 1, we can simply follow the aforementioned conversion trees. At Hub 2, however, we can see that each conversion at Hub 1 corresponds to one conversion at Hub 2 only.

What is more, even a combination of two identical hub types might yield completely different results in the conversion. Figure 5.47 shows possible different conversions for identical base trajectories. While all the cases shown are of type A–A in the original interval, they yield different conversions for larger intervals.

Since we kept the information of the individual hub times and the riding speeds in the `hublist`, we can use this information for the hub type conversion. We first permute the hub type for the first hub and reconstruct the second hub type by adding the obtained riding time and matching it with the corresponding hub types for the larger interval.

We obtain the individual, successive hub times $t_{\text{Hub 2},j+1}$ first by altering the hub times of the first hub in the first instance $t_{\text{Hub 1},j}$ and then adding the corresponding number of intervals $j \cdot T_i$ and the individual riding time t_r:

$$t_{\text{Hub 1},j+1} = t_{\text{Hub 1},j} + j \cdot T_i \tag{5.3}$$

$$t_{\text{Hub 2},j+1} = t_{\text{Hub 1},j+1} + t_r \tag{5.4}$$

$$j = \{0, {}^{T_{i+1}}\!/_{T_i}\} \in \mathbb{N}^0 \tag{5.5}$$

For trains where *selective hub service* applies (i. e. $T_i \leq t_{\text{tr,max}}$, see section 5.1.3), we additionally must include first hub departure times shifted by half the interval, so that the selectively served hubs are reached as well:

$$t_{\text{Hub 1},T_i \leq t_{\text{tr,max}},j+1} = t_{\text{Hub 1},j} + {}^{T_i}\!/_{2} + j \cdot T_i \tag{5.6}$$

$$t_{\text{Hub 2},T_i \leq t_{\text{tr,max}},j+1} = t_{\text{Hub 1},j+1,T_i \leq t_{\text{tr,max}}} + t_r \tag{5.7}$$

Then, we can make use of the vectors A, B, $C_{\Delta t}$, and D that we needed for riding time calculation and match them with $t_{\text{Hub 1},j+1}$ and $t_{\text{Hub 2},j+1}$ as mentioned:

$$\texttt{searchspace} = A \cup B \bigcup_{\Delta t} C_{\Delta t} \cup D \tag{5.8}$$

$$t_{\text{Hub 1},j} \overset{?}{\in} \texttt{searchspace} \overset{?}{\ni} t_{\text{Hub 2},j} \tag{5.9}$$

$$j \in \mathbb{N}^0 \tag{5.10}$$

Tables 5.4 and 5.5 show the conversions of the `hublist` from table 5.3 as discussed. Furthermore, table 5.4 shows, highlighted in red, the additional transfers for the case that $T_i \leq t_{\text{tr,max}}$. These have been omitted in table 5.5 for readability. The final result is a list of valid hub type combinations for one value of T_i (or ν_i, respectively) only.

ν_i	Train System	Hub Name 1	Hub Type 1	$t_{\text{Hub 1}}$	Hub Name 2	Hub Type 2	$t_{\text{Hub 1}}$	Profile No.
~~4~~	~~S11~~	~~Hub 1~~	~~A~~	~~00~~	~~Hub 2~~	~~A~~	~~60~~	~~1~~
2	S11	Hub 1	A	00	Hub 2	A	60	1
2	S11	Hub 1	B	15	Hub 2	B	75	1
2	S11	Hub 1	D	07.5	Hub 2	D	67.5	1
2	S11	Hub 1	D	22.5	Hub 2	D	82.5	1
1	IC11	Hub 1	A	00	Hub 2	D_2	45	1
1	IC11	Hub 1	D_1	45	Hub 2	B	90	2
2	R21	Hub 2	B	15	Hub 3	A	60	1
1	IC31	Hub 3	A	00	Hub 4	D_2	45	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 5.4: Conversion of `hublist` entries, $\nu_i = 4$ to $\nu_i = 2$.

5.3.7 Network-Wide Hub Type Application

With the final converted `hublist`, we end up with a network-wide list of valid hub type combinations. In other words, we have found all trajectories to match hub types along the edges of the network. Figure 5.48 shows the corresponding network to the `hublist` discussed before with such valid trajectories. The train systems in the example are marked in their respective colours.

5.4 Trajectory Matching

Once we are able to pick combinations in such a way that the same hub type at one hub can be met from either adjacent edge, we can find valid network target timetables. In order to do so, we need to match all trajectories according to the hub types served by them.

ν_i	Train System	Hub Name 1	Hub Type 1	$t_{\text{Hub 1}}$	Hub Name 2	Hub Type 2	$t_{\text{Hub 1}}$	Profile No.
~~2~~	~~S11~~	~~Hub 1~~	~~A~~	~~00~~	~~Hub 2~~	~~A~~	~~60~~	~~1~~
1	S11	Hub 1	A	00	Hub 2	A	60	1
1	S11	Hub 1	B	30	Hub 2	B	90	1
~~2~~	~~S11~~	~~Hub 1~~	~~B~~	~~15~~	~~Hub 2~~	~~B~~	~~75~~	~~1~~
1	S11	Hub 1	D	15	Hub 2	D	75	1
1	S11	Hub 1	D	45	Hub 2	D	105	1
1	IC11	Hub 1	A	00	Hub 2	D_2	45	1
1	IC11	Hub 1	D_1	45	Hub 2	B	90	2
~~2~~	~~R21~~	~~Hub 2~~	~~B~~	~~15~~	~~Hub 3~~	~~A~~	~~60~~	~~1~~
1	R21	Hub 2	D	15	Hub 3	A	60	1
1	R21	Hub 3	D	45	Hub 3	B	90	1
1	IC31	Hub 3	A	00	Hub 4	D_2	45	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Table 5.5: Conversion of `hublist` entries, $\nu_i = 2$ to $\nu_i = 1$.

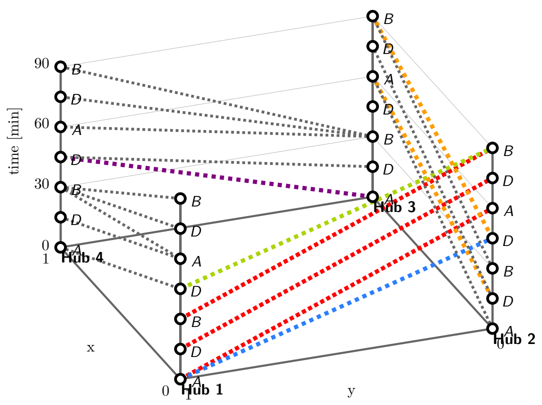


Figure 5.48: Sample network with valid trajectories for hub type matching.

5.4.1 Transformation to Truth Function

From the final converted `hublist`, we can deduct a list of statements to be fulfilled in order to obtain a valid timetable. Taking the list from table 5.4, we can, for train system S11, state that

- Hub 1 is of type A *and* Hub 2 is of type A *or*
- Hub 1 is of type B *and* Hub 2 is of type B *or*
- ...

In other words, we can formulate a *truth function* to be fulfilled. The statement from table 5.4 can be rephrased to read:

$$((\text{Hub 1} = A) \wedge (\text{Hub 2} = A)) \vee ((\text{Hub 1} = B) \wedge (\text{Hub 2} = B)) \vee \ldots \tag{5.11}$$

However, a mere adherence to this notation would jeopardise the idea of increased flexibility through denser intervals. Figure 5.49 shows three train systems, running at $T_i = T$, $T_i = {}^{T}/_{2}$, and $T_i = {}^{T}/_{4}$, respectively. The base trajectories are highlighted in thicker lines, while the incremental copies that stem from the hub type conversion are depicted in thinner lines. If every line in the `hublist` were translated to one clause in the truth function, only exactly those hub type combinations reachable by the trajectories could be used for the truth function. But since we do allow trajectories to be attached to a hub hinge on one hub only and let the other hub be served by an instance shifted by $n \cdot T_i$ (see chapter 5.3.1), we can also allow a permutation within each riding time profile.

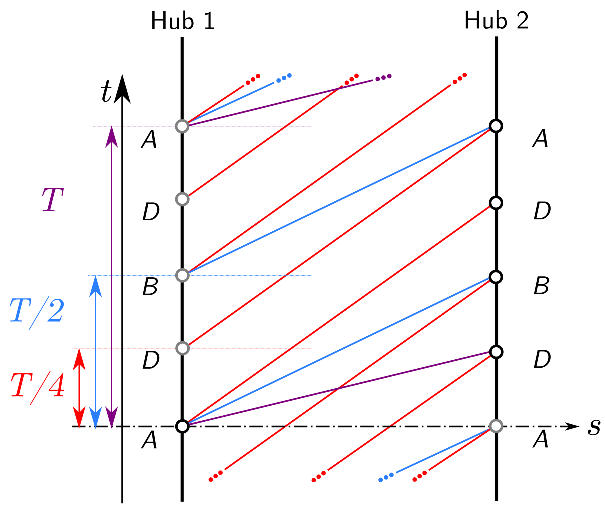


Figure 5.49: Effects of hub type compatibility for truth function

The truth function 5.11 therefore needs to be rephrased for each riding time profile:

$$\underbrace{((\text{Hub } 1 = A) \vee (\text{Hub } 1 = B)) \wedge ((\text{Hub } 2 = A) \vee (\text{Hub } 2 = B))}_{\text{Profile 1}} \vee \\ \underbrace{((\text{Hub } 1 = D) \vee (\text{Hub } 1 = D)) \wedge ((\text{Hub } 2 = D) \vee (\text{Hub } 2 = D))}_{\text{Profile 2}} \wedge \ldots \quad (5.12)$$

Note that the second clause incorporates two tautologies, namely (Hub $1 = D) \vee$(Hub $1 = D$) and (Hub $1 = D) \vee$ (Hub $1 = D$). These stem from the algorithmic handling of the `hublist` tables. There is, however, no need to filter them out, since their processing does not affect the running time.

For one edge, we need to group these clauses per train system, so as to keep the information on which prerequisites must be met in any case (grouped with *and*) and which are alternatives (grouped with *or*). Extending formula 5.12, we can formulate:

$$\underbrace{[(((\text{Hub } 1 = A) \vee (\text{Hub } 1 = B)) \wedge ((\text{Hub } 2 = A) \vee (\text{Hub } 2 = B))) \vee \ldots]}_{\text{S11}} \\ \wedge \quad \underbrace{[((\text{Hub } 1 = A) \wedge (\text{Hub } 2 = D)) \vee ((\text{Hub } 1 = D) \wedge (\text{Hub } 2 = B)) \vee \ldots]}_{\text{IC11}} \\ \wedge \quad \ldots \quad (5.13)$$

For the whole network, we face the same requirement, since we can state that requirements need to be matched for any train system as such, i. e. we can simply append further edges with an *and* clause. This would, in the example from table 5.4, lead to an extension of formula 5.13:

$$\underbrace{[(((\text{Hub } 1 = A) \vee (\text{Hub } 1 = B)) \wedge ((\text{Hub } 2 = A) \vee (\text{Hub } 2 = B))) \vee \ldots]}_{\text{S11, edge 1}} \\ \wedge \quad \underbrace{[((\text{Hub } 1 = A) \wedge (\text{Hub } 2 = D)) \vee ((\text{Hub } 1 = D) \wedge (\text{Hub } 2 = B)) \vee \ldots]}_{\text{IC11, edge 1}} \\ \wedge \quad \underbrace{[(((\text{Hub } 2 = D) \vee (\text{Hub } 2 = D)) \wedge ((\text{Hub } 3 = B) \vee (\text{Hub } 3 = A))) \vee \ldots]}_{\text{R21, edge 2}} \\ \wedge \quad \underbrace{[((\text{Hub } 3 = A) \wedge (\text{Hub } 4 = D))]}_{\text{IC31, edge 3}} \\ \wedge \quad \ldots \quad (5.14)$$

This way, we obtain a formula of constraints to be obeyed in order to achieve a valid timetable.

5.4.2 Boolean Satisfiability Problem

The Boolean Satisfiability Problem (SAT) is a common solving strategy for complex decision problems (Franco et al. 2009). It is based on a mere combination of *literals*, i. e. variables that can be either true or false.

For the aforementioned constraint formula to be used with SAT, we need a dismembering into Boolean literals. Using the constraints directly as literals would fail to work, since the information that one hub is of a certain type does not contain the information that the hubs is *not* of another type.

In order for a constraint to become a set of literals that include the information of which type a hub is *not*, we need to extend one constraint to a set of literals that comprise both the information of which type a hub *is* and *is not*.

Extending just the first clause of formula 5.14, this reads:

$$
\begin{aligned}
\Bigg[\Bigg(\Bigg(&\underbrace{(\text{Hub } 1 = A) \wedge \neg(\text{Hub } 1 = B) \wedge \neg(\text{Hub } 1 = C) \wedge \neg(\text{Hub } 1 = D)}_{\text{Hub } 1 = A}\Bigg) \\
&\vee \Bigg(\underbrace{\neg(\text{Hub } 1 = A) \wedge (\text{Hub } 1 = B) \wedge \neg(\text{Hub } 1 = C) \wedge \neg(\text{Hub } 1 = D)}_{\text{Hub } 1 = B}\Bigg)\Bigg) \\
&\wedge \Bigg(\Bigg(\underbrace{(\text{Hub } 2 = A) \wedge \neg(\text{Hub } 2 = B) \wedge \neg(\text{Hub } 2 = C) \wedge \neg(\text{Hub } 2 = D)}_{\text{Hub } 2 = A}\Bigg) \\
&\vee \Bigg(\underbrace{\neg(\text{Hub } 2 = A) \wedge (\text{Hub } 2 = B) \wedge \neg(\text{Hub } 2 = C) \wedge \neg(\text{Hub } 2 = D)}_{\text{Hub } 2 = B}\Bigg)\Bigg) \vee \ldots\Bigg] \\
&\wedge \ldots
\end{aligned}
\tag{5.15}
$$

We will first encode all variables to numbers. Theoretically, any numbering will do, but for debugging purposes and plausibility checks it is useful to assign the variables a systematic numbering.

$$\underbrace{0001}_{\text{hub ID}}\underbrace{001}_{\text{hub type}}$$

Figure 5.50: Numeric encoding of variables

Figure 5.50 shows such a (simple) encoding. The hub ID can be assigned arbitrarily according to the local needs, while the hub type is, basically, a numeric translation of the hub types. The only peculiarity applies to hub type C, where several values of Δt

can occur within the same network simultaneously. We therefore need to construct the whole set of possible values of Δt upon preprocessing, and then assign each hub type a corresponding ID. In order to simplify visual recognition (i. e. obtain recognisable naming), we assign hub type C the largest possible namespace in a three-digit naming and put type D to the end of the namespace. This results in the following encoding:

100 Type A

200 Type B

[301,399] Type $C_{\Delta t}$

300 Type D

Note again that the distinction between $C_{\Delta t,1}$ and $C_{\Delta t,2}$ and between D_1 and D_2 is *not* relevant here, i. e. these will be treated equally.

In the example in formula 5.15 we face only one value of Δt for type C hubs, so we can sufficiently code the hub types with $\{100, 200, 301, 300\}$. Formula 5.15 therefore reads:

$$\begin{aligned}\Bigg[\Bigg(\Bigg(&\Big(\underbrace{0001100 \wedge \neg 0001200 \wedge \neg 0001301 \wedge \neg 0001300)}_{\text{Hub } 0001=A}\Big)\\ &\vee \Big(\underbrace{\neg 0001100 \wedge 0001200 \wedge \neg 0001301 \wedge \neg 0001300}_{\text{Hub } 0001=B}\Big)\Bigg)\\ &\wedge \Bigg(\Big(\underbrace{0002100 \wedge \neg 0002200 \wedge \neg 0002301 \wedge \neg 0002300}_{\text{Hub } 0002=A}\Big)\\ &\wedge \Big(\underbrace{\neg 0002100 \wedge 0002200 \wedge \neg 0002301 \wedge \neg 0002300}_{\text{Hub } 0002=B}\Big)\Bigg) \vee \dots\Bigg]\\ &\wedge \dots\end{aligned} \tag{5.16}$$

5.4.3 Boolean Treatment of Interval Parting

Up to now, we have not touched the problem of *interval parting* sketched in section 5.2.1. In a trajectory-based environment, any formulation of two lines parting each other's intervals on a common edge yields a comparatively cumbersome combination of constraints.

However, we can directly construct a Boolean formulation if we add a special case of hubs with their own set of possible hub types. Figure 5.51 shows a setting with an inner-city trunk line where several lines (with branches A and B) share one common edge along which they part each other's intervals. Figure 5.52 depicts possible time relations of the

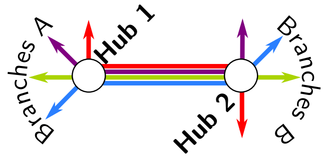


Figure 5.51: Map of the sample combined edge

lines serving the hubs. As can be seen, the combined edge features an interval of $^{T}/_{2 \cdot n_{\mathrm{Tr}}}$ while, in this example, each of the lines runs at interval T.

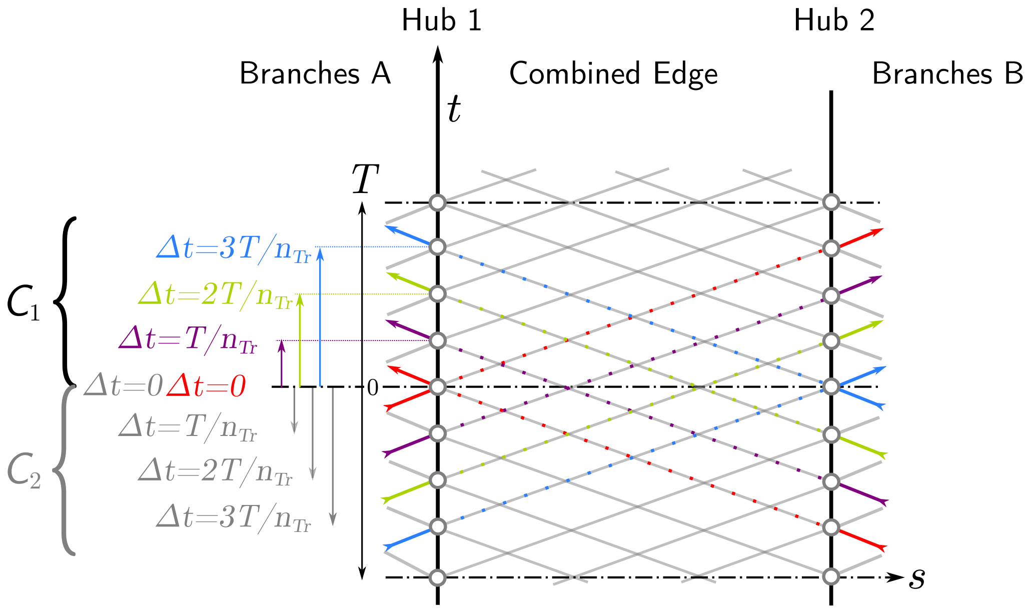


Figure 5.52: Central, combined edge with interval parting

For an operationally feasible result, we require (i) all trajectories in the central edge to be served, and (ii) every adjacent branch to serve Hub 1 (and Hub 2, respectively) at a different point in time.

Again, we can make use of timetable symmetry by changing the `searchspace` in Hub 1 and Hub 2 to a set of hubs of type $C_{\Delta t}$, where the number of Δt in place amounts to $n_{\Delta t} = n_{\mathrm{Tr}}$. The timespan between adjacent Δt consequently amounts to $^{T}/_{n_{\mathrm{Tr}}}$. This leads to a set of type C hubs potentially different from the set present in the rest of the network[36].

Therefore, `searchspace` for these hubs then reads:

[36]Note that edges served by interval parting are *not* touched by the exclusion of intervals $T_i < t_{\mathrm{tr,max}}$ set up in section 5.3.1.

$$\mathtt{searchspace} = \bigcup_{m=0}^{\lfloor n_{\mathrm{Tr}}/2 \rfloor} C_{m \cdot T/n_{\mathrm{Tr}},1} \cup C_{m \cdot T/n_{\mathrm{Tr}},2}$$

which is why we can simply use the existing riding time calculation algorithm, albeit with different `searchspace`.

Note that, instead of denoting the whole range $0 \leq \Delta t < T$ for obtaining values for Δt, we only use $0 \leq \Delta t < T/2$. This results from the split of directional transfer hubs into $C_{\Delta t,1}$ and $C_{\Delta t,2}$ as described in sections 5.3.1 and 5.3.5, which renders $T/2$ sufficient for hub type description. In figure 5.52, we can actually spot that all coloured trajectories serve the hub with type $C_{\Delta t,1}$, while the grey trajectories do so with type $C_{\Delta t,2}$.

Additionally, we do not include these lines in the hub type conversion process, but carry out all computations on the interval level of T. We do therefore not rewrite hub type C_0 to A, $C_{T/2}$ to B, and type $C_{T/4}$ to D, as we would in the standard conversion process, since all Boolean treatment happens strictly at the one hub in question and strictly with the limited set of values for Δt. Instead, the definition of $C_{\Delta t,1}$ and $C_{\Delta t,2}$ leads to the degeneration $C_{0,1} = C_{0,2}$ and $C_{T/2,1} = C_{T/2,2}$[37].

To obtain an interval parting as requested, we must change the way the Boolean formula is set up for the involved hubs: First, we must drop the requirement from equation 5.15 that a hub be served as one hub type but *not* another. Second, we need to establish a bracket formula for each hub to ensure only *one train system at a time* serves the hub in question. While in general, this cannot be accomplished with plain Boolean logic, we can, again, use the peculiarities of the hub structure we require: Since the `searchspace` contains exactly as many possibilities as there are connecting train systems by definition, we simply need to require a hub to be of *all* the types set up beforehand; if we find out one is not satisfied, this automatically renders a second hub type served at least twice. A possible bracket formula for a hub with four adjacent lines at $T_i = 60$ each (i. e. a 15-minute headway on the common edge) reads:

$$(\text{Hub } 1 = C_{0,1}) \wedge (\text{Hub } 1 = C_{15,1}) \wedge (\text{Hub } 1 = C_{0,2}) \wedge (\text{Hub } 1 = C_{15,2}) \tag{5.17}$$

Note that this bracket formula can be simply appended to the Boolean formula we obtained beforehand.

Since we handle long planning horizons and comparatively large variations of actual riding times (see section 2.2.3), *one*, fixed, set of trajectories for the common edge suffices for the definition of Δt when $T_i \leq t_{\mathrm{tr,max}}$. For $T_i > t_{\mathrm{tr,max}}$ (and as a general solution), however, the set of hinges for the attachment of the branches can be shifted by a starting value of $0 \leq t_{\mathrm{shift}} < \Delta t$.

[37] We do not touch on mixed interval partings (such as 15+30+30=7.5 min) here, since we cannot ensure that one instance of a denser intervals interferes with another of the less dense interval. See section 8.3 for this and other thoughts on further research.

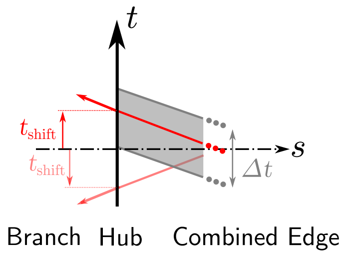


Figure 5.53: Range of values for t_{shift}

Figure 5.53 depicts the possibilities for t_{shift}. The range of possibilities is shaded in grey, while the red trajectory depicts this type of shift. Note that, by the symmetrical nature of the timetable, any shift of one trajectory *from* the axis of symmetry also triggers the complementary trajectory to shift *from* the axis of symmetry, i. e. in the opposite direction.

By the Boolean nature of the algorithm, we need a way to discretise the set to a finite, small number of possibilities. It is, additionally, sufficient to consider only $t_{\text{shift}} \leq t_{\text{tr,max}}$ by the nature of $t_{\text{tr,max}}$. We set

$$t_{\text{shift}} = \frac{\Delta t}{\left\lceil \frac{\Delta t}{t_{\text{tr,max}}} \right\rceil} \tag{5.18}$$

Thus, we obtain a t_{shift} that (i) is small enough to allow for fine-grained possibilities and (ii) yields a number of possibilities small enough not to obstruct the search algorithm. Note that the ceiling function in $\lceil \Delta t / t_{\text{tr,max}} \rceil$ prevents the search for shifts when $\Delta t \leq t_{\text{tr,max}}$ by definition, since $\lceil \Delta t / t_{\text{tr,max}} \rceil = 1$ for all $\Delta t \leq t_{\text{tr,max}}$. Table 5.6 shows examples of common relations between T_i, n_{Tr}, t_{shift}, and sets of Δt corresponding to the respective t_{shift}.

T_i	n_{Tr}	Δt	$\lceil \Delta t / t_{\text{tr,max}} \rceil$	t_{shift}	Δt-sets
60	2	30	3	10	$\{0\}, \{10\}, \{20\}$
60	3	20	2	10	$\{0, 20\}, \{10, 30\}$
60	4	15	2	7.5	$\{0, 15\}, \{7.5, 22.5\}$
30	2	15	2	7.5	$\{0\}, \{7.5\}$
30	3	10	1	10	$\{0, 10\}$
30	4	7.5	1	7.5	$\{0, 7.5\}$
30	8	3.75	1	3.75	$\{0, 3.75, 7.5, 11.25\}$

Table 5.6: Examples for common relations between interval T_i, number of trains n_{Tr}, t_{shift}, and the number of sets of Δt, at $t_{\text{tr,max}} = 10\,\text{min}$.

This extends the approach stated above. First, the `searchspace` is extended further when $|t_{\text{shift}}| > 1$:

$$\texttt{searchspace} = \bigcup_{k=0}^{\lceil \Delta t / t_{\mathrm{tr,max}} \rceil} \bigcup_{m=0}^{\lfloor n_{\mathrm{Tr}}/2 \rfloor} C_{m \cdot T / n_{\mathrm{Tr}} + k \cdot t_{\mathrm{shift}},1} \cup C_{m \cdot T / n_{\mathrm{Tr}} + k \cdot t_{\mathrm{shift}},2}$$

Consequently, the bracket function is also extended, so clause 5.19 (which equals the third line of table 5.6) then reads

$$\begin{aligned} &((\text{Hub } 1 = C_{0,1}) \wedge (\text{Hub } 1 = C_{15,1}) \wedge (\text{Hub } 1 = C_{0,2}) \wedge (\text{Hub } 1 = C_{15,2})) \\ &\quad \vee ((\text{Hub } 1 = C_{7.5,1}) \wedge (\text{Hub } 1 = C_{22.5,1}) \wedge (\text{Hub } 1 = C_{7.5,2}) \wedge (\text{Hub } 1 = C_{22.5,2})) \end{aligned} \tag{5.19}$$

This way, `searchspace` and the bracket formula are linked and the Boolean function will be satisfiable if, and only if, all trajectories arrive at the hub in question with the same t_{shift}.

Finally, we need to offer a restriction to allow for classical branching in a semi hub, i. e. a transfer between branchings if there are only two of them (as described in section 5.1.2 and depicted in figures 5.6a and 5.6b), thus actively requiring a hub to be of type D.

By the nature of a semi hub at branchings, this setting requires $n_{\mathrm{Tr}} = 2$. Furthermore, there is no need for t_{shift}, since the situation works with hub type D only. Therefore, this setting can be sufficiently modelled with (i) unmodified, i. e. standard `searchspace`, (ii) unmodified truth function for the train system, and (iii) a bracket function that reads (for *Hub X* being such a semi hub):

$$(\text{Hub X} = D_1) \wedge (\text{Hub X} = D_2) \tag{5.20}$$

Therefore, when a transfer between two branches is requested, no further action besides a simple bracket function needs to be taken.

5.4.4 Boolean Treatment of Hub Pairs

As already mentioned in section 5.3.1, we need to restrict hub pairs to types $\{A, B\}$. Just as mentioned for interval parting, this can be done by appending the following bracket function to the truth function (considering *Hub Y* is the hub pair in question):

$$(\text{Hub Y} = A) \wedge (\text{Hub Y} = B) \tag{5.21}$$

This way, no further differentiation as to how hub pairs are treated needs to be made.

5.4.5 Hub Type Conflict Diagnosis

Upon creation of the truth function, any SAT solver will be able to check it for satisfiability. However, satisfiability will rarely be the case a priori. It is likely to rather obtain an unsatisfiable formula—demanding for a smart way to diagnose and, finally, resolve hub type conflicts.

Note that *hub type conflicts* tackled in this section have nothing in common with *route conflicts* found in microscopic railway modelling (see Pachl 2011: 93f.). A *hub type conflict* borrows its naming from *conflict diagnosis* in Anomaly Management in software engineering, where *conflicts* are "[...] subsets of the knowledge base that are responsible for a faulty behavior" (Felfernig, Reiterer, et al. 2014: 73). As such, it is a situation where the initial setup of hub type combinations, i. e. the `hublist` trajectories, prevent the creation of a network-wide set of compatible hub types.

The FastDiag algorithm as presented by Felfernig, Reiterer, et al. is a *divide and conquer* approach to handle over-constrained problems. It yields a *minimal conflict set*, i. e. we obtain a minimal set of conflicts that prevent the formula from being satisfiable. This way, we can spot those parts of the truth formula (and thus those restrictions of the timetable constructions) to be changed in order to make the network timetable feasible (Felfernig, Reiterer, et al. 2014: 82ff.).

FastDiag (see the algorithm in the appendix) splits the set of literals roughly in half and checks each part for consistency[38], i. e. satisfiability. As soon as one subset is satisfiable, it is removed from the check, while persistently insatisfiable subsets remain in the evaluation process. By recursively applying the algorithm, a *minimal conflict set* is finally obtained (Felfernig, Reiterer, et al. 2014: 82ff.).

Considering the network and timetable graph in figure 5.48, for the environment of *Hub 2*, the literals as in equation 5.22 are present (colours as in graph, Boolean terms simplified for readability).

$$
\begin{aligned}
&\underbrace{(((\text{Hub } 1 = A) \wedge (\text{Hub } 2 = A)) \vee ((\text{Hub } 1 = D) \wedge (\text{Hub } 2 = D)) \vee ((\text{Hub } 1 = B) \wedge (\text{Hub } 2 = B)))}_{\text{c1}} \\
&\wedge \underbrace{(((\text{Hub } 1 = A) \wedge (\text{Hub } 2 = D)))}_{\text{c2}} \wedge \underbrace{(((\text{Hub } 1 = D) \wedge (\text{Hub } 2 = B)))}_{\text{c3}} \\
&\wedge \underbrace{(((\text{Hub } 2 = D) \wedge (\text{Hub } 3 = A)) \vee ((\text{Hub } 2 = D) \wedge (\text{Hub } 3 = B)))}_{\text{c4}} \\
&\wedge \underbrace{(((\text{Hub } 2 = A) \wedge (\text{Hub } 3 = A)) \vee ((\text{Hub } 2 = D) \wedge (\text{Hub } 3 = D)) \vee ((\text{Hub } 2 = B) \wedge (\text{Hub } 3 = B)))}_{\text{c5}}
\end{aligned}
\tag{5.22}
$$

[38] Naming by Felfernig, Reiterer, et al. 2014

In this excerpt of the complete truth formula, we can spot that Hub 2 is to be of type A, D, or B *and* D *and* B *and* D *and* A, D, or B, i. e. a contradiction. Note that for practical application, knowledge about the whole network is crucial, since every literal touches two hubs at a time. This example (looking at one hub within each literal only) is therefore just to exemplify the procedure of the algorithm.

Figure 5.54 shows the execution path of FastDiag on this example. We first split the set of literals roughly in half. Each step after a split checks whether all literals in the current other half are satisfiable. If they are, we do not touch the current half, since we can consider it satisfiable. If they are not, either half is checked further.

In this case, the formula will be first split into {c1, c2, c3} and {c4, c5}[39].

The first part, looking at Hub 2 only, yields a hub type conflict, since c2 and c3 contradict in asking Hub 2 to be of type D *and* B. We must therefore continue with this "half" and split it again, this time {c1, c2} and {c3}. We can now see that the first "half" is satisfiable (with the solution (Hub 2 = D)). The second half is a singleton, i. e. consists of one literal only. By definition, this must be the conflicting literal in question. Literal c3 is then passed over to the algorithm again.

Then, {c4, c5} are checked for satisfiability, which is the case (solution (Hub 2 = D)). Therefore, an empty set $\emptyset$ is passed over as hub type conflict set.

This means that {c3} is the minimal conflict set, i. e. train system IC12 causes a problem, since it cannot serve Hub 2 with type D as required by the other train systems.

With this algorithm, a minimal hub type conflict set can be spotted fast and precisely. If these hub type conflicts are solved, the timetable becomes a *target timetable*. However, there are two major reasons why *several* conflict sets are required:

1. Not all hub type conflicts found might yield a solution in the timetable view (see section 5.4.5). Therefore, choosing a different minimal conflict set might ease the hub type conflict resolution.

2. According to the design process in which the Target Timetable Phase is embedded, a *set* of target timetables, rather than only one, is required, in order to cope with structure or timetable development changes during the design and implementation period.

Therefore, the FastDiag algorithm shall be embedded within a breadth-first *HSDAG* algorithm. This algorithm constructs an envelope around FastDiag to systematically check subsequent conflict sets (Felfernig and Schubert 2010: 4f.).

In figure 5.54 we can see that the HSDAG wrapper invokes FastDiag again, but with {c3} removed. This way, the procedure can be restarted. However, when a set is found to be satisfiable, c3 is appended again to check whether satisfiability persists. As can be seen, this yields {c2, c4} (solution (Hub 2 = B)) as the second conflict set. The final run,

[39]Note that the first "half" is the number of clauses is odd.

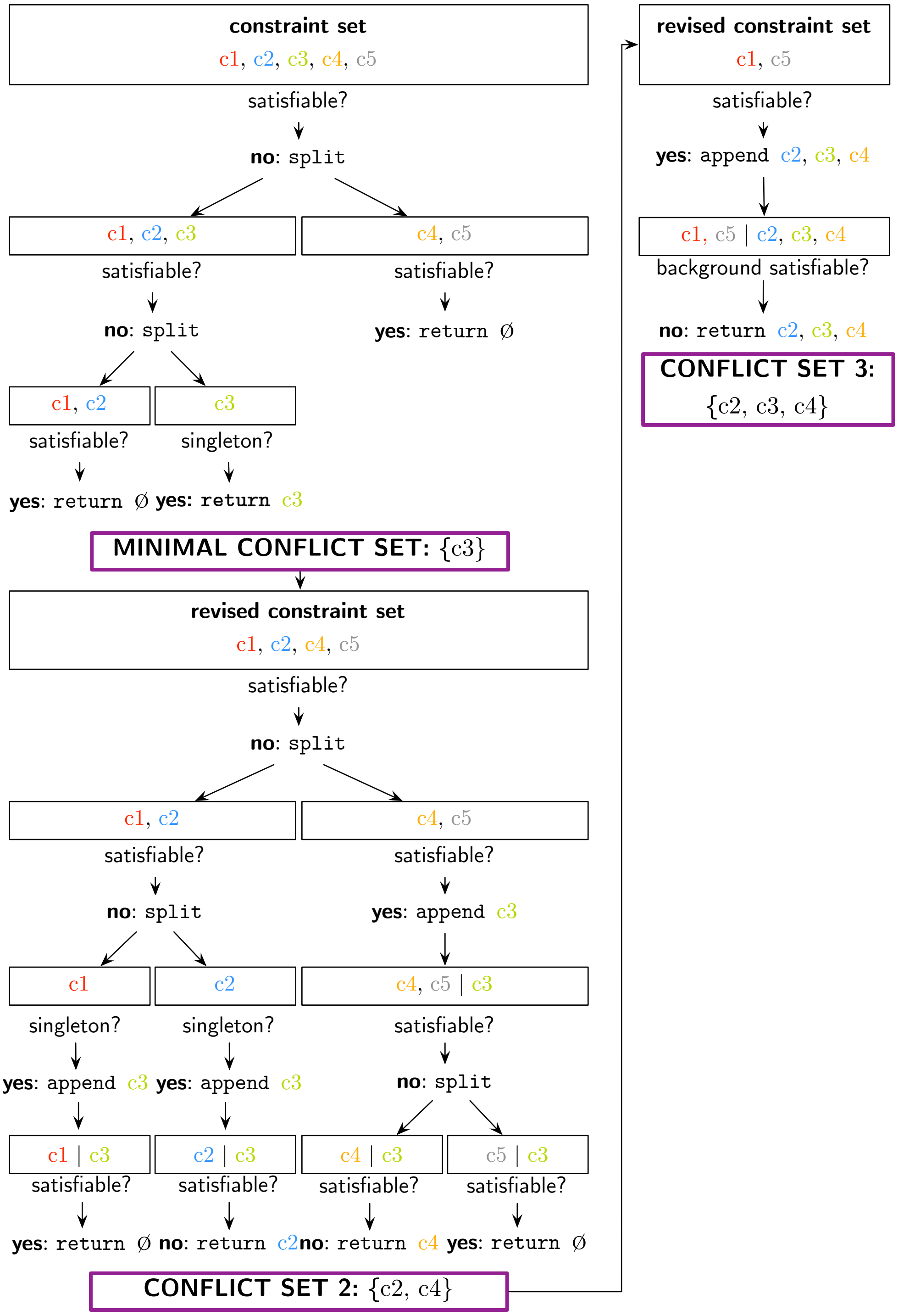


Figure 5.54: Execution path for HSDAG-wrapped FastDiag and the constraints in the example, only relevant checks displayed; based upon Felfernig and Schubert 2010 and Felfernig, Reiterer, et al. 2014.

removing {c2, c3, c4} and invoking FastDiag again, yields a *satisfiable* truth formula right away. However, when appending {c2, c3, c4}, we find that these three literals (the *background literals*) are unsatisfiable, rather than the two in question. This leaves {c2, c3, c4} (solution (Hub 2 = A)) as the last conflict set.

It is important to note that a hub type conflict resolution, on the level presented in this section, can only be carried out by *deleting* the literal in question. This is a valid procedure in terms of achieving satisfiability in the truth function, but is *not* sufficient in a timetable construction process—deleting a literal is equal to deleting a train system. Therefore, the hub type conflicts needs to be solved by timetable construction methods rather than Boolean algebra.

5.4.6 Hub Type Conflict Resolution

The last step to obtain a set of target timetables is to resolve the hub type conflicts found. Each satisfiable, i. e. conflict-free, Boolean formula is equal to one target timetable.

In order to resolve the hub type conflicts found, they have to be traced back to the hub type combination. Since the truth formula has been constructed to allow for a transparent backtracking, we can simply collect the retrieved conflict sets and check them against the timetable obtained in the riding time calculation process.

The hub type conflict resolution itself is carried out with standard timetable construction methods, since there are numerous possibilities to adjust the trajectories in question.

In trying to resolve a hub type conflict in question, there are several ways to obtain a proper hub service for the train systems in question:

1. Speed up or slow down a train system so as to serve the hub with a different type.

   This essentially means expanding the funnel of possible trajectories to allow for a lower $v_{\min}$ or a higher $v_{\max}$ than initially set. Either relaxation needs a check for ridership reaction or technical feasibility, respectively.

   Figure 5.55 shows a trajectory found when applying $v_{\min} \leq v_r \leq v_{\max}$ that allows for a service of *Hub 2* with type B only. As indicated at *Hub 2*, all other lines in question can serve the hub with type D only. To be able to reach a feasible result, the train system in question needs to be sped up to $v_r > v_{\max}$, which needs a technical evaluation of this possibility. Or, otherwise, it could be slowed down to $v_r < v_{\min}$, which could result in a negative ridership reaction and thus needs to be evaluated. This is done by checking the amount of passengers bound for *Hub 2* directly against the amount of passengers to transfer.

2. Join two train systems of similar train type to form one line with a jointly served, denser interval.

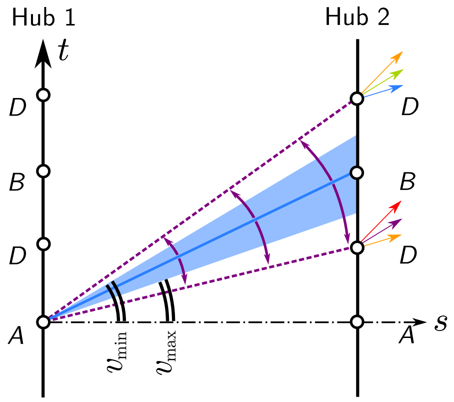


Figure 5.55: Change of v_r in order to change hub service

When train systems with low intervals face comparatively narrow riding speed ranges, the solution space for valid hub type combinations is only small. Therefore, a proper hub service on either end of an edge might be rendered impossible. If two train systems of similar train type are combined, one of the two train systems can serve one hub, while the other train system serves the other hub. Note that this procedure essentially cuts transport relations, since any travel chain with an interval $T_{\text{chain}} \geq T_{\text{joint}}$ starting *and* ending outside the edge, but routed across the edge, is broken at one end of the edge (see section 5.3.3). However, if the combined edge is long enough and most travel relations start and/or end within the combined edge, this can serve as a useful solution strategy.

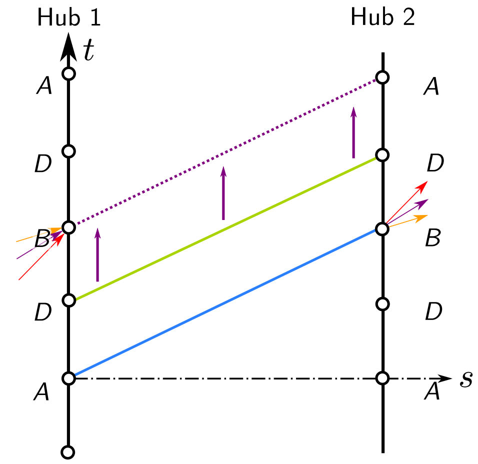
Figure 5.56a shows such a situation: While IC11 can serve only *Hub 1* properly, the IC12 can only do so at *Hub 2*. Combining the two yields a valid solution.

As described in section 5.1.3, train systems with intervals $T_i/2 \leq t_{\text{tr,max}}$ may serve hubs selectively. If the joining of two train systems creates one line with such a dense interval, a shift by $T_{\text{joint}}/2$ might allow for a selective service of both hubs of the edge without the need of a riding time modification. Note that this solution requires a counter check at the points where the joint edges split again, so as to account for potentially broken transfers.

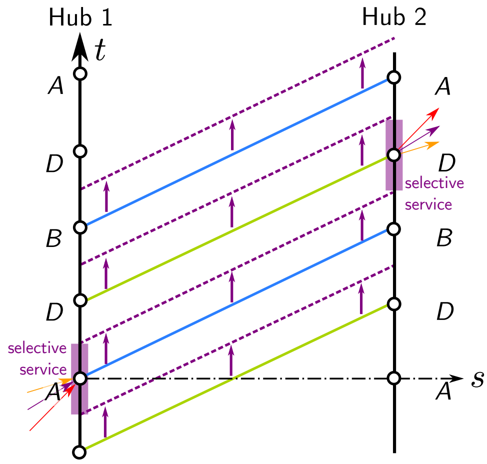
Figure 5.56b shows IC11, denoted in half the interval for illustrative reasons, reaching *Hub 2* to serve it with type *B* only. However, all other lines serve it with type *D*. IC12, also at half the interval, manages to reach Hub 2 with type *D*. However, if the two are combined and shifted by half the combined interval (purple trajectories), they can serve both *Hub 1* and *Hub 2* selectively.

3. Change the interval of a train system to serve either hub.

   Similar to the combination of two train systems to one with a denser interval, we might as well change the interval of the train system in question to allow for a hub service at either end. Halving an interval, i. e. $T_{i,\text{modified}} = T_i/2$, equals the



(a) One hub served by each train system



(b) Joint selective hub service

Figure 5.56: Combination of two train systems

situation described beforehand and depicted in figure 5.56a. However, the interval change can be used more extensively. Since the focus in this stage of the design already allows for a precise prediction of consequences, the interval at this stage can even be changed by a divisor that has been ruled out for compatibility reasons. This procedure can then be used to relieve initially tight riding time requirements and thus produce a valid hub service.

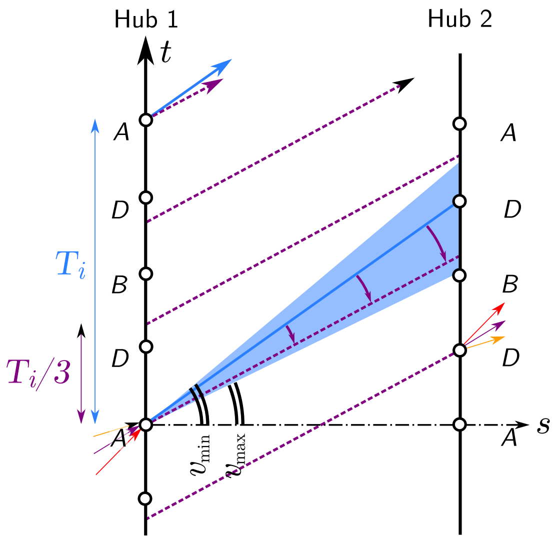


Figure 5.57: Interval change to an a priori incompatible interval to relieve riding time requirements

Figure 5.57 shows such a case: Train system IC11 with interval T_i cannot serve *Hub 2* properly. The riding speed range does not allow for a change either. However, if the interval in divided into thirds, there appears a solution within the riding speed range: The trajectory starting before the base trajectory can serve *Hub 2* with Type D_1 as required, but does not serve *Hub 1* properly. The base trajectory, sped up slightly, serves *Hub 1* with type A as requested, but not *Hub 2*. The third trajectory does not serve either hub and could even be cancelled completely, leaving the interval of the train system to be $\{T_i/3, 2 \cdot T_i/3\}$.

4. Drop the hub service of the train system in question completely.

   For hubs with high centrality, i. e. main train stations with important connections to urban transport and a low amount of transfer passengers, one valid solution is not to take any action at all: If the node flows from the Service Intention Phase show a large percentage of passengers bound to the hub itself, a non-transfer might not even harm the overall demand model at large. In this case, the overall time gain for reaching a hub earlier might (or might not) stem from urban passengers only, but the strict adherence to the target riding time for full hub service can be dropped.

   Note that this does *not* imply the deletion of a train system, as noted in the final remark of section 5.4.5; Rather, the hub on one end of a line can still be served, while the other is dropped.

For the evaluation of competing hub type conflict resolutions, we need to make use of the modified node flows obtained in the Service Intention Phase. As already indicated in section 2.3.5, these node flows imply the importance of the train systems in question. In order to sort the hub type conflicts by demand, the node flow for the hub in question needs to be disintegrated into *local flows* $q_{\text{mod, local}}$ starting/ending in the hub and *transfer flows* $q_{\text{mod, tr}}$ that take part in transfers. When a conflict set is detected, the only information available (and necessary at that point) is that one train system does *not* reach a transfer hub. Therefore, it suffices to compute $\sum q_{\text{mod, tr}}$ per train system and, subsequently, sum up all train systems included in one hub type conflict. In this case, a *low* number of passengers is preferable. This means more transfers are satisfied right away and solutions that include potential breaks of travel chains can be justified more easily. This way, a hierarchy of which hub type conflicts to tackle first is established.

Note that it is possible to obtain one conflict set ranked higher in terms of passenger flows, while another is ranked better in terms of (rough) investment costs. Since either possibility leads a way to a valid target timetable, the respective advantages need to be checked against each other during the Feasible Timetable Phase.

In the given example from section 5.4.5, three conflict sets have been obtained:

{c3} $(0001351 \wedge 0002200)$,

{c2, c4} $((0001100 \wedge 0002350) \wedge ((0002350 \wedge 0003100) \vee (0002351 \wedge 0003200)))$, and

{c2, c3, c4} $((0001100 \wedge 0002350) \wedge (0001351 \wedge 0002200) \wedge ((0002350 \wedge 0003100) \vee (0002351 \wedge 0003200)))$.

Figure 5.58 shows the trajectories from figure 5.48 redrawn with respect to *Hub 2* and with mirrored and completed trajectories between *Hub 2* and *Hub 3* for better readability. As can be seen, the train system IC12 can only serve *Hub 2* with type *B*, IC11 can only serve it with type *D*, and R21 can only serve it with type *D*. S11 and S21 can serve either hub with either type due to their dense intervals.

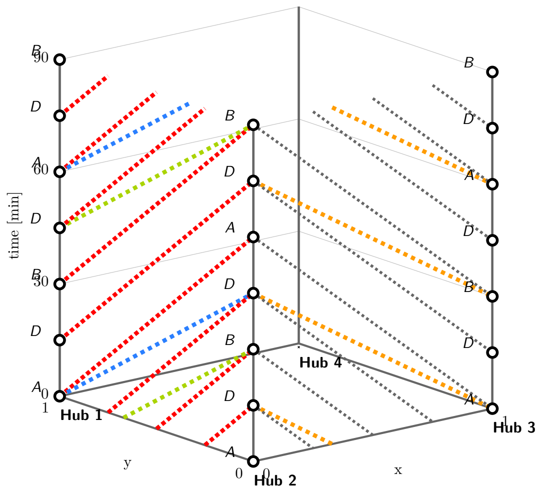
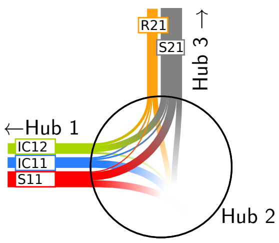


Figure 5.58: Reduced sample network for hub type conflict resolution

Furthermore, modified node flows as in figure 5.59a can be obtained. With the individual intervals and the travel distance weights from section 2.1, we can obtain modified node flows as stated in table 5.59b.



(a) Node flows

	S11	IC11	IC12	R21	S21	$q_{\text{local, mod}}$	$\sum q_{\text{tr, mod}}$
S11		–	–	80	525	455	605
IC11			–	670	503	9,380	1,173
IC12				1,675	2,010	2,010	3,685
R21					–	800	2,425
S21						350	3,038

(b) OD matrix

Figure 5.59: Modified node flows from Service Intention Phase

Summing up $\sum q_{\text{tr, mod}}$ for each minimal conflict set, we can obtain the order

{c2, c4} $\sum q_{\text{tr, mod}} = 3,598$

{c3} $\sum q_{\text{tr, mod}} = 3,685$

{c2, c3, c4} $\sum q_{\text{tr, mod}} = 7,283$

As can be seen, the amount of weighted passengers affected by the minimal conflict set {c3} is (albeit slightly) *higher* than the amount affected by the second set {c2, c4}. If *no* timetable solution for the conflict is found, i. e. no change in the system is desired or possible, it is preferable to make the hub type *B* and drop the transfers for trains IC11 and R21, because the least amount of weighted passengers is involved.

Any further solutions that make a transfer work will rank better in terms of $\sum q_{\text{tr, mod}}$, since all aim for a transfer improvement. However, the side effects can be judged: if a solution improves a transfer but has positive/negative effects for local passengers by increasing/decreasing the riding time along an edge, the side effect can be quantified.

In accordance with the possibilities presented before, we can translate the solutions given to resolve the hub type conflicts. For the minimal conflict set {c3}, the solutions are:

1a. Speeding up IC12 to reach Hub 2 with type *D*, allowing for all transfers and additionally improving the situation for the weighted 2,010 weighted local passengers bound for Hub 2.

1b. Slowing down IC12 to reach Hub 2 with type *D* half an interval later. This solution would essentially transform IC12 into one instance of S11, turning one of the two obsolete from a system point of view[40]. While the transfer passengers do benefit from the improved transfer situation, the lower riding speed worsens the situation for the 2,010 weighted local passengers[41].

2a. Joining IC11 and IC12. This requires one of the train systems to be shifted by $T_i/4$ to construct an even distribution of trajectories. Additionally, in this example the view onto *Hub 1* is omitted; therefore, IC12 might lose transfers at *Hub 1* if it is shifted. With only *Hub 2* in view, this setting neither worsens the situation for local passengers nor for those of IC11 and improves the situation for the transfer passengers of IC12.

2b. Joining IC12 and IC11 and shifting them so that both hubs are served selectively. This can be done when $T_{\text{joint}}/2 \leq t_{\text{tr,max}}$. In this case, both IC12 and IC11 would serve both *Hub 1* and *Hub 2* selectively. This setting does not worsen the situation for the local passengers but worsens the transfer conditions for IC11 passengers and improves them for IC12 passengers, thus affecting 1,172 negatively.

3. Altering the interval of IC12 to allow for a service on either end. Without a shift or speed change, half the interval ($T_{i,\text{new}} = T_i/2$) would already allow for a selective service at both *Hub 1* and *Hub 2*. As noted already, the network context needs to

[40] We can have IC12 run in one of the slots of S11 and drop the corresponding S11 service in that slot.

[41] Note that the negative effect of a lower riding speed can also affect the transfer passengers in a network view, which is not tackled here for brevity reasons.

be kept in mind, so that the transfers at Hub 1 are not lost. This solution does not affect local passengers.

Not surprisingly, the presumably most costly alternative (speeding up[42] IC12 beyond $v_{\max}$) seems to be the most beneficial one. More notably, the alternatives that do *not* include a speed increase, but improve the transfer situation, rank second in having no negative effects on local passengers. However, the last option requires double the service offer for IC12.

For the second conflict set, {c2, c4}, both IC11 and R21 need to be altered. The alternatives are:

1. Speeding up or slowing down R21: The same restrictions as for IC12 apply: Speeding up by $T_i/2$ might work, but slowing down by that amount would only turn R21 to one trajectory of S21. If R21 is sped up to serve *Hub 2* with type B, not only transfer passengers, but also the 800 weighted local passengers benefit from that solution. At the same time, also IC11 needs to be sped up likewise, so another 9,380 weighted local passengers will benefit.

2. Joining two train systems will not apply for R21, since there is only one system per train type. However, it might be shifted by $T_i/2$ to allow for a selective hub service at both *Hub 2* and *Hub 3*. While the transfer passengers will perceive a difference, the local passengers will not. However, in any case, IC11 needs to be sped up, too (affecting 10,553 weighted passengers positively).

Compared to the first conflict set, the two measures presented here affect significantly more (weighted) passengers positively as a side effect. However, either solution requires that IC11 be sped up, which is likely to cause larger infrastructure measures.

Finally, the last conflict set, {c2, c3, c4}, will not be described here, since all possibilities described already apply analogously.

In total, option 2a of conflict set c1 can be considered least measure-intensive at that point, since a mere timetable measure already makes the hub work without harming local passengers.

Of course, a detailed view onto the demand investigation needs to be carried out in the Feasible Timetable Phase, when all dependencies between riding time, interval, and transfer quality can be judged properly; but with this basic approach, a preliminary judgement of the alternatives can be carried out.

[42]Note that speeding up a fast train system usually requires great-scale infrastructure measures (Lai et al. 2011: 7f.).

5.5 Preliminary Infrastructure Dimensioning

After the resolution of all hub type conflict sets, we obtain a set of target timetables. For each edge, we know by then:

1. Number of train systems per edge
2. Type of hub service per train system per edge (i. e. departure and arrival times at each hub)
3. Target riding time per train system per edge

From this information, we need to deduct a *preliminary* target infrastructure. This should *not* depict a final target infrastructure yet, since there is not yet any information about which infrastructure, operational, vehicle, or other measures need to be taken to fulfil the timetable requirements. Furthermore, there is still a *set* of target timetables which needs to be kept in mind when handing over the timetable data to the feasible timetable phase.

However, we can deduct some information about the infrastructure from the target timetable already and can pass it on right away. This information shall be called *functional infrastructure requirements* and shall comprise the following:

1. Rough location and length of sections with more than one track (in case of train crossings) or two tracks (in case of train overtakings),
2. Number of parallel exits/approaches of hubs, and
3. Target riding times per train system.

Most of the last information can be directly retrieved from the riding time calculation, since every possible trajectory has been calculated beforehand. Only trajectories altered in the hub type conflict resolution process need to be added in order to obtain all target riding times.

The location and length of multi-tracked sections can be derived from the target timetable. For single-track sections, both crossings and overtakings need to be calculated, for double-track sections the calculation of overtakings is sufficient.

Note that in this section, we will handle *route conflicts* in the microscopic sense (see Pachl 2011: 93f.) rather than *hub type conflicts* as described before.

5.5.1 Dimensioning with Scan Line

As already presented in several places throughout this work, the approach of *Inverse Capacity Determination* (Wieczorek 2006) allows for an infrastructure dimensioning as soon as a target timetable is present.

However, the approach, as originally presented, requires a timetable exact to the minute in order to work. From the Target Timetable Phase, the result is an edge riding time, i. e. trajectories departing right at the defined hub time per hub type. Therefore, both the input data and the methodology must be altered for this approach.

The *parallel and sequential performance requirements* in Wieczorek's methodology are derived by a *Scan Line* approach, where, for each track element, a scan on the time axis is carried out. The performance requirements are determined from the number of events occurring simultaneously. The maximum number of parallel events then defines the number of tracks (or routes, for station gridirons) and the time between the events (i. e. sequential events) defines the necessary signal headways.

To make this approach work for the preliminary infrastructure dimensioning, the approach presented here is altered to a *continuous* rather than a *discrete* view on infrastructure elements.

For each edge, the width of each target timetable trajectory is extended to match the typical length of a signal block l_{block} with the trajectory in the centre. Contrary to the approach presented by Wieczorek, the parallel and sequential performance requirements, when used in the context described here, can be considered communicating vessels: The longer the chosen signal block length, the more parallel performance requirements will come as a result, and the less critical the sequential ones become[43].

Figure 5.60 shows the continuous scan line approach for an edge with two train systems: the faster train system with interval $T_i = T$, and the slower train system with $T_i = T/4$. The basic trajectories from the Target Timetable Phase (serving *Hub 1* with type *A* and *Hub 2* with type *B*) are shown in thick, continuous lines, while the additional trajectories coming from the interval of the slower train system are depicted in dashed lines. All trajectories are enlarged by adding $l_{\text{block}}/2$ on either side, resulting in the simplified blocking scheme per train.

Highlighted in green are distinctive cross-sections and their respective scan line diagrams. In these diagrams, each train system and each direction is assigned a proper column, and a line in that column denotes a blocked track. This diagram is drawn continuously across the timetable[44], and the corresponding maximum parallel events yield the minimum number of parallel routes. The minimum track layout is depicted below.

Figure 5.60 shows some model cross-sections: (a) shows a stretch where no crossings of trains occur, i. e. a single track would suffice. However, the parallel entry and exit of the train systems creates the need for two tracks. (b) shows the crossing of the slower train system as well as the still present parallel station entry/exit of the faster train system. Therefore, a third track is necessary. (c) shows an ambivalent section: As can be seen,

[43] The selection of a signal block length is, in fact, a predetermination of the sequential performance requirements.

[44] Note that the result can also be viewed as a set of stacked two-dimensional timetable slices, where each train system and direction occupies one slice, such that a view from *Hub 1* towards *Hub 2* yields these cross-sections.

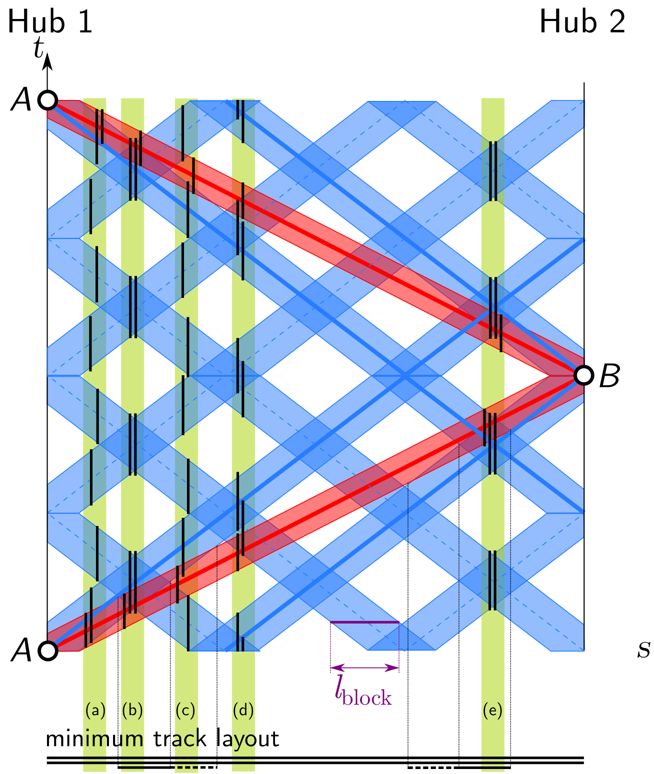


Figure 5.60: Parallel and sequential performance requirements with scan line

the occupancy overlapping between the two train systems is still existent, yet only small. Since the crossing train is not present in this section at the same time, the *minimum* requirement in this section is two tracks only. (d) finally shows the section where two tracks suffice again in any case, since the trajectories of the two train systems can be arranged sequentially already. (e) finally depicts the mirrored situation of (b), just after a new three-track section starts.

Note that this approach yields the *minimum* track layout. This implies a flexible use of the tracks. If the dimensioning were carried out by determining crossings and overtakings separately, it would yield a four-track layout spanning into the hubs. The minimum track layout, however, requires the three tracks to be used flexibly in the direction of overtaking and the two tracks in the end to be used in counterflow upon parallel entry and exit.

In section (c), however, such a flexible use is hard to achieve. Having two tracks in this section essentially means that one of the *Hub 2*-bound trains needs to use the second track for a parallel ride *and* change back to the regular one before the next section. This shows that a minimum track requirement will not automatically lead to an operationally reasonable solution. Therefore, this section will, if no grave constraints prohibit this, be three-tracked, as depicted by the dashed lines in the track layout graph.

5.5.2 Treatment of Common Stretches of Different Edges

The preliminary infrastructure dimensioning would be sufficient for passing on the information to the Feasible Timetable Phase if all edges possessed their own, isolated tracks. This is not the case, since at least train stations form overlappings of edges on common stretches. The information on which edges share common stretches can be retrieved easily from the existing network structures and can be used directly with the scan line approach. Since the length of the common stretches is known, the scan line timetables can simply be overlapped for this length to allow for a judgement of parallel and sequential performance requirements.

Figure 5.61 shows an example where a common stretch is shared by two edges with four train systems each. As can be seen, the *minimum* track layout already requires a five-track section, while a more robust infrastructure layout would require eight parallel tracks. This can be regarded as a starting point for infrastructure design, but will likely require a timetable reconstruction in the Feasible Timetable Phase.

5.5.3 Limitations of Preliminary Infrastructure Dimensioning

As described in section2.3.2, we have so far ignored variances in riding speed along an edge. When it comes to calculating the point of overtaking, such variances become important. We therefore need to quantify the effects of an unevenly distributed riding

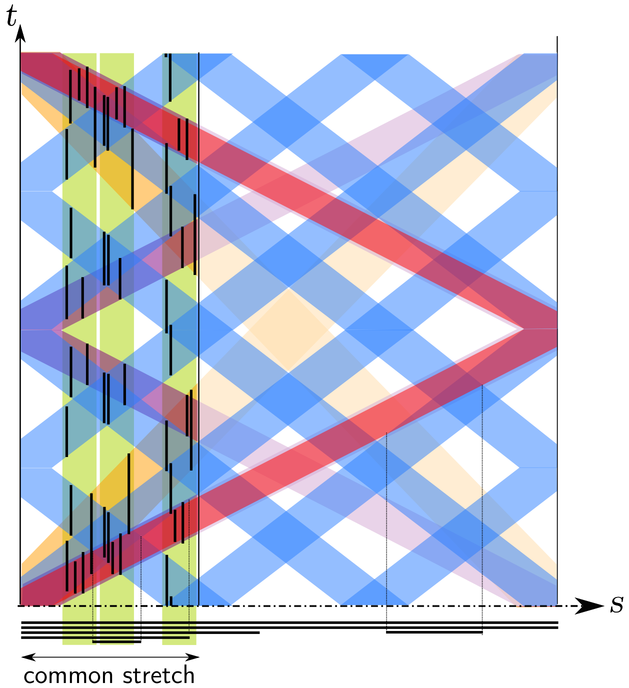


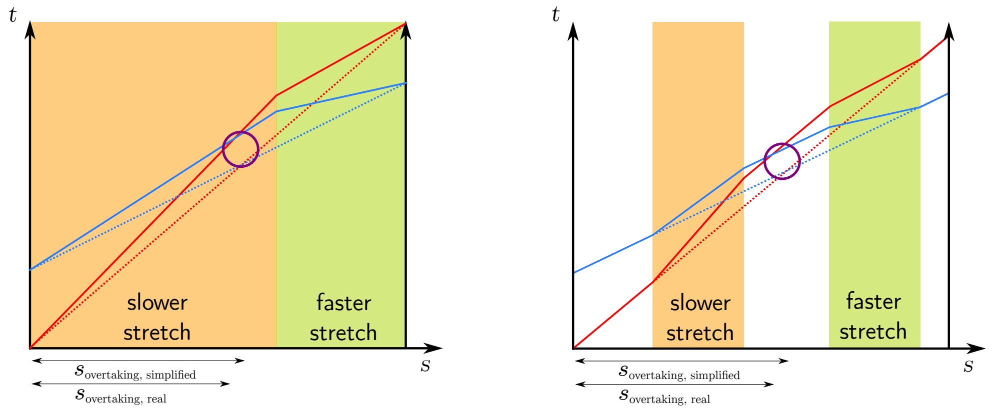
Figure 5.61: Treatment of common stretches with scan line

speed along an edge. Figures 5.62b and 5.62a show these effects. It is important to note that, as long as the hub type on either end of an edge remains unchanged, a faster section always needs to be followed by a slower section (or vice versa) in order to make up for the speed gain in the big picture. If both the faster and the slower section occur outside the point of overtaking, the point will remain in the same place.

However, when (i) the point of overtaking is within a stretch of divergent riding speed (see figure 5.62a) or (ii) the point of overtaking lies between a faster and a slower stretch (see figure 5.62b), the point of overtaking can be shifted. If the point of overtaking is—as described here—used as a preliminary infrastructure dimensioning only, its result can be used as an initial value for the Feasible Timetable Phase. Nevertheless, the result should not be mistaken as a final calculation result.

Finally, figures 5.63a and 5.63a show extreme cases when one of the train trajectories is concave and the other is convex. In this situation, route conflicts and/or overtakings could potentially occur anywhere along the edge. However, if we take a closer look on these situations, they are unlikely to occur at all: This arrangement would require zoning trains (see chapter 5.2.3) from either hub to the centre of the edge *without* a further hub within the centre[45]. As described in chapter 5.2.3, zoning trains require a transfer hub at

[45]This situation *cannot* appear for track-related reasons, since a riding speed variation would affect all trains similarly.

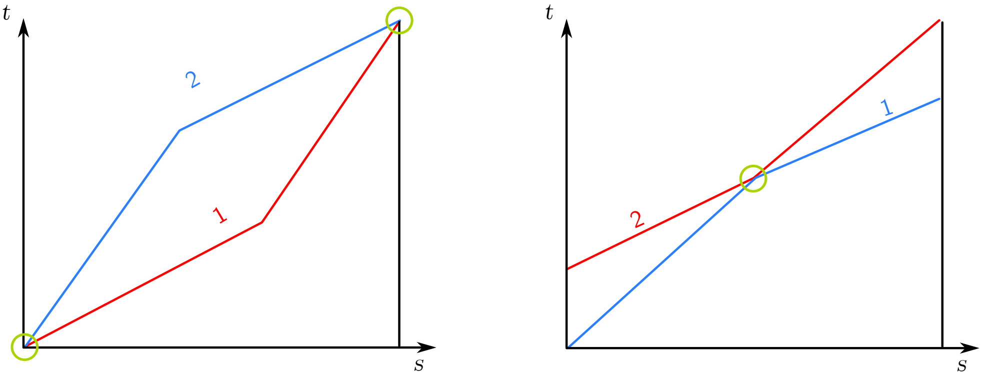


(a) point of overtaking within section with varying riding speed

(b) slower and faster sections outside the point of overtaking

Figure 5.62: Effects of varying riding speeds along an edge

the speed change in the trajectory to work as required, so we can legitimately rule out these situations.



(a) route conflict on both ends

(b) route conflict on edge

Figure 5.63: Hypothetic route conflicts with concave and convex trajectories

For a preliminary dimensioning, multi-tracked sections can be sufficiently modelled with the simplified trajectories as described in section 2.3.2, but need to be treated further within the feasible timetable phase as described.

5.6 Target Timetables for Feasible Timetable Phase

As noted in section 4.5, we need the following data to pass over to the Feasible Timetable Phase:

1. Target Timetables
2. Functional Infrastructure Requirements

As noted before, we obtain a *set* of target timetables, each resembling one approach to fulfil the hub type compatibility requirements. We need this set of target timetables to cope with later disturbances in the design process, be it by changes in the demand, uncertainties in the implementation process, or simply policy changes. These target timetables, though, need to be structured and ranked in order to avoid following *several* strategies at once instead of only one. Only *one* target timetable at a time shall be the design goal, the others serving as alternatives at hand in case of changes underway.

Therefore, the target timetables retrieved in this design phase must be (i) clustered by their hub structure properties and (ii) ranked by their overall benefit. The latter can be done comfortably by adding up the (weighted) amount of satisfied transfers and ranking the timetables that way. The former, however, is more delicate: Consider a situation where all planning focuses on one particular timetable process. Then, one disturbance renders a part of the target timetable impossible. This is where we are bound to fall back to the next target timetable in line. However, this second-ranked timetable might significantly differ from the first one in terms of hub structure. A change in strategy towards this alternative can therefore yield a great number of stranded investments.

If target timetables with similar features, but distinct differences in network parts only, are clustered and then ranked, we can rely upon a set of subsequent alternatives to follow without the need to sacrifice investment decisions already made at large.

The clustering of the target timetables is carried out by comparing the hub types found in the final timetable and clustering timetables with similar features. Since the amount of hubs per network is small, it is fast enough to compare all target timetables mutually. From these comparisons, a graph can be established where each timetable is represented as a vertex and linked to its closest resemblances by an edge, i. e. the less differences there are, the closer two timetables occur within this graph. The edges carry information about which hubs differ between the two timetables. The graph needs to be constructed so to resemble the overall variance of timetables: Edges between timetable versions are drawn only when the difference between two timetables remains in the first quartile of the overall differences[46].

Upon failure to create one target timetable, a neighbourhood search can be carried out to find the closest alternatives. The location of the hub type conflict preventing a target

[46] Of course, this restriction can be changed, but as analysed in section 7.2.3, the first quartile allows for both sufficient possibilities and useful graph creation.

timetable to be constructed is known. Therefore, only those neighbours differing in these respective hubs need to be taken into account for further processing. Finally, amongst the chosen neighbours, the best-ranked alternative can be selected for the further design process. This target timetable graph can be used throughout the further design process, since it sufficiently summarises the information gained in the Target Timetable Phase. If a different target timetable is selected for further investigation, its properties can be simply retrieved from the results of the Target Timetable Phase and used for further processing.

Figure 5.64 shows such a timetable graph. As can be seen, there is a central, high-ranked timetable in the centre, linked to six other variants, of which one, slightly lower-ranked, only differs in three hubs. The second-ranked timetable, however, is on a far end of the graph, differing from the best-ranked in so many hubs that there are three other timetables between them in the graph. So when the creation of the best ranked model fails, the next best possibility is the closer, rather than the better-ranked, one.

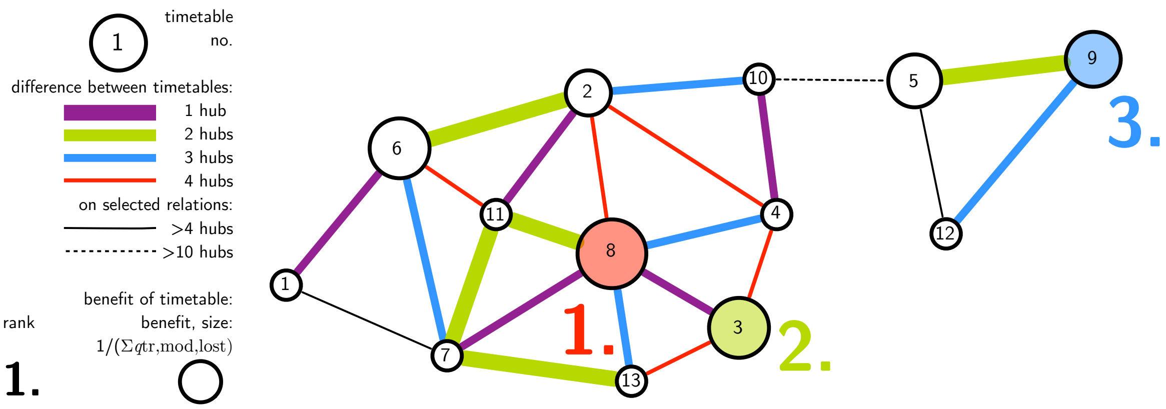


Figure 5.64: Target timetable graph

Every timetable now consists of

1. a hub structure;
2. target riding times per edge and per train system;
3. a preliminary infrastructure;
4. a benefit for ranking; and
5. a similarity neighbourhood.

This information is then used in the Feasible Timetable Phase.

6 Feasible Timetable and Infrastructure

The Target Timetable Phase yields both target timetables and the corresponding preliminary infrastructure. The dimensioning does yield a feasible infrastructure, but only for cases where (i) the target riding time is achievable, (ii) the riding speed is constant, and (iii) the signalling is able to deploy adequately short signal headways to reflect the continuous approach as shown. Since these prerequisites are rarely present in reality, the feasible timetable needs to be designed iteratively.

Note that, upon the creation of an infrastructure predimensioning, the Feasible Timetable Phase might also be replaced by an algorithmic timetable optimisation method (see section 3.1 for an elaboration of respective methods). This way, the predimensioned infrastructure, perhaps with a more generous approach to the number of track elements, can be fed into a timetable optimisation routine and checked for feasibility. To date, all corresponding routines would require a significant feature expansion, but this direction of research seems promising in the context of the design approach presented here. See section 8.3 for thoughts on further research.

6.1 Status Quo

Thus far, no attention has been paid to the actual current timetable situation of a network. For the establishment of a target timetable, this is a valid approach—it makes it possible to obtain a timetable free of subjective restrictions that stem from the status quo. However, a feasible timetable requires an adequate infrastructure to render it so. Therefore, at this point, a reference to the status quo has to be established.

6.1.1 Status Quo Riding Times

This status quo is to contain the information about the current riding time (i) for each edge and (ii) for each train system. To simplify the comparison between a target timetable and the status quo, the train systems are to be simplified to offer the same set of categories as the target timetable[47].

[47] As noted in section 5.2.2, the category of train systems might be misleading. One train system might serve different purposes on subsequent stretches or different train systems might serve the same purpose. Therefore, the respective groups need to be handled like their category in terms of riding time and stopping policy rather than in terms of train system name.

Table 6.1 shows a `statusquolist` for the sample network introduced in the Target Timetable Phase already. If compared to the `hublist` in section 5.3.6, the table shows train system S from *Hub 1* to *Hub 2* and IC from *Hub 3* to *Hub 4* to already feature riding times at or slightly below the target riding time (55 vs. 60 and 45 vs. 45 min). Train system IC from *Hub 1* to *Hub 2* features a riding time that is slightly too high (50 vs. 45 min) and train system R from *Hub 2* to *Hub 3* features a riding time that is far too high (75 vs. 60 min).

Train System	Hub Name 1	Hub Name 2	t_r	$t_{r,\text{target}}$
S	Hub 1	Hub 2	55	60
IC	Hub 1	Hub 2	50	45
R	Hub 2	Hub 3	75	60
IC	Hub 3	Hub 4	45	45
⋮	⋮	⋮	⋮	⋮

Table 6.1: Sample `statusquolist` table.

In order to reduce the manual workload (since this step has to be taken for each target timetable in discussion), it is desirable to retrieve the information, per target timetable, on which train system no significant change in riding time[48] is to be expected, so the measures to be taken in order to obtain the target riding time will be minor[49]. From the definition of $t_{\text{tr,max}}$, the range of tolerable status quo riding times shall be

$$t_{r,\text{target}} - t_{\text{tr,max}} \leq t_{r,\text{status quo, tolerable}} \leq t_{r,\text{target}} \tag{6.1}$$

With this lower bound, we ensure that a status quo riding time is competitive enough when obeying the target riding time and does not have to be slowed down below the current riding time.

In the sample `statusquolist`, this yields train system S from *Hub 1* to *Hub 2* and IC from *Hub 3* to *Hub 4* to be moved to `statusquolist_noaction`, so as to mark them as having an already adequate riding time at first sight.

Furthermore, train system R from *Hub 2* to *Hub 3* features a target riding time of 15 minutes (or 20 %) lower than in the status quo. Walter and Fellendorf as well as Veit, Walter, et al. evaluated various strategies to reduce riding times for regional railways;

[48]Due to the nature of riding time processing in railway engineering (see section 2.2.3), strictly no *over*pass of a target riding time, no matter how slight, can be tolerated here.

[49]Note that in a bigger picture, with a view on the complete edge and network timetable, mutual conflicts might impose extra riding time requirements.

Lai et al. did the same for upgrading conventional lines to high-speed lines. Given a currently adequate infrastructure condition (i. e. only singular slow orders, or a lack thereof, as well as a signalling system that matches the alignment design speed), even highly sophisticated measure combinations (save skipping *all* intermediate stops) allow for riding time reductions of 15 % at most (Walter and Fellendorf 2015: 40, Veit, Walter, et al. 2014, 80ff. Lai et al. 2011). We will therefore set the next threshold to remove target riding times far out of reach:

$$t_{r,\text{status quo, thinkable}} \leq \frac{t_{r,\text{target}}}{1 - p_{\text{reduction}}}, \quad p_{\text{reduction}} := 0,15 \tag{6.2}$$

Completely new alignments, however, can achieve significantly higher values of riding time reduction that rise with increasing line length (Reinold et al. 2012: 49, Ellwanger 2004: 417). Furthermore, on lines with a currently bad infrastructure condition, significantly higher values can be achieved by maintenance actions and thus by lifting slow orders (Berghold 2011: 16, 53). Neither possible new alignments nor bad infrastructure conditions are reflected upon in the input data. Therefore, using a `statusquolist_unattainable` for these train systems marks them *presumably* unattainable, but keeps them for in-depth processing.

In the example, we can move train system R from *Hub 2* to *Hub 3* to `statusquolist_unattainable` to mark it for further processing as presumably unattainable riding times.

Finally, target riding times that are *far too long*, i. e. $t_{t,\text{target}} > t_{r,\text{status quo}} + t_{\text{tr,max}}$, shall also be moved to `statusquolist_unattainable`, since they require similar treatments as riding times that a*far too short*, which might also include a shift to a different target timetable.

This leaves the `statusquolist` as the list of train systems for which action is necessary and presumably possible.

6.1.2 Status Quo Track Layout

In section 5.5, a *target track layout* for each edge has been created. This track layout is created on the basis of a homogeneous speed band and thus on an equally scaled space axis *and* time axis. As noted, this is to allow for a better arrangement of station locations and the modification of riding times rather than a relocation of stations. However, the current track layout only exists on the space axis, which does usually *not* feature a homogeneous speed band. Therefore, for a comparison of the target track layout with the status quo, the target track layout needs to be rescaled according to the *current* riding time distribution.

Figure 6.1 shows such a rescaling. The topmost step depicts the result of the timetable scan line. Since the target timetable features simplified trajectories, the time axis and the space axis can be easily transformed into one another. Provided the target riding

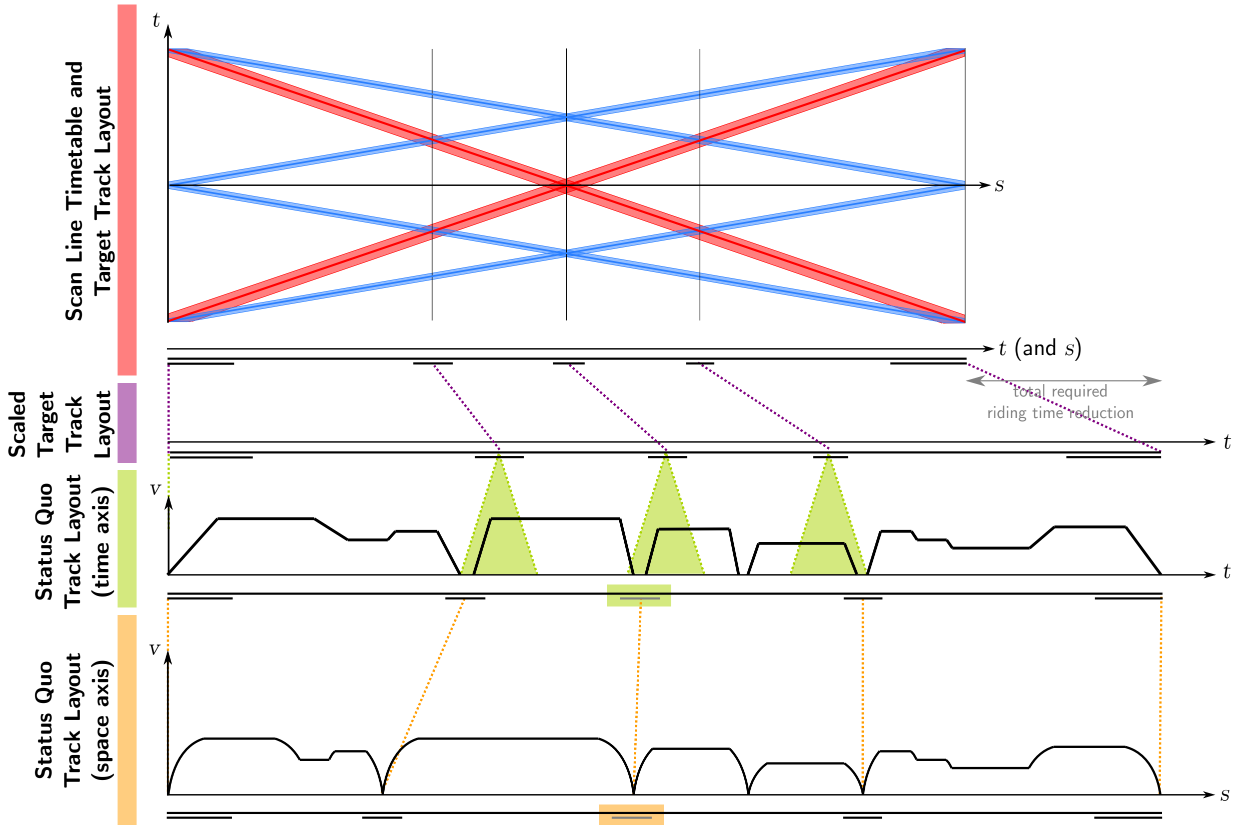


Figure 6.1: Rescaling of target track layout to status quo

time is lower than the current one, the target track layout is to be scaled to the current riding time for one of the existing train systems, i. e. the length difference of the diagram equals the total riding time reduction required. It is not important which current train system to choose in this step, as long as (i) as no zoning train is picked and (ii) the chosen train system remains the reference for further treatment of this edge.

Next, the speed-time graph of the status quo infrastructure and the chosen train system is plotted. With a proximity search around the crossing points in the target track layout, the points in time for crossings on the status quo infrastructure can be aligned with actual stations[50].

Finally, since the stations for crossing are known, this information can be traced back to the speed band and thus the space axis. Note that the example yields the need for an additional crossing track on the second intermediate station.

6.2 Timetable and Infrastructure Construction

Since, for both status quo and target structure, both riding times and track layout are known, the detailed construction of timetable and infrastructure can be triggered.

6.2.1 List of Upgrade Measures

For all entries in the `statusquolist`, the existing speed band is examined for major negative deviations from the average speed, since these areas are the most promising for riding time reductions. For every speed restriction, an estimation of upgrade costs and the benefit in riding time is to be calculated. Some speed restrictions will feature several upgrade possibilities, which all are to be included in the list of upgrade measures. For a judgement of infrastructure upgrade costs and benefits, refer to Walter and Fellendorf 2015: 40, Veit, Walter, et al. 2014, 104ff. and Veit, Fellendorf, et al. 2016: 53ff.

Apart from strictly local problem solving as described, some measures will not affect the (local) infrastructure, but the whole edge, or network. These include vehicle or operational measures, such as a change in the signalling system, electrification, modified vehicle properties, and others. Such measures cannot deploy their benefit locally, but rather across the whole edge (or network). The benefit, however, must be quantified sectionally, so as to offset it with other (local) infrastructure measures.

Note that some measures will be dependent upon others, such as consecutive speed increases or measures that need a different signalling system by force. The selection of measures must, upon selection, respect these *downwards* dependencies. However, some

[50] Note that the alteration of station locations for demand reasons (i. e. moving stations closer to settlements) is *not* in scope here. If necessary, this is to be evaluated in the Service Intention Phase already!

measures, such as a modified signalling system, might yield a different measure set on other parts of the edge if selected (i. e. render solutions possible that would not have been chosen individually). Therefore, *upward* dependencies shall be imposed, automatically selecting measures that depend upon other measures, but would not have been selected otherwise. This way, the measure bundles also reflect the technical consequences of upgrade decisions.

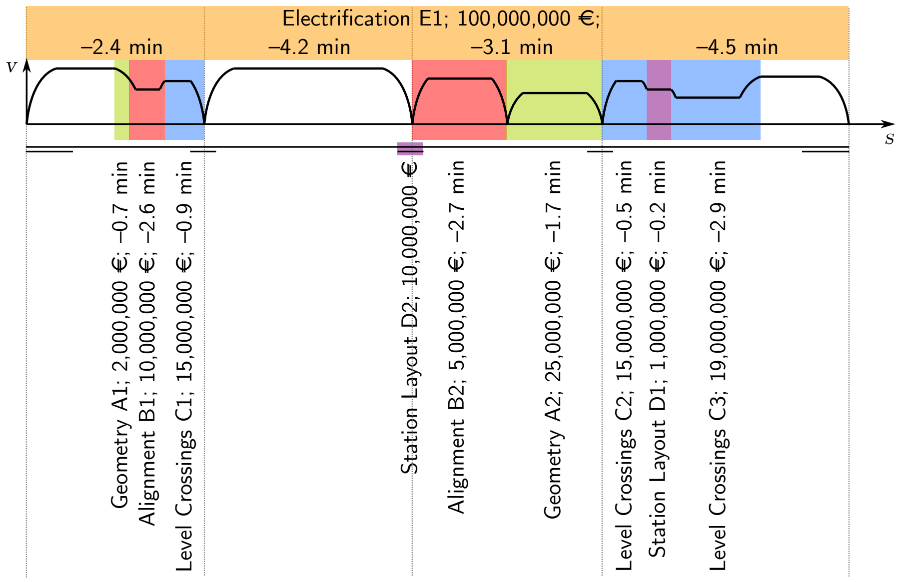


Figure 6.2: Sample speed band and quantified measures, grouped by measure type.

Figure 6.2 shows the speed band and target track layout from figure 6.1 with the corresponding measures highlighted and grouped. Note that (i) the “Electrification” measure is laid across the whole edge, but the riding time benefits are appended to the corresponding sections, and (ii) the track layout measure in the second crossing station does not yield riding time benefits, but an additional track in that section. Note also that the projects and their costs in this example are of illustrative nature. For an evaluation of both costs and riding time benefits, refer to Walter and Fellendorf 2015: 40.

6.2.2 Timetable Construction

We start off with the best ranked target timetable from the Target Timetable Phase (see section 5.6). In order to translate a target timetable to a timetable used for actual timetable construction, some transformations need to be accomplished: First, the final hubs (i) per train system and (ii) per edge need to be constructed.

From the target timetable, at first only the hub type at $\nu_i = 1$, i. e. at the base interval T, is known. This hub type is to be rolled out according to ν_i per train system, so as to retrieve distinct departure and arrival events per hub, per edge, and per train system. This information can be directly deducted from the compacted `hublist` table, so a translation from the numeric hubtype as described in section 5.4.1 back to the actual hubtype, and subsequently departure and arrival times, can be carried out without further effort.

However, for all entries that remained in the `statusquolist` table, i. e. for all entries that have not been moved to `statusquolist_noaction` (or `statusquo_unattainable`) tables, a standard timetabling procedure will *not* yield a satisfactory, let alone a feasible, timetable. Therefore, a feasible timetable needs to be created *iteratively* until the timetable can feature the departure and arrival events, in both directions, of the target timetable.

This iterative approach means (i) using a standard timetable construction routine[51] and (ii) subsequently changing infrastructure elements until (at first) a target riding time becomes attainable.

6.2.3 Sectional Target Riding Times

We shall split target riding times per edge and per train system further to retrieve *sectional target riding times*. This shall denote riding times between points of operational importance. For single-track lines, these are points of crossing and overtaking; for multi-tracked lines, these are the points of overtaking, as long as there are different train types in place on the edge. If not, the section length equals the edge length. Note that the calculation of sectional target riding times incorporates list entries from both `statusquolist` and `statusquolist_noaction`.

The sectioning of edges is, again, done on the time axis rather than the space axis. This way, the problem with simplified trajectories as described in section 5.5.3 can be overcome by adjusting the infrastructure following the sectional target riding times. Since the amount of necessary riding time reduction is known, the possible combinations of upgrade measures to achieve the target riding time can be identified.

On single-tracked lines, the next step is to retrieve the point in time for crossings. Within a train system, this is at the axes of symmetry. Between train systems (of different speeds), the point in time is calculated as shown in figure 6.3a. The offset time between trains in opposite directions t_{offset} is, if either train is connected to a hub, a multiple of the denser interval; otherwise it can be computed from the different hub types. Since the relation between the riding speeds is inversely proportional to the subdivision of t_{offset}, the first crossing $t_{\text{crossing},1}$ and the subsequent crossings $t_{\text{crossing},2}$ can be calculated as

[51] by using standard software such as FBS, VIRIATO, and the like

$$t_{\text{crossing},1} = \frac{t_{\text{offset}} \cdot v_2}{v_1 + v_2} \tag{6.3}$$

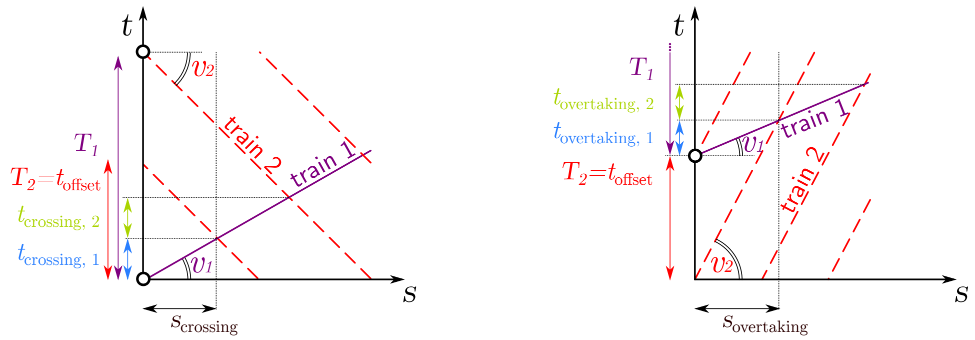
$$t_{\text{crossing},2} = \frac{T_2 \cdot v_2}{v_1 + v_2} \tag{6.4}$$

For all other edges with more than one train type, likewise, points of overtaking can be computed as

$$t_{\text{overtaking},1} = \frac{v_2 \cdot t_{\text{offset}}}{v_1 - v_2} \tag{6.5}$$

$$t_{\text{overtaking},1} = \frac{v_2 \cdot T_2}{v_1 - v_2} \tag{6.6}$$

Figures 6.3a and 6.3b show the respective geometrical relations.



(a) crossing (b) overtaking

Figure 6.3: Geometric relations for section target riding times

With the information about sectional target riding times, the selection of upgrade measures can be simplified significantly: Since the points are defined by their points in time rather than their topological position, the infrastructure upgrade measures can be grouped in such a way that existing train stations are reached for crossing and overtaking. The location of stations is therefore considered fixed and the riding times are considered variable. The riding time between the stations, determined in such a way, is considered variable. This way, the judgement of upgrade feasibility is made considerably easier, since crossing and overtaking events can be moved to existing stations rather than the open track.

Note that the sectional target riding times are to be computed for every train system on an edge.

6.2.4 Treatment of Unattainable Riding Times

The `statusquolist_unattainable` has not yet received attention in this phase. This list comprises all target riding times for train systems that are *far too low* or *far too high* for the direct incorporation into the feasible timetable.

First, all entries in the list with a target riding time that is *far too low* need to be investigated in detail. Just as described for sectional target riding times, all possible upgrade measures are to be listed, in case a riding time is attainable irrespective of the preliminary filtering.

Then, all other trajectories can be treated just as described in section 5.4.6 (including the dropping of a hub service and trajectory cancellation). This information can then be translated into a modified timetable that can be treated just like train trajectories with attainable target riding times.

6.2.5 Upgrade Measures Selection

From the sectional target riding times and the list of upgrade measures, we construct an initial solution for a measure set. Selecting just as many measures as necessary to fulfil the sectional riding time requirements, we can create *upgrade measure groups* that, each by itself, allow for an adequate riding time. However, each bundle has different total costs, rendering possible a ranking according to investment costs.

Figure 6.4 shows the measure list from figure 6.2, updated with sectional target riding times, status quo, and measure bundles. We consider the line to be single-tracked and the sectional target riding times to be measured between points of crossing.

As can be seen, the sectional target riding times call for considerable upgrade measures in every section. Measure bundle 1 incorporates measures to be taken without electrification, which is included in measure bundle 2.

In this example, an electrification makes for a total riding time reduction of 14.9 minutes. In the last three sections, with the electrification alone, it is possible to achieve the sectional target riding times, overshooting the necessary riding time reductions. In the first section, another measure accounting for at least 0.4 minutes is necessary. As depicted, the sum of measures comprising electrification is more costly than other upgrade measures with similar effect.

After the selection of measures for each train system, the measures are to be aligned to form one measure bundle per edge.

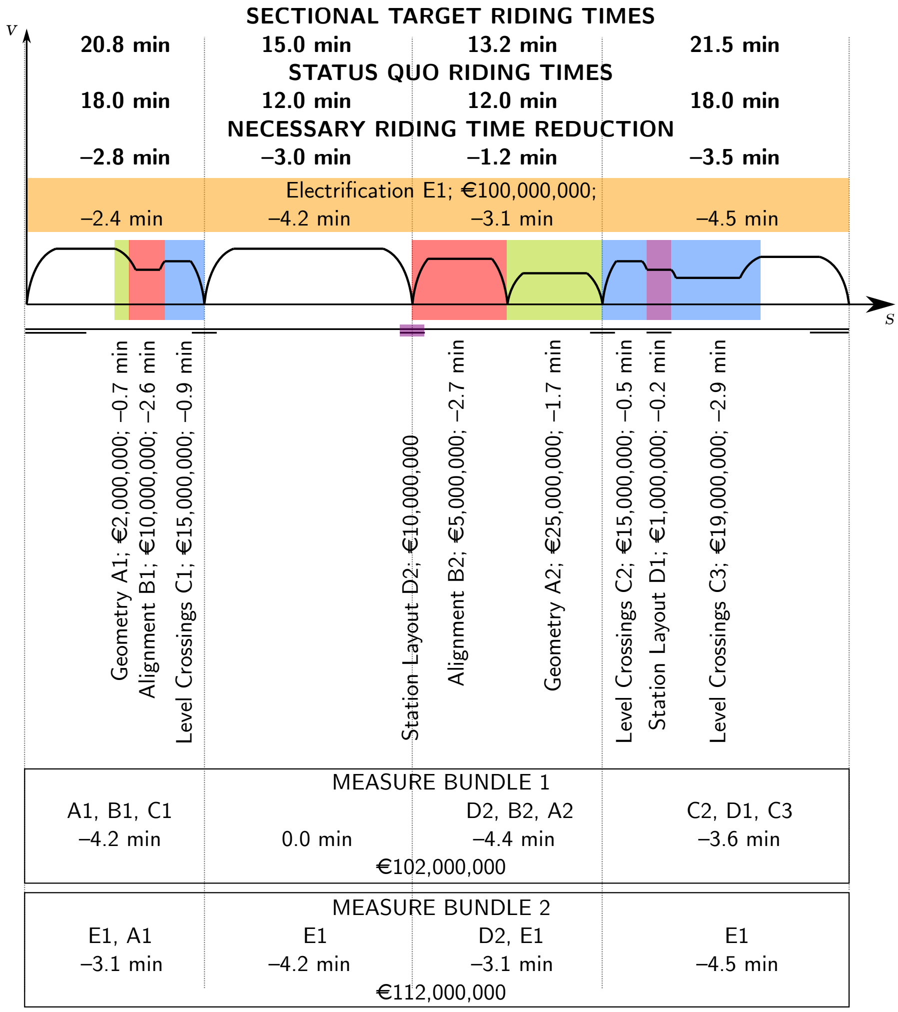


Figure 6.4: Sample speed band and quantified measures, grouped by measure type.

6.3 Iterative Upgrade Concept Creation

When both the target infrastructure layout and the sectional target riding times per train system are achieved, a target timetable becomes feasible, since all parallel and sequential performance requirements can be met[52]. However, it is likely that certain parameters will *not* be met, either due to principally insufficient upgrade possibilities or to factual impossibilities of measure implementation. In this case, an iteration between timetable, infrastructure, and demand needs to be carried out.

6.3.1 Timetable and Infrastructure Iteration

Within the example in figure 6.4, we can investigate measure bundle 1 further[53]: As can be seen, there is no infrastructure upgrade measure besides electrification in the second section. Therefore, the measure bundle without electrification must compensate for the lack of measures. In the list of riding time reductions made possible by measure bundle 1, we can spot that the sum of riding time reductions is 12.2 minutes, thus overshooting the necessary 10.5 minutes required in total from the target riding time. However, the gains are unevenly spread across the line, such that the crossings, as designed, will not be met.

Figure 6.5 shows the model timetable for this edge. The crossing point calculation yields a crossing between the faster train system and the slower train system at minutes .18 and .42 in the first and the third station, while the crossing within the train systems happens at minutes .30 (slower train system and faster train system) and .15/.45 (faster train system only), respectively.

The dashed lines mark the planned trajectories if all sectional target riding times are met. However, in measure bundle 1 there is no upgrade possibility in the second section. The riding time reduction is, therefore, to be obtained in the neighbouring sections. Since the measures in both the first and the third section allow an overshoot above the target riding time reduction, this is possible in the big picture. The crossings, however, will not work out in this timetable. The thicker, continuous lines depict the modified trajectories due to the shifted upgrade measures. As can be seen, the crossing is moved inwards into the second section. Therefore, additional double-tracked sections are necessary to allow for shifted crossings[54]. Therefore, two additional upgrade measures are necessary to allow for double-tracking a longer stretch. If these two projects amount to more than

[52]Note that an adequate signalling system and adequate rules of operation are to be presupposed at this stage; if not, they are to be included in the list of measures.

[53]Note that this example covers one train system only for easier understanding. Covering several train systems, however, works the same way, but with the aligned measure bundles treated jointly.

[54]Note that, in addition to the shifted point of crossing, the length of the crossing also increases considerably, since the crossing happens between moving trains rather than during a stop. For details on calculating crossings in movement, refer to Pachl 2011: 197f.

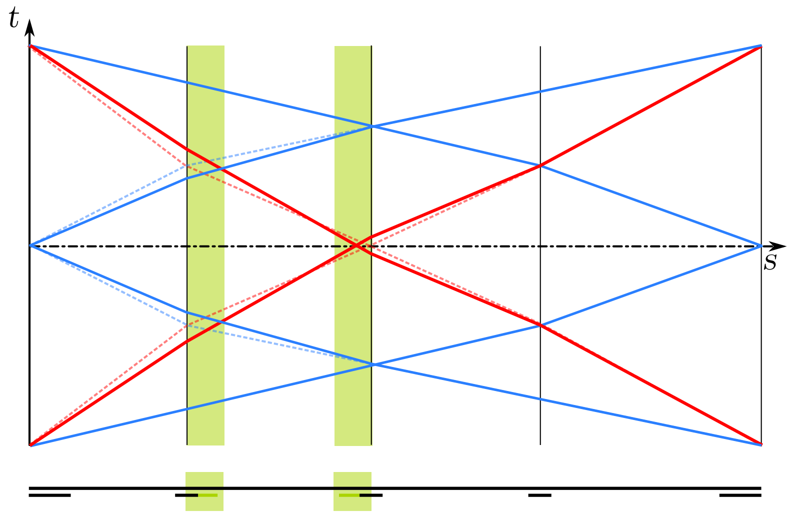


Figure 6.5: Timetable adjustment following inadequate riding times

€5,000,000 each, the investment costs of measure bundle 1 rise above those of measure bundle 2, i. e. an electrification becomes the favourable option.

There might be situations where parts of a track layout turn out to be entirely impossible. In this case, parallel performance requirements have to be changed into sequential ones. This means that trains must leave and enter stations sequentially and overtakings cannot take place where planned. Since the target riding time needs to be held in any case, this implies a further riding time decrease for the train system held back at a station.

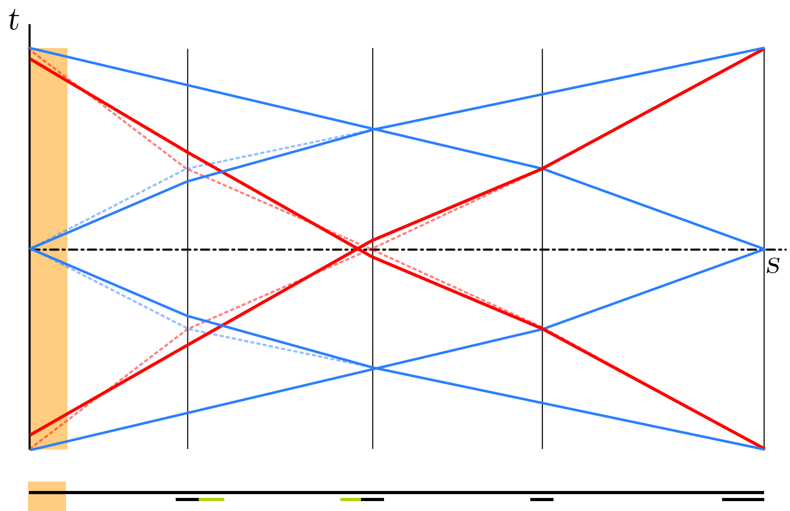


Figure 6.6: Timetable adjustment due to gridiron conflicts

Figure 6.6 shows the situation of figure 6.5 again, but implies that a double-tracking of the first station is impossible. Therefore, the slower train system is held back and leaves the station at signal headway after the faster train system. This essentially lowers the target riding time in this section again, such that either the crossing is shifted further into the section or additional riding time reductions are necessary in the first section.

6.3.2 Demand Investigation

In the best case, the setup of target track layout, sectional target riding times, and a possible iteration yields a valid feasible timetable. This *Feasible Timetable*, as it is derived from a rough demand investigation in the Service Intention Phase and prioritised along node flows in the Target Timetable Phase, is to be evaluated concerning its demand impact. Since an exact timetable is present now, a complete, simultaneous, activity and timetable based trip generation, trip distribution, and mode choice can be carried out without limitations to the coarseness of the timetable model.

At this step, the expectations from the Service Intention Phase can be verified. Within the results of demand investigation, the most interesting results are (i) the achievement of the initially stated target mobility patterns, (ii) the revised node flows (and, consequently, transfer relations), and (iii) the resulting trip loads. If all of these parameters are satisfactory, the feasible timetable can be considered as set, the measure bundles can be forwarded to detailed planning, and the Stage Development Phase can be triggered.

If not, an in-depth analysis of node flows and flow bundles is to be carried out to investigate which inadequacies of the feasible timetable lead to a reduced performance compared to the service intention. Since this analysis will vary greatly depending on the actual network, it can only be sketched here. For a more in-depth view on corresponding analysis methods, refer to Friedrich et al. 2001.

Walter used this method to assess which elements of the feasible timetable were responsible that one line featured a modal split way off the initial prognosis. A sensitivity analysis showed that the riding time between *Graz* and the *Voitsberg-Köflach* region, i. e. the connection of a central city with a regional centre, has risen so much during the construction of the feasible timetable that the reaction was a significant ridership decrease. A further speed-up of the constructed line trajectories would have been possible by skipping so many stops that, again, no positive effect in the modal split could be achieved. Finally, it was decided to introduce extra express trains that were not part of the service intention, so as to both satisfy the required service level along intermediate stops and attract new passengers from the regional centre. This way, the target modal split could almost be reached by a narrow margin(Walter 2016: 85f.).

6.3.3 Iteration between Infrastructure, Timetable, and Demand

There are three situations calling for a further iteration between infrastructure, timetable, and demand:

1. when the target track layout cannot be reached by upgrade measures;
2. when the sectional target riding times or the target riding time turn out to be unattainable;
3. when the demand investigation produces unexpectedly low values for target mobility patterns.

Though all of these timetable infeasibilities stem from different sources, the result is similar in all cases: Target parameters in timetable and demand are not met. Just like in section 6.3.1, the demand model must also be incorporated into the iterative design process.

For the case when infrastructure upgrade measures turn out to be impossible, first a feasible timetable as close as possible to the target timetable is constructed per edge and then evaluated concerning lost transfers in the node flows. This timetable will automatically yield lost transfers that can be evaluated and treated just as described in section 5.4.6.

When a predetermination of transfer redesign has been achieved, a new run of complete demand modeling can start. This way, the process is iterated until a target state, as described in section 6.3.2, is reached.

6.4 Change of Target Timetable Version

Since the Target Timetable Phase yielded a *set* of target timetables, we might as well leave the target timetable currently worked on. This is necessary when there is an accumulation of expensive or impossible measures around distinctive hubs. We can obtain a different timetable version by a proximity search in the target timetable graph. Since the edges contain the differences in hub types between the timetables, the search can be carried out for a target timetable differing in the hub type of the hub with the accumulation of inadequacies as described. This way, a close but different timetable is obtained, such that the difference in measure evaluation and demand investigation remains manageable. Upon selection of a new target timetable, the iteration can start again as described.

6.5 Target Infrastructure and Final Timetable for Stage Development Phase

As already noted in section 6.3.2, the goal of the Feasible Timetable Phase is the satisfaction of (i) the target mobility pattern, (ii) a feasible timetable model, (iii) the predicted demand structure. The set of iterations between infrastructure measure design, timetable construction, and demand investigation is stopped upon reaching these target parameters. By then, there is a comprehensive list of

1. train trajectories, including stopping policy, arrival and departure events, crossings, and overtakings;
2. infrastructure, vehicle and operational upgrade measures, including rough planning, mutual dependencies, and investment cost calculations; and
3. transfer relations, cross-section and vehicle loads, timetable-based node flows, modal split, trip lengths, and key mode choice parameters.

This information can then be passed to the Stage Development Phase for further processing.

7 Practical Application

The principles described in the preceding sections shall finally be applied to a practical example. The *Ostregion* around *Wien* has been chosen as a model application, since it features (i) a network of many cycles, (ii) a heterogeneous interval structure, and (iii) dense suburban railways with an inner-city trunk line, (iv) remote regional railways and (v) a vast long-distance railway network.

Note that this practical example comprises a *model demand*, rather than the demand actually present in this area, for data availability reasons. Furthermore, statements concerning measure feasibility have been evaluated in coarse granularity only. The applications given here are therefore to be viewed as proof-of-concept, rather than real-life project application.

For real-life applications with similar methodology, refer to Veit, Walter, et al. 2014, Walter 2016, and Veit, Fellendorf, et al. 2016.

7.1 Preprocessing

7.1.1 Hub Structure

Figure 7.1 shows the project area with the nodes and edges numbered. Minor lines without circular relations to the rest of the network have been left out of scope, since their attachment to the network timetable is of no importance for the solution of the network. Furthermore, the transfer hubs are located in such a way that they model transfers better than current line termini[55]

As presented in section 4.5, we require, from the Service Intention Phase, (i) the line network, (ii) the service intention, and (iii) the node flows.

7.1.2 Service Intention

The service intention presented here differs in some points from both the current status and the currently valid official target timetable in order to allow for a more distinctive modeling of the key features presented in this work.

[55]See the station of *Obersdorf*, located at the branching of the line to *Bad Pirawarth*, rather than in *Wolkersdorf* one station to the north, where currently half of the S-Bahn trains end.

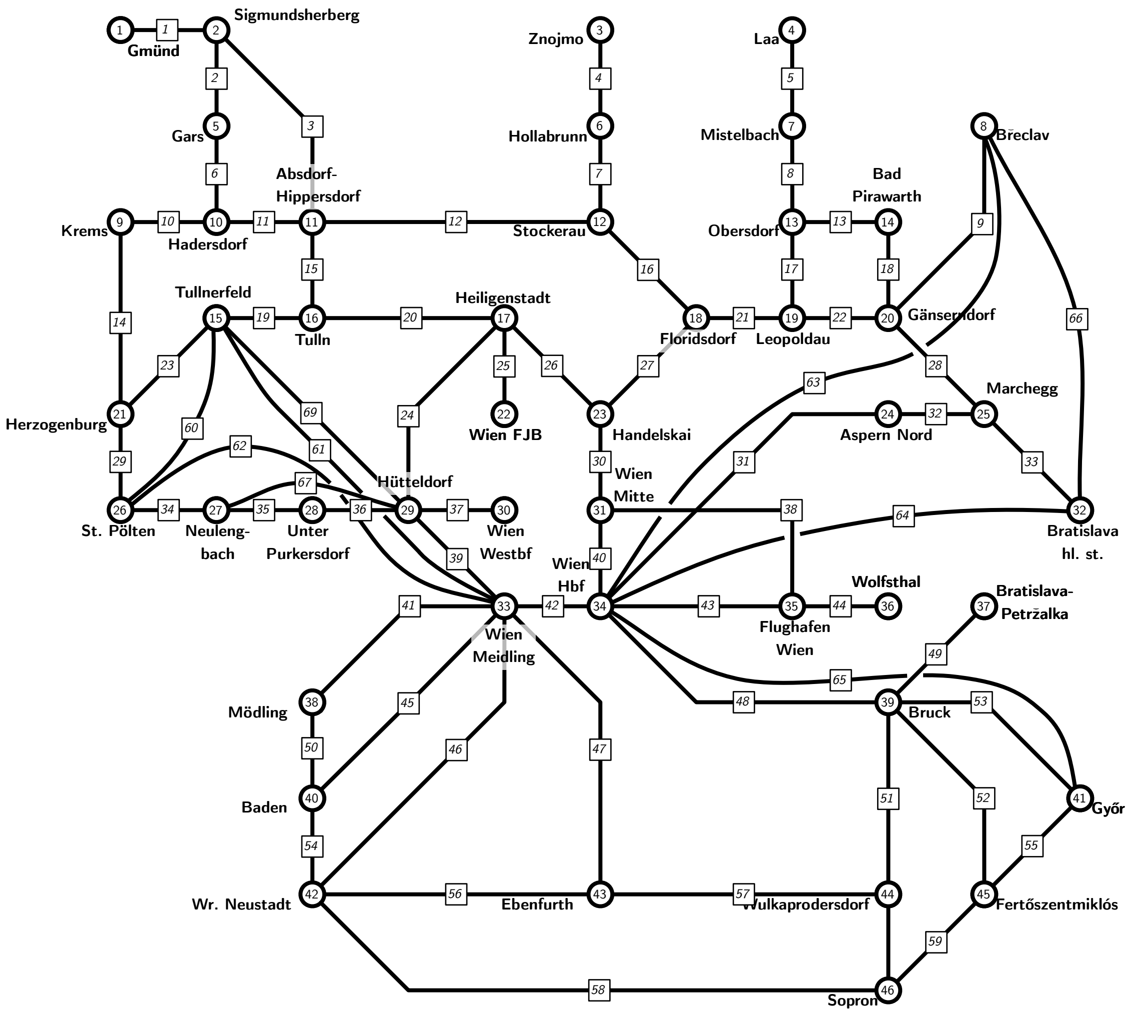


Figure 7.1: Principal nodes and edges in the *Ostregion* area.

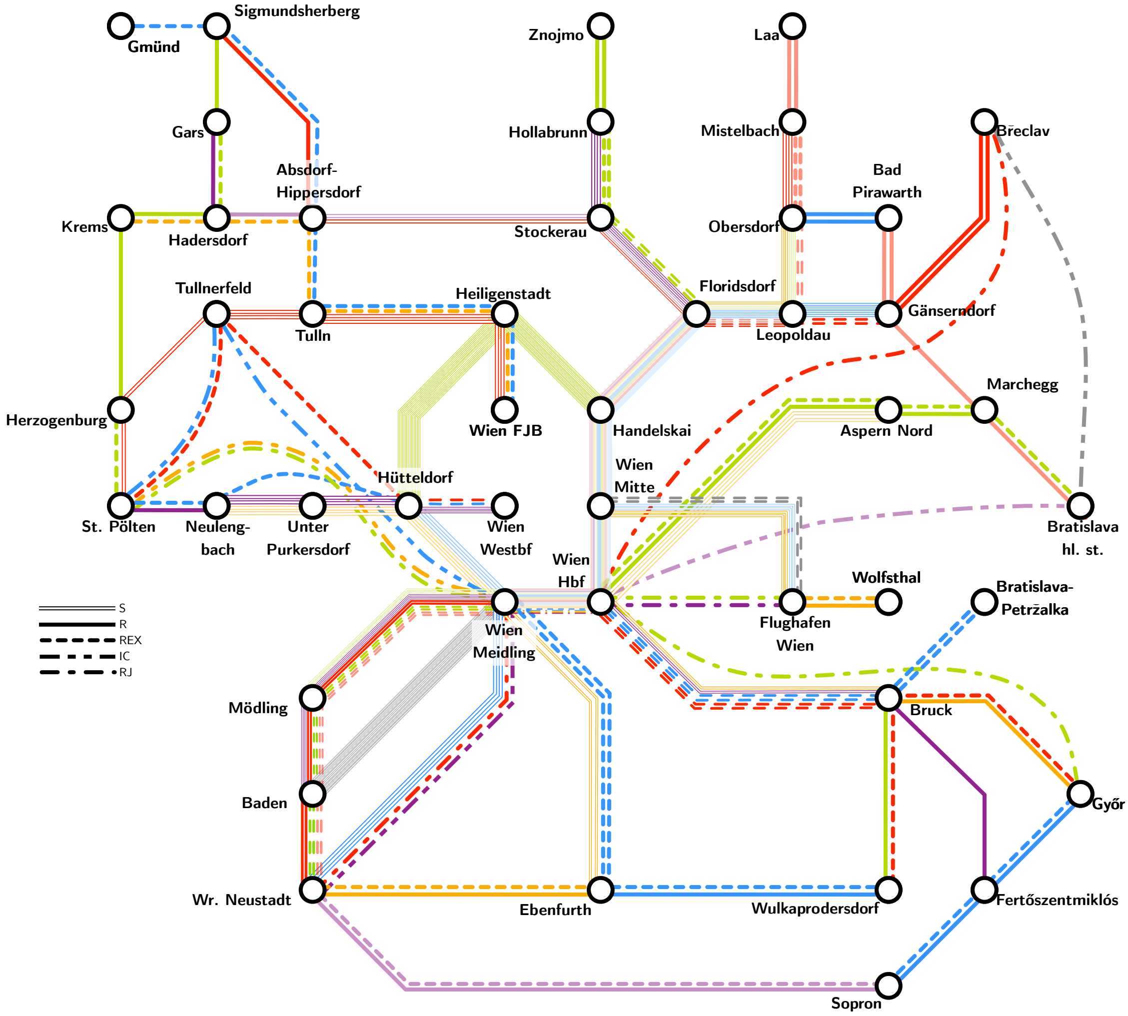


Figure 7.2: Service intention of *Ostregion* as adapted for the practical application

Figure 7.2 shows the service intention upon which the timetable has to be based. Every line in the graph corresponds to one hourly ride, so the number of lines per edge directly corresponds to the ν_i present per train system. The stations displayed are those that should serve as timetable hubs, thereby defining the node and edge structure. Consecutive edges of continuous lines are highlighted in distinct colours. As can be seen, some lines do change their train types between edges, working as zoning trains. Furthermore, some edges are modelled as bypasses, when trains do not serve intermediate hubs.

From this service intention, the lines are to be disintegrated into edges and their corresponding intervals. If we have a closer look at the edge 20 *Tulln–Heiligenstadt*, we can spot, in total, four lines: Two are the Regional-Express (REX) trains serving *Gmünd–Wien FJB* and *Krems–Wien FJB*, respectively, and two are the S-Bahn trains between *St. Pölten–Wien FJB*. Upon disintegration, these lines are, on this stretch, modelled as

```
train  edge type nu
S201   20   S    2
REX201 20   REX  1
REX202 20   REX  1
```

We also define the hubs 18 *Floridsdorf*, 23 *Handelskai*, 31 *Wien Mitte* and 34 *Wien Hbf* as termini of inner-city trunk lines, assigning the edges 27 and 30, *Floridsdorf–Handelskai–Wien Mitte* a $\nu_i = 24$ and edge 40 *Wien Mitte–Wien Hbf* a $\nu_i = 20$. Theoretically, the same could be applied to other stretches as well, but we will limit the search to the *Stammstrecke* section here. This way, the named edges (numbers 27, 30, and 40) are not considered in the normal riding time calculation process, but separately.

Furthermore, edges 24 *Hütteldorf–Heiligenstadt*, 26 *Heiligenstadt–Handelskai*, and 45 *Wien Meidling–Baden* can be excluded completely from the set of edges, since the one line each (current lines S45 and WLB) runs at an interval dense enough to skip transfer design completely (see section 5.3.1).

7.1.3 Trajectory Construction

The service intention yields a network with 46 hubs, 69 edges, 5 train types (RJ, IC, REX, R, and S), and 140 train systems.

For the trajectory construction, a survey of the possible riding speeds needs to be carried out. Figure 7.3 shows the results for all S, R, and REX trains in the *Ostregion* project area, and all IC and RJ trains in Austria.[56]

For the trajectory construction, a sensitivity analysis of found trajectories was carried out so as to find the matching `searchspace` ranges (see section 5.3.4). The trains moved

[56]Note that, as shown in section 2.2.4, the train types have been normalised, e. g. RJ trains *Wien–Graz* were categorised as IC trains, EC trains *Wien–Budapest* as RJ trains, etc.

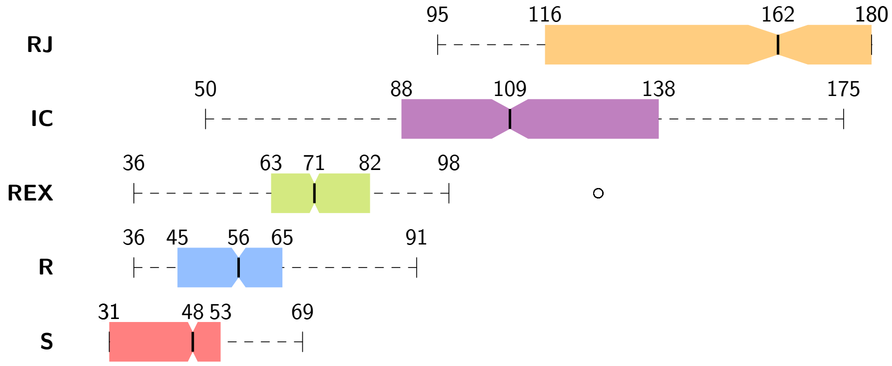


Figure 7.3: Boxplot of riding speeds per train system in the *Ostregion* project area (IC and RJ: in Austria), 2016 timetable, in km/h.

to `nonefound` are those where no trajectory could be found (see section 5.3.4). Table 7.1 shows the results for varying ranges when riding times are calculated with the values from figure 7.3. Remarkably, not even the largest set of limits, i. e. the current maximum and minimum riding speeds, allows for a satisfaction of all trajectory construction prerequisites, leaving at 9% of all the trains outside the initial calculation.

case no.	limits	number of trains in `nonefound`
1	min–max	12
2	1st–4th quintile	31
3	1st–3rd quartile	48
2	2nd–3rd quintile	80

Table 7.1: Train systems without trajectory found per riding speed limits, of 140 train systems in total

7.1.4 Network Remodelling

In order to quantify the problem arising from the last calculation step, we shall investigate the trains within `nonefound` in detail. Table 7.2 shows the train systems without found trajectory in case 1.

train system	on edge	train system	on edge	train system	on edge
REX101	10	S251	25	S371	37
REX151	15	REX251	25	R421	42
S191	19	REX291	29	REX423	42
REX211	21	REX371	37	IC421	42

Table 7.2: Train systems in `nonefound`

An analysis quickly finds that (i) for all REX systems in `nonefound`, the edges are too short to allow for a riding time calculation, since the combination of v_r and ν_i does not yield a possible solution; (ii) for the systems on edges 25 *Heiligenstadt–Wien FJB* and 37 *Hütteldorf–Wien Westbf*, the edges to the terminal stations are of no relevance, since the only connections at *Wien FJB* and *Wien Westbf* are urban services with dense intervals and thus no necessity for the construction of a timetable hub; the trains on these edges can simply be excluded from the calculation process; and (iii) for edge 42 *Wien Meidling–Wien Hbf* a hub pair is a more adequate solution than the treatment as two separate hubs.

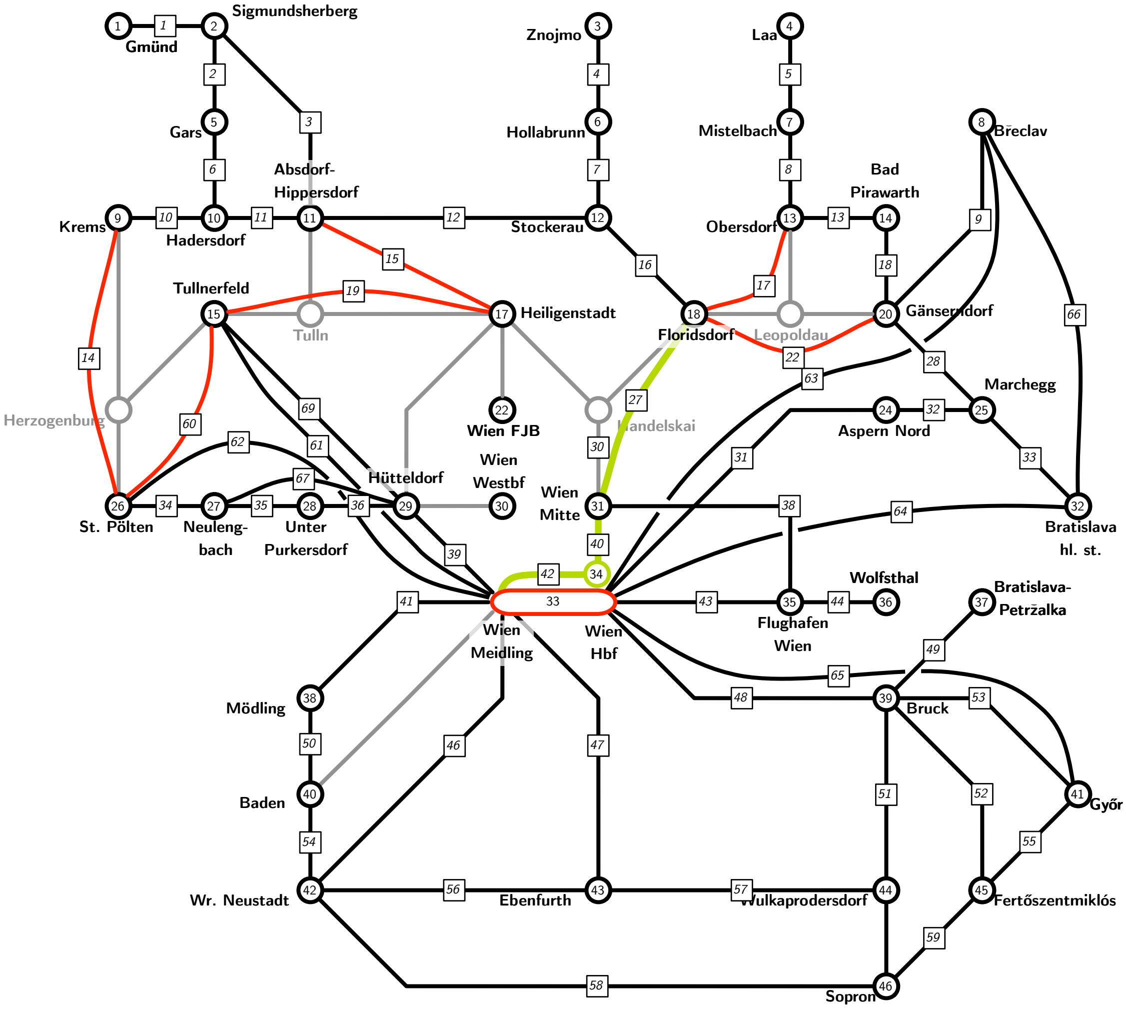


Figure 7.4: Modified network after remodelling

We can therefore remodel the network as shown in figure 7.4:

1. Edges 10 and 11, 15 and 20, 19 and 20, 21 and 22, 21 and 17, 29 and 14, and 29 and 23 will be combined to one edge each, removing hubs 14 *Herzogenburg*, 16 *Tulln*, and 19 *Leopoldau* from the network. This does *not* mean that the stations are not served; there is, however, no possibility of implicitly modelling a complete transfer hub there, affecting connections *Krems–Herzogenburg–Tullnerfeld*, *Absdorf-H.–Tulln–Tullnerfeld*, and *Obersdorf–Leopoldau–Gänserndorf*[57]. These connections might, however, be possible with slight tweaks to the timetable upon feasible timetable construction.

2. Edges 25 *Heiligenstadt–Wien FJB* and 37 *Hütteldorf–Wien Westbf* are removed from the network. Rides on these edges will be simply attached to the incoming trajectories, but without the need to serve *Wien FJB* and *Wien Westbf* as transfer hubs.

3. Edges 24 and 26 *Hütteldorf–Heiligenstadt–Handelskai* and 45 *Wien Meidling–Baden* are removed from the network as explained.

4. Edges 27, 30, and 40 *Floridsdorf–Handelskai–Wien Mitte–Wien Hbf* are moved to the inner-city trunk line processing. This means that hub 23 *Handelskai* can be removed, too, and edges 27 and 30 joined to one. Also, hubs 18 *Floridsdorf*, and 31 *Wien Mitte*, will feature a completely different set of hub types for incoming trains. Also, the hub 34 *Wien Hbf* has to be reinstalled as special hub just for the treatment of the inner-city trunk line, attaching it only then to the hub pair 33 *Wien Meidling–Wien Hbf*.

7.2 Target Timetable

7.2.1 Trajectory Construction

After the initialisation of the (modified) network, the trajectory construction can be started. The hub type classification, at first, yields 5 hub types: A, B, $C_{7.5}$, $C_{22.5}$, and D. Without a hub type transformation, there is a `hublist` comprising 1,180 entries with $\nu_i = \{1, 2, 4\}$ for the edges treated with the classical approach, and 192 entries with $\nu_i = \{20, 24\}$ for the inner-city trunk line. After transformation, the 5 hub types remain in place, but the 1,180 entries in the `hublist` expand to 1,756 entries. As already noted in section 5.4.3, the 192 entries on the inner-city trunk line remain unchanged.

[57] In other words: since the only train systems that *could* serve these stations as transfer hubs are the S-Bahn trains, jeopardising the intention of creating a transfer hub.

7.2.2 Trajectory Matching

The completed and converted `hublist` can be used directly within the HSDAG-wrapped FastDiag algorithm (see section 5.4.5). The truth formula obtained consists of 21,218 literals. The complete hub type conflict detection algorithm for the *Ostregion* takes 89 min to run.

The hub type conflict detection yields 34 different minimal conflict sets, sized 7 to 13 elements. As noted in section 5.4.5 already, each of these conflict sets refers to one target timetable. Due to the nature of embedding the FastDiag algorithm in an HSDAG envelope, some of the conflict sets are identical in both the conflict set and the SAT result, i. e. the hub type assignments. This reduces the number of target timetables to 23. Furthermore, timetables 17 and 20 yield identical hub type assignment, though with different conflict sets.

rank	timetable no.	n_{cs}	$\sum q_{\text{tr, mod, lost}}$
1	17	11	33,120
2	11	11	34,400
3	10	9	38,920
4	12	11	40,600
5	13	10	41,600
⋮	⋮	⋮	⋮
18	31	7	69,440
⋮	⋮	⋮	⋮

(a) weighted with $\sum q_{\text{tr, mod, lost}}$

rank	timetable no.	n_{cs}	$\sum q_{\text{tr, lost}}$
1	21	8	136,000
2	27	8	149,000
3	22	11	151,000
4	10	9	184,000
5	17	11	195,000
⋮	⋮	⋮	⋮
18	31	7	272,000
⋮	⋮	⋮	⋮

(b) weighted with $\sum q_{\text{tr, lost}}$

Table 7.3: Weighted ranking of target timetables, best five timetables

As described in section 5.6, we can rate the timetables by their benefit or, if not available, by the amount of lost passenger transfers. The former requires a complete demand calculation for each target timetable, which has not been carried out in this example. Therefore, the number of passengers as retrieved in the Service Intention Phase is used as a weight $\sum q_{\text{tr, lost}}$. Furthermore, we can use parameter k_{sr} as introduced in section2.3.5 to obtain $\sum q_{\text{tr, mod, lost}}$ and thus a weighted ranking per train type.

Table 7.3 shows the five best-ranked target timetables, ranked both with $\sum q_{\text{tr, mod, lost}}$ and $\sum q_{\text{tr, lost}}$. Interestingly, the actual *minimal conflict set*, i. e. timetable 31, with only 7 trains in the set, ranks 18 out of 23 for this given demand. Since no complete demand investigation has been carried out, it is, at first, sufficient to stick to a weighting with $\sum q_{\text{tr, mod, lost}}$.

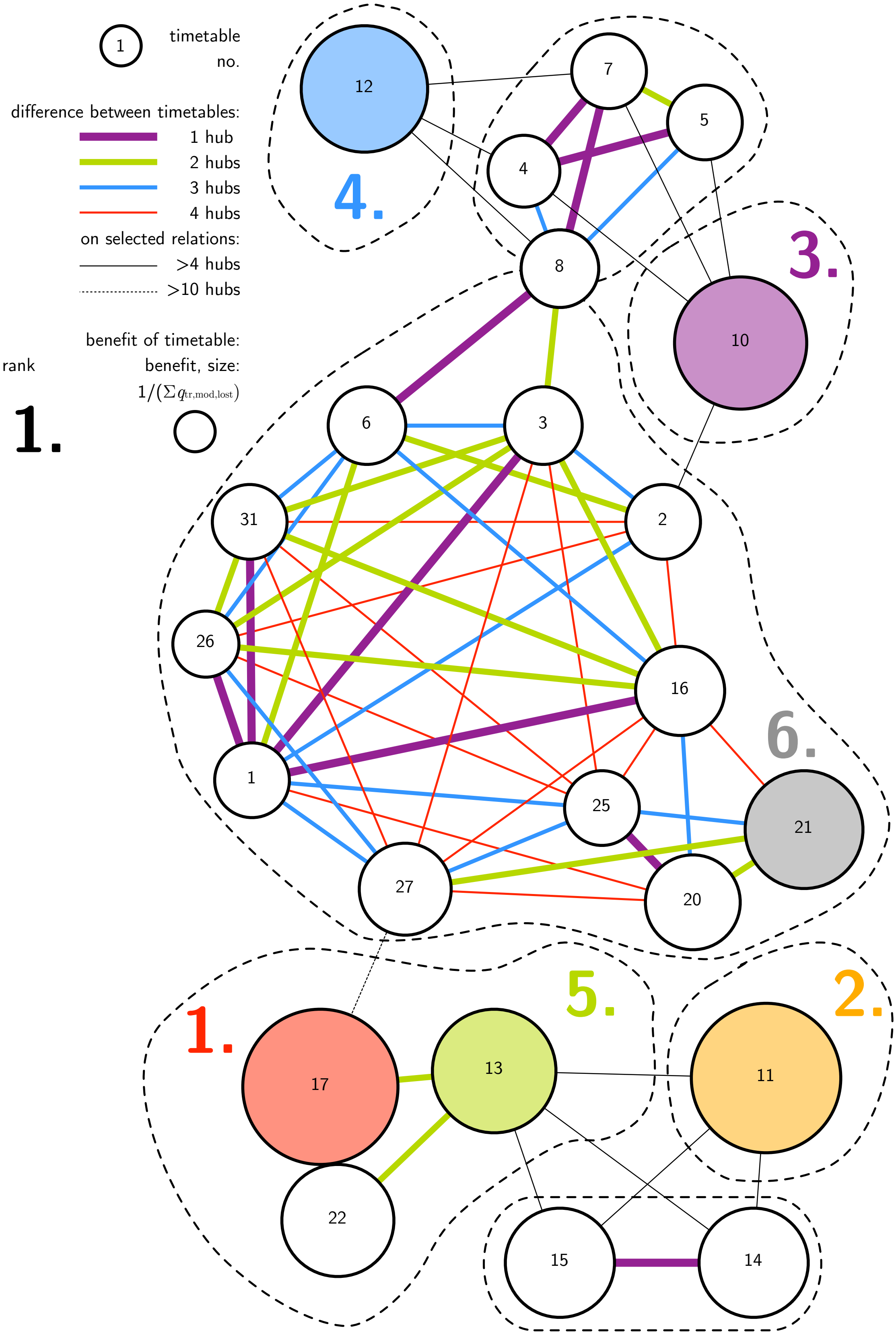


Figure 7.5: Target timetable graph, weighted with $\sum q_{\text{tr, mod, lost}}$.

7.2.3 Target Timetable Graph

Figure 7.5 shows the target timetable graph. As can be seen, the timetables can be roughly split into seven groups.

1. The group around the best-ranked timetable, no. 17, only comprises three timetables. Connections to other groups involve a difference of at least 5 hubs (timetable 13 to 14 and 15). The difference between the best and the second-best-ranked timetable, no. 11 is 11 hubs, so there is not much flexibility when moving to a different timetable model.

2. The second-best-ranked timetable, no. 11, is the most isolated in the graph. The closest neighbours differ by 8 to 9 hubs (timetables 13, 14, and 15), timetables 17 and 22 differ by 11 hubs, and all other timetables differ by at least 22 hubs.

3. The third-best-ranked timetable, no. 10, is also comparatively isolated, but differs by its closest neighbours (2, 5, and 7) by only 5 to 7 hubs.

4. The fourth-best-ranked timetable, no. 12 also stands isolated, differing at best by 8 hubs from its neighbours.

5. The central group, comprising 11 timetables, is densely connected, i. e. the timetables differ 1 to 4 hub types only; however, there is no version in this group with a good ranking. Timetable 21 ranks best at sixth place in this ranking—when ranked without k_{sr}, however, it ranks first.

6. Timetables 4, 5, 7, and 8 are attached to the central group via timetable 8 only, so they form one more group.

7. Finally, timetables 14 and 15 form one last, comparatively isolated, group.

However, the timetables are, in total, comparatively homogenous amongst each other: The largest difference between two target timetables is 25 hubs and the median difference is 14 hubs. In other words, typically 32 hubs are identical between two target timetables. At first sight, it might seem strange that so many hubs are identical in all target timetables. However, this is plausible when viewed within the network: Since the interval on the edges around the involved hubs is long, there is not much flexibility when it comes to serving these hubs differently.

From a demand point of view, the priority of which timetable to pick first is clear, provided we stick to the use of $\sum q_{\text{tr, mod, lost}}$. However, when we consider the requirements to the Railway Infrastructure Design Process as summarised in section 4.5, it is better to start off with a timetable in the central group, so as to maintain the flexibility of changing between timetable versions.

For the sake of better demonstration of this method, we pick timetable 21 as a starting point. This timetable is closely linked to four other timetables, so there are alternatives to move to with reasonable effort. Figure 7.6 shows the assignment of hub types to the modified *Ostregion* network. The hub type conflict set obtained is

```
REX591  REX601  IC601 REX671  REX501  REX502  REX541  REX542
```

The edges with train systems in the hub type conflict set are highlighted in grey shading.

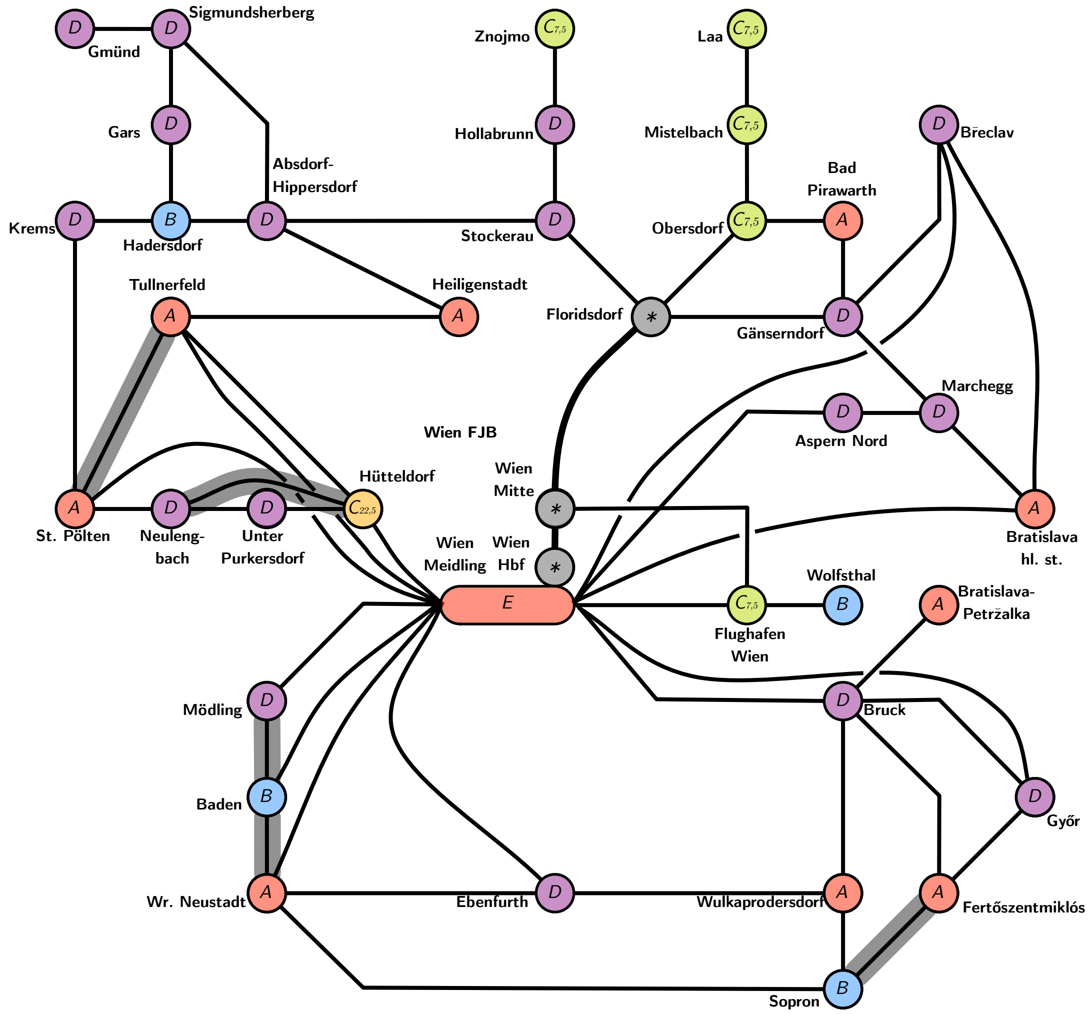


Figure 7.6: Hub structure of timetable 21 with conflicting edges highlighted

As can be seen, the vast majority of the network is constructed with semi hubs (type *D*). This is not surprising, since (i) semi hubs at interval $T = 60\,\text{min}$ denote full hubs of type *B* for interval $T_i = 30\,\text{min}$ and full hubs of type *A* for interval $t_i = 15\,\text{min}$, (ii) the network is too dense to allow for full hubs with hourly intervals only. The important hubs at *St. Pölten, Wr. Neustadt*, and both *Bratislava* stations are of type *A*, and the hub in *Sopron* is of type B. Furthermore, the hub pair at *Wien Hbf/Meidling* is of type E, i. e. also a full hub.

Finally, the hubs of the inner-city trunk line are denoted with starred hub type $(*)$ to denote that they, by definition, consist of ν_i/n_{T_r} different instances of type C, rather than one type only.

7.2.4 Conflict Resolution

We shall investigate the implications of this timetable for the riding times. We focus on the *St. Pölten–Tullnerfeld–Hütteldorf/Wien* triangle, the stretch *Wr. Neustadt–Baden–Mödling–Wien*, and the hub in *Floridsdorf*.

St. Pölten–Tullnerfeld–Hütteldorf/Wien

As can already be seen in figure 7.6, the edges *St. Pölten–Tullnerfeld* and *Neulengbach–Hütteldorf* feature hub type conflicts. Figure 7.7 shows the network and timetable graph of this area.

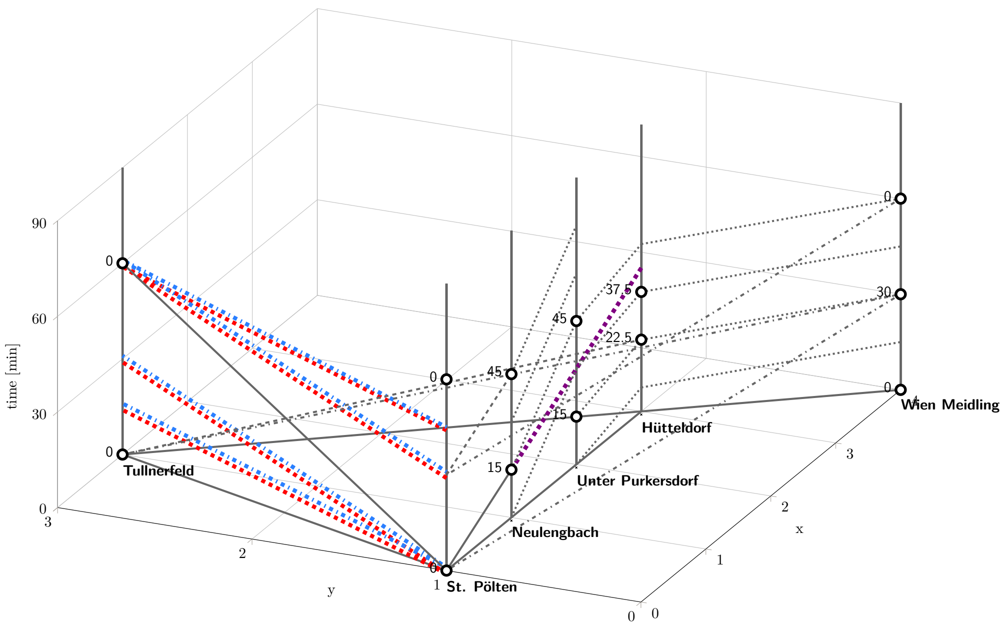


Figure 7.7: Hub structure and conflict sets in the *St. Pölten–Tullnerfeld–Hütteldorf/Wien* triangle

The first group of train systems in question are IC601 and REX601 on the edge *St. Pölten–Tullnerfeld*. As can be seen, IC601 und REX601 both can offer edge riding times of 15 or 30 minutes; none of these possibilities allow for a hub of type A in both *St. Pölten* and *Tullnerfeld*.

This conflict can offer two solutions:

1. Arranging IC601 and REX601 to part each other's intervals, which turns both *St. Pölten* and *Tullnerfeld* into a full hub at at 30-minute interval. This means that none of the adjacent edges need to be touched. However, transfers between the train system shifted by 30 minutes and other train systems at *Tullnerfeld* and *St. Pölten* are rendered impossible.

2. Dropping the hub service at *Tullnerfeld* and joining the trajectories *St. Pölten–Tullnerfeld–Hütteldorf(–Wien Westbf)* for REX601/REX691 and *St. Pölten–Tullnerfeld–Wien Meidling* for IC601/IC611. This means we have to look up alternative trajectories on the adjacent edges. From the `hublist`, we can fetch this information and find

nu	train	hub_id_a	hub_type_a	t_a	hub_id_b	hub_type_b	t_b	profile
1	REX691	29	A	0.0	15	C_7.5	22.5	1
1	IC611	15	A	0.0	33	B	30.0	1

   This means that a trajectory with $t_r = 7.5\,\text{min}$ was found for *Tullnerfeld–Hütteldorf* and one with $t_r = 15\,\text{min}$ for *Tullnerfeld–Wien Meidling*. This way, the total riding time *St. Pölten–Wien* amounts to 30 min instead of the original 45 to 60 min, which amounts to the status quo of 30 min.

   In this case, transfers at *Tullnerfeld* are rendered impossible for *both* train systems, missing the hub there by 15 min.

For either possibility, the dropping of the transfer at *Tullnerfeld* is the best possible solution, since *all* timetables found feature *Heiligenstadt* as type A. From the model node flows in *Tullnerfeld* (see figure 7.8) , we can deduct a strong flow from *Herzogenburg* to *Hütteldorf* (transfer R291–REX691) and vice versa, while the flow from *St. Pölten* to *Heiligenstadt* (transfer IC601/REX601–S191) is minor. Therefore, a hub structure serving *Tullnerfeld* with type D_1, i. e. 15 min *after* the full hub, allows for a transfer in the stronger relation with only 15 min transfer time, while the less important relation face a 45 min transfer time.

The second conflict, incorporating REX671, can be solved easily: Since there are four S-Bahn trains per hour serving *Hütteldorf*, dropping the connection to hub type $C_{22.5}$ at *Hütteldorf* yields no drawback at all: Train REX671 continues to *Wien Westbf*, so there is no drawback in missing train REX691 directed there. All other directional transfers in this relation can be served by the dense S-Bahn lines.

Figure 7.9 shows the trajectories modified according to possibility 2.

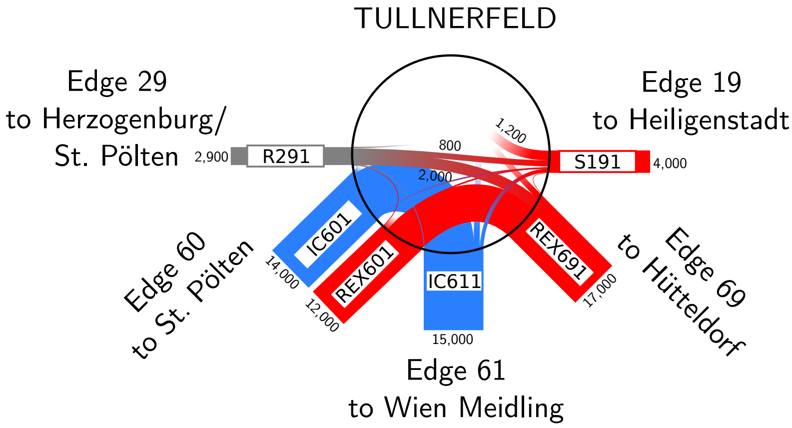


Figure 7.8: Node flows for *Tullnerfeld*

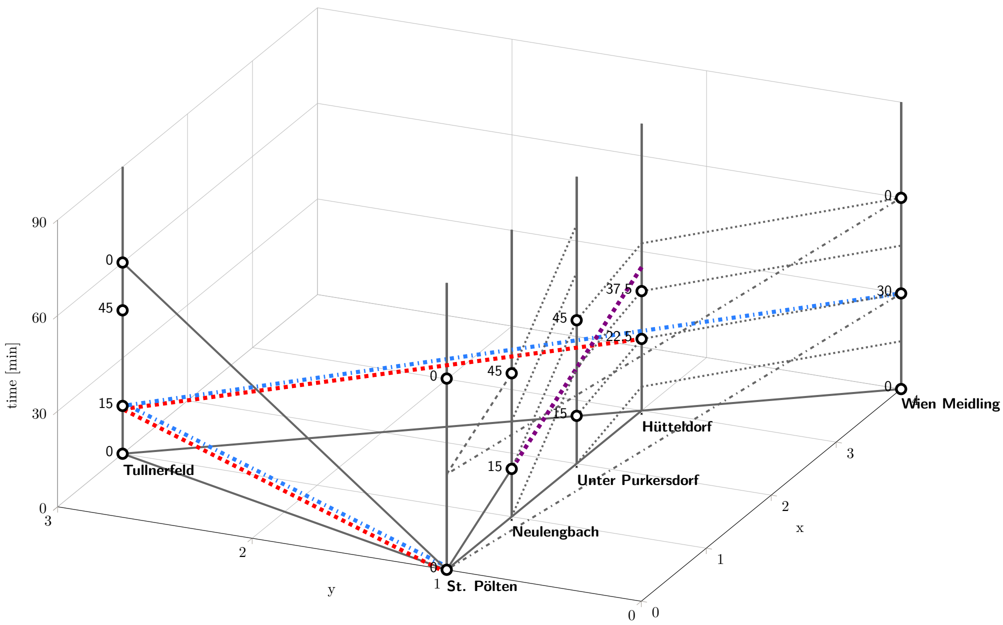


Figure 7.9: Modified trajectories for IC601 and REX601 on the line *St. Pölten–Tullnerfeld–Hütteldorf/Wien*

Wr. Neustadt–Wien Meidling

The second area touched by the conflict sets is on the *Südbahn* line. It incorporates REX501 REX502 REX541, and REX542.

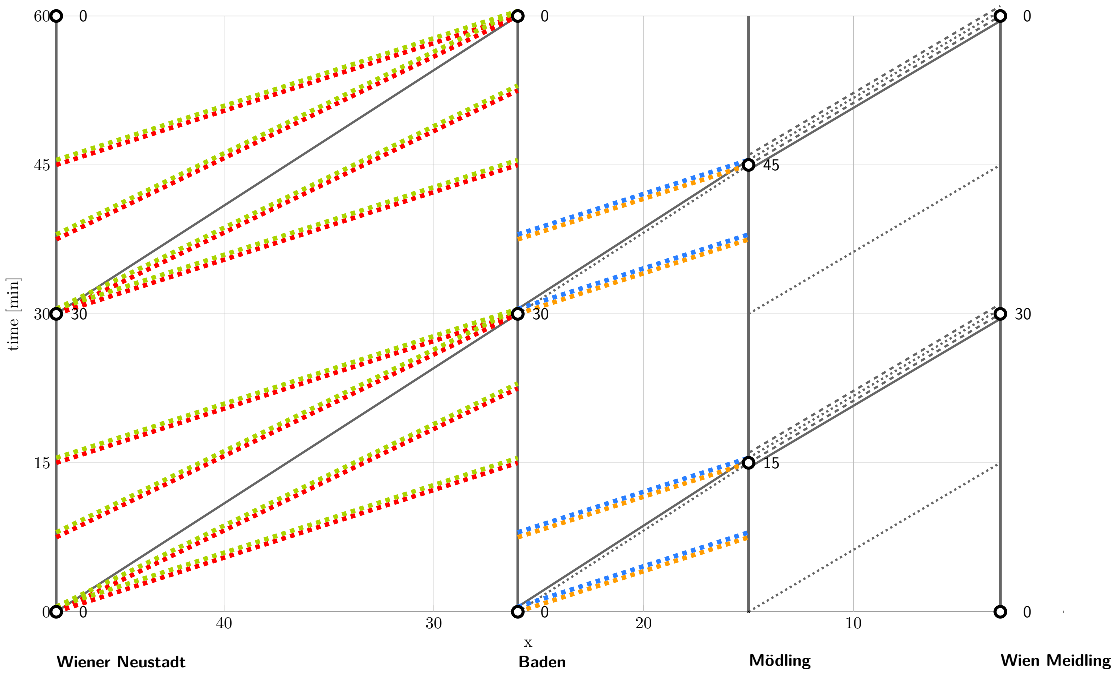


Figure 7.10: Hub structure and conflict sets on the *Südbahn* line

Figure 7.10 depicts the initial situation on the *Südbahn* line. If we recall the network from figure 7.2, we can state that neither *Baden* nor *Mödling* serve as important transfer hubs within the railway network. Both are, however, important transfer hubs to regional bus lines, of which most run at intervals of $T_i = 60\,\text{min}$. Taking the values obtained in the `hublist`, train sytems REX541 and REX542 on the *Wr. Neustadt–Baden* edge as well as REX501 and REX502 on the *Baden–Mödling* edge are *too fast* to allow for a hub service at all three hubs. A quick fix for this problem is to also obtain alternative trajectories for the *Mödling–Wien Meidling* edge. There, we can find

nu	train	hub_id_a	hub_type_a	t_a	hub_id_b	hub_type_b	t_b	profile
1	REX411	38	C_22.5	22.5	33	B	30.0	5

i. e. a trajectory considerably faster than the ones selected within the trajectory matching process. Adding together the faster trajectories of the trains in question *plus* the faster trajectory found, we obtain a continuous trajectory travelling *Wr. Neustadt–Wien Meidling* in 30 min. This is exactly what the current situation is like for long-distance trains[58]. However, we can already foresee that this will not allow for many stops, so we can go for a different approach.

Bringing all trajectories found on the set of edges together, we can construct a *zoning train* system on this line. This way, (i) no overtakings occur, (ii) a strict 15-minute headway can be offered along the line, and (iii) selectively served hubs can be offered in *Wien Hbf*, *Baden*, and *Mödling*, allowing for the installation of full timetable hubs for the hourly bus departures in the latter two. Figure 7.11 shows this concept.

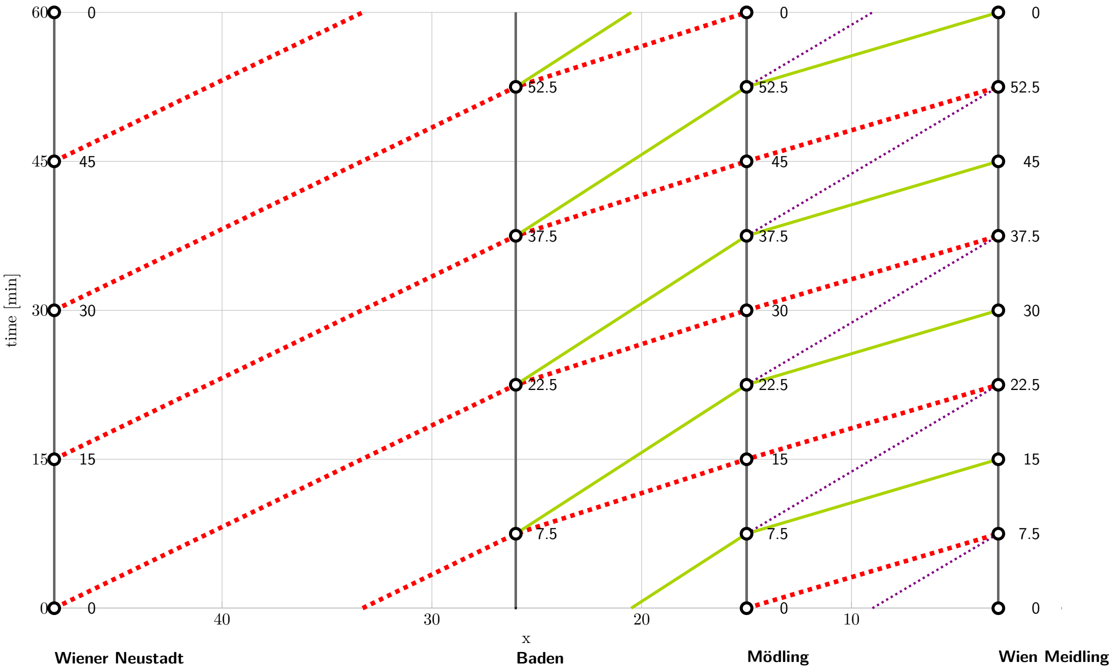


Figure 7.11: Zoning train system on the *Südbahn* line

Floridsdorf Hub

The last section of the network we shall investigate in detail is the hub structure around *Floridsdorf*, the northern end of the *Stammstrecke* inner-city trunk line. We can recognise three branches emerging from the trunk line to the north: one to *Stockerau*, one to *Obersdorf*, and one to *Gänserndorf*[59]. On the trunk line, $\nu_i = 24$, i. e. a headway of 2.5 min, has been passed over from the Service Intention Phase. From the outer branches, $\sum \nu_i = 18$, i. e. 6 slots per hour will start southwards from *Floridsdorf*.

[58] Note that we consider the *Pottendorfer Linie* link to be in operation already, denoted as edge 46 in figure 7.1. Therefore, we do not consider long-distance trains on the conventional *Südbahn* line.

[59] Note that the latter two jointly serve the stretch to *Leopoldau*, which has been removed from the network upon remodelling.

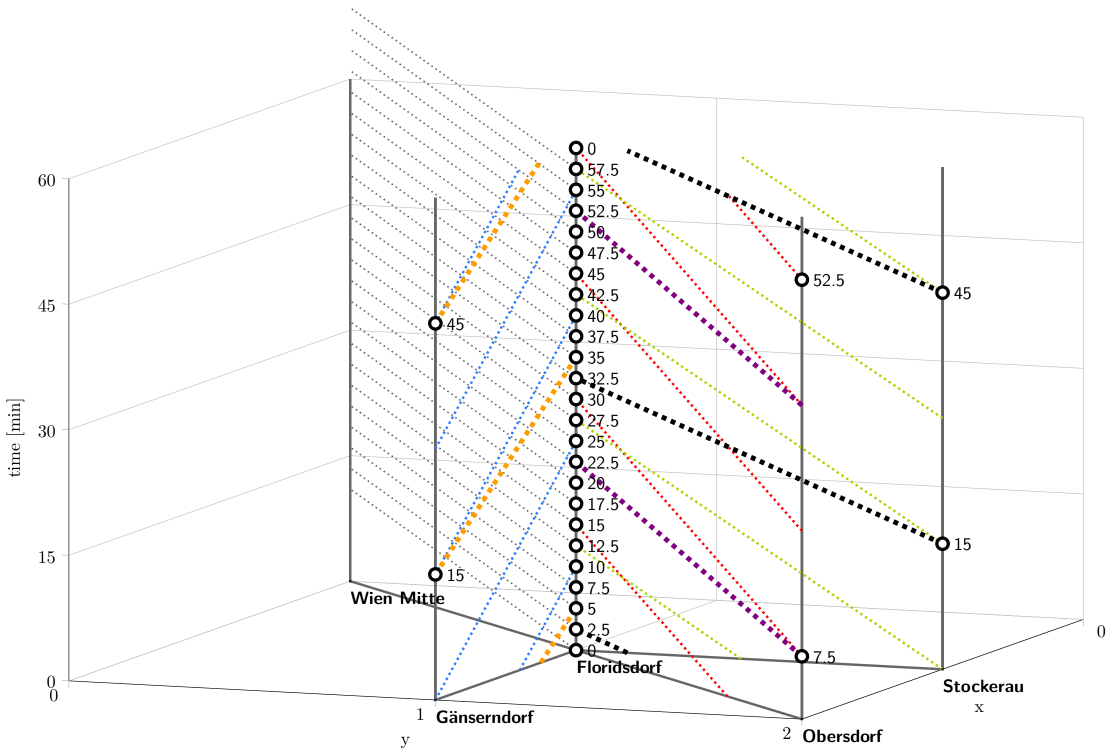


Figure 7.12: The inner-city trunk line hub at *Floridsdorf*

Figure 7.12 shows the trajectories found in the trajectory matching process. Even though the flexibility is high due to the large number of slots designed to be left unserved, there is only one solution satisfying all constraints. If we take a closer look, this becomes obvious: All three branches feature riding time ranges of 15 to 22.5 min for REX trains and 22.5 to 45 min for S-Bahn trains[60]. Also, all three branches have the same number of train systems: two S-Bahn systems and one REX system. Adding to this, both *Stockerau* and *Gänserndorf* are type D hubs. Therefore, we can expect a trajectory clustering around minutes .00, .15, .30, and .45. Indeed, we can see in the figure that the free slots are distributed quite unevenly at minutes .07.5, .17.5, .20, .37.5, .47.5, and .50. Also, it requires both the S-Bahn and the REX trains on the edges to *Obersdorf* and *Gänserndorf* to be sped up slightly compared to today; slowing them down is not even a possibility, since the next free slot would be 15 minutes later!

7.2.5 Preliminary Infrastructure Dimensioning

For this step, we pick edge 31 *Wien Hbf–Aspern Nord* for a detailed investigation. Apart from the train systems on the very edge, we need to account for five other edges that share a common stretch: 63 *Wien Hbf–Břeclav*, 64 *Wien Hbf–Bratislava hl. st.*, 43 *Wien Hbf–Wien Flughafen*, 65 *Wien Hbf–Győr*, and 48 *Wien Hbf–Bruck*.

On the edge itself, we find an S-Bahn train system every 30 min, and both an R and a REX systems at hourly intervals. We know that *Wien Hbf* is of type E and *Aspern Nord* is of type D. The timetabling routine from before yields a homogenous speed band for all three train systems, such that four S/R/REX trains per hour are equally split across the hour. Additionally, the timetable of the IC train *Wien–Bratislava*, running on the same tracks as the local trains treated in this step, serves *Wien Hbf* every full hour, as does the RJ train *Wien–Břeclav* sharing the tracks on the first 11 of 16 kilometres of the edge. Finally, the first two kilometres are also occupied by *Ostbahn* trains to *Wien Flughafen*, *Győr*, and *Bruck*, also serving two hourly long-distance trains and eight local trains per hour.

The minimum track layout, when determined with scan line (see figure 7.13), for the first two kilometres yields six tracks, then three tracks for the first two kilometres after the branching, two tracks until the line to *Břeclav* branches off, and then one track until shortly before *Aspern Nord*.

As noted in section 5.5.1, this is but a minimum track layout for a preliminary dimensioning and will be subject to change in the Feasible Timetable Phase.

[60]Note that t_{shift} as described in section 5.4.3 is *not* present in this case, since, on the trunk line, $T_i < t_{\text{tr,max}}$.

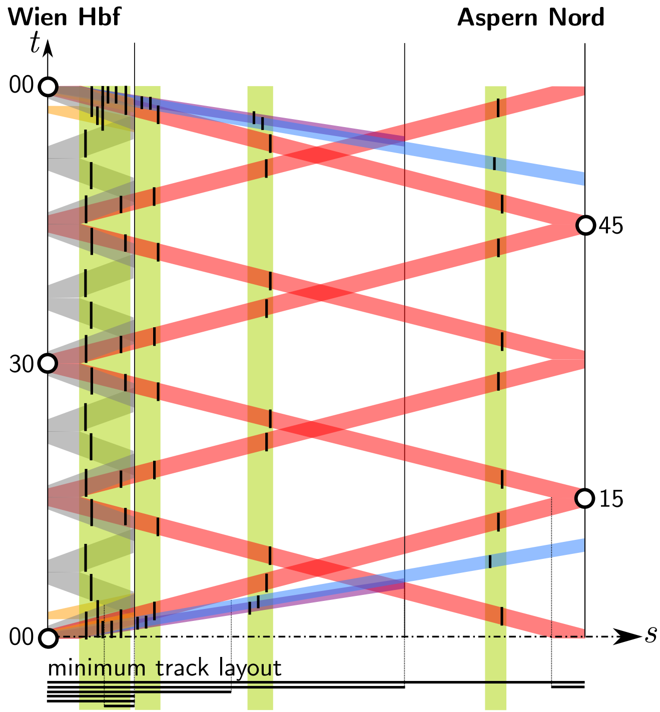


Figure 7.13: Parallel and sequential performance requirements for the *Ostbahn* line

7.3 Feasible Timetable

As described in section 6.1.1, we can compare `hublist` and `statusquolist` to obtain a list of edges where action is presumably required. Figure 7.14 shows, for each edge, whether the train systems in this edge fall into `statusquolist_noaction`, `statusquolist`, or whether they are moved to `statusquolist_unattainable` because of unattainably low target riding time or a unattractively high riding time. As already described, this offers merely a presorting, but helps in structuring the problem for further processing.

7.3.1 Status Quo

We shall pick the edges 2 and 6, *Hadersdorf–Gars–Sigmundsherberg*, for further investigation. Figure 7.15 shows the target timetable as it resulted from the Target Timetable Phase. We can spot a train R61 calling at *Hadersdorf* at minute .00 and arriving at *Gars* at minute .15. There is also a zoning train REX61/R21 calling at *Hadersdorf* at minute .30, crossing train R61 in *Gars* at minute .45 and arriving at *Sigmundsherberg* at minute .15.

We can find both train systems on edge 6 *Hadersdorf–Gars* in `statusquolist_unattainable`. Therefore, the current train trajectories have been plotted in the train graph. We can

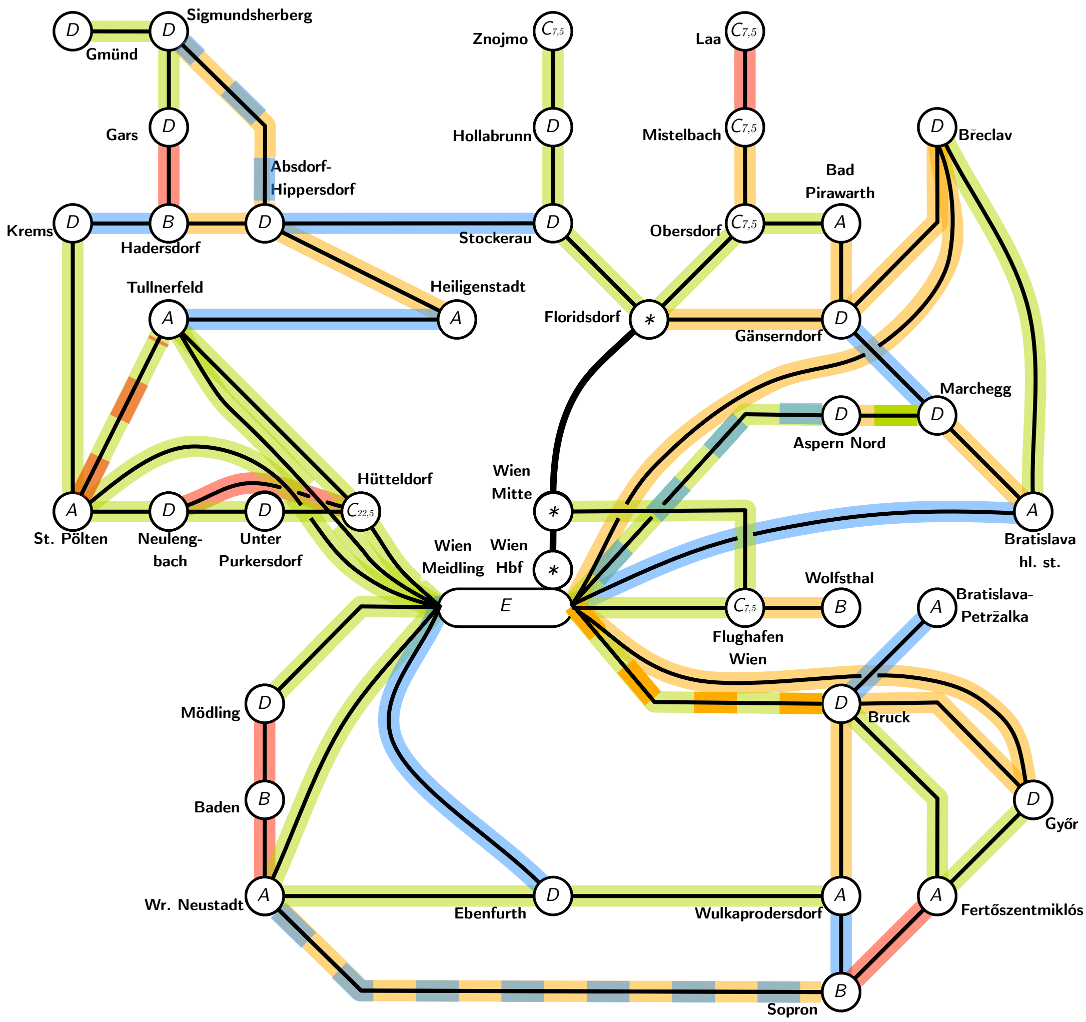


Figure 7.14: Overview of edges in statusquolist, statusquolist_noaction, and statusquolist_unattainable (too low or too high riding time).

spot that the current riding time on edge 6 *Hadersdorf–Gars* is 15 minutes longer than the target riding time, which means a reduction by 50%, thus rendering the target riding time unattainable for either train system[61]. The second edge, on the other hand, already features a riding time of 32 min, which is close to the 30 min required from the target riding time.

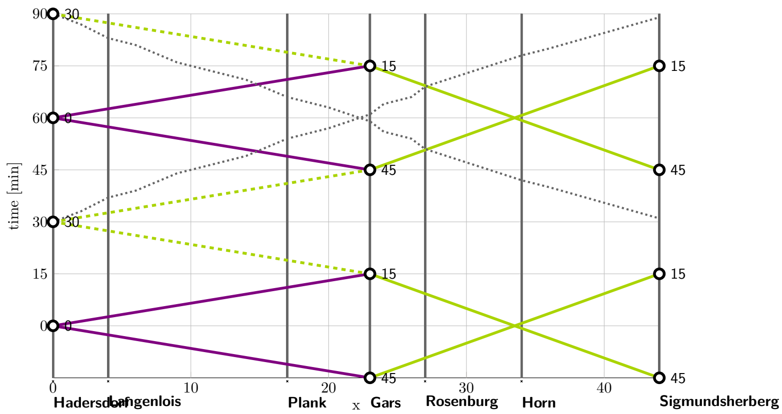


Figure 7.15: Target timetable and current riding time for *Kamptalbahn*

Therefore, the first step is to reassess the target timetable. The hubs at *Hadersdorf* and *Sigmundsherberg* cannot be altered, since they are identical in all target timetables. Therefore, the changes need to happen along the line.

7.3.2 Timetable and Infrastructure Iteration

For the further process, we need to (i) view the line as one edge rather than two and (ii) investigate the demand situation along the line. The total status quo riding time is 59 min, the target riding time is 45 or 75 min. A reduction from 59 to 45 minutes equals a reduction by 23%, i. e. can still be considered unattainable.

As noted in section 6.2.4, this is to be said without a detailed knowledge of the infrastructure condition and the safeguarding of level crossings. In fact, *Kamptalbahn* is only 43 km long, so a riding speed of 58 km/h would allow for 45 min edge riding time (which is why the trajectory construction algorithm yielded the initial solution). Taking the possible riding time reductions as evaluated by Walter and Fellendorf and assuming a design speed of 80 km/h, safeguarding or lifting 20 level crossings outside stations and skipping three stops would allow for the required riding time reduction. Figure 7.18 shows the target timetable for this timetable version.

[61] A rough estimate for skipping all intermediate stops (see Walter and Fellendorf 2015: 40) would cut the riding time to 16 to 18 minutes, but this is out of scope due to the regional service function of the *Kamptalbahn*.

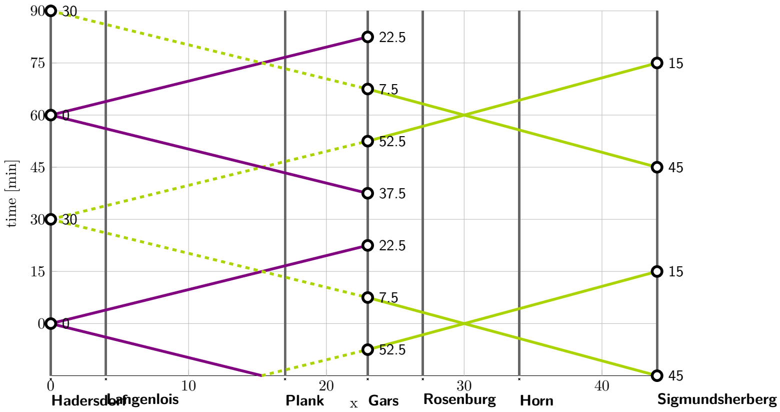


Figure 7.16: Modified target timetable for *Kamptalbahn*, versions 1 and 2

If we, on the other hand, treat the target riding time as unattainable, we must *stretch* the current riding time in order to serve the hubs on either end. When we take a look at the demand, a stretched riding time is actually a feasible solution: The number of people travelling across *Horn* is negligible compared to the number of hub-bound passengers (see figure 7.17), so the train ride might as well be split at this station. Figure 7.18 shows this timetable version. Additionally, it indicates a possibility of alleviating the broken transfer a little by connecting trains R61 and the newly created R22 line to one, albeit with significantly longer riding time between *Gars* and *Horn*.

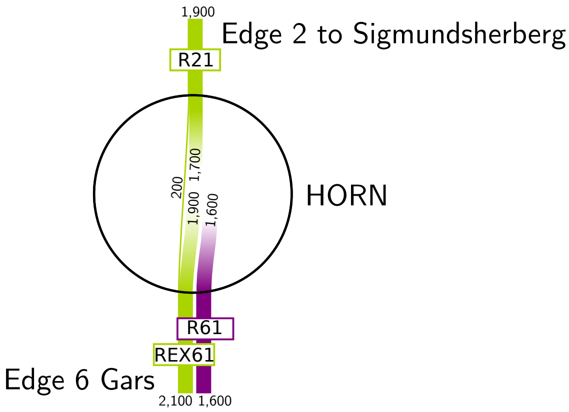


Figure 7.17: Node flows for *Horn*

With these two target timetable modifications, we can finally start the creation of a feasible timetable, aiming at the first solution, but keeping the second solution as a backup.

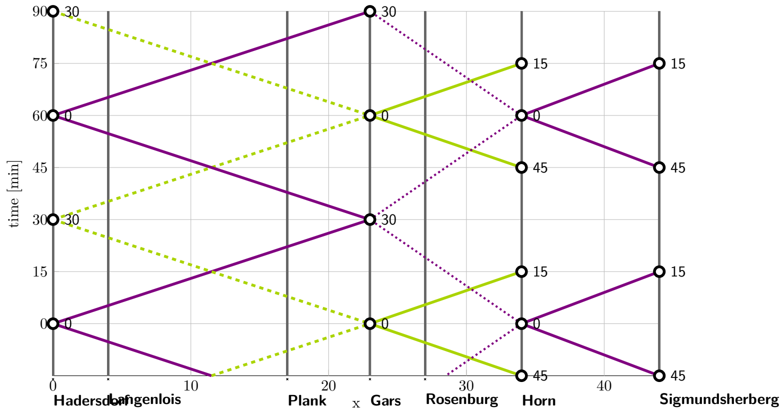


Figure 7.18: Modified target timetable for *Kamptalbahn*, version 3

Version 1

From the preliminary infrastructure dimensioning, we obtained a target track layout for the initial timetable structure. Since it has changed, we would, theoretically, need to recalculate the target track layout. However, since we are carrying out a detailed timetable evaluation at this point, this is not necessary.

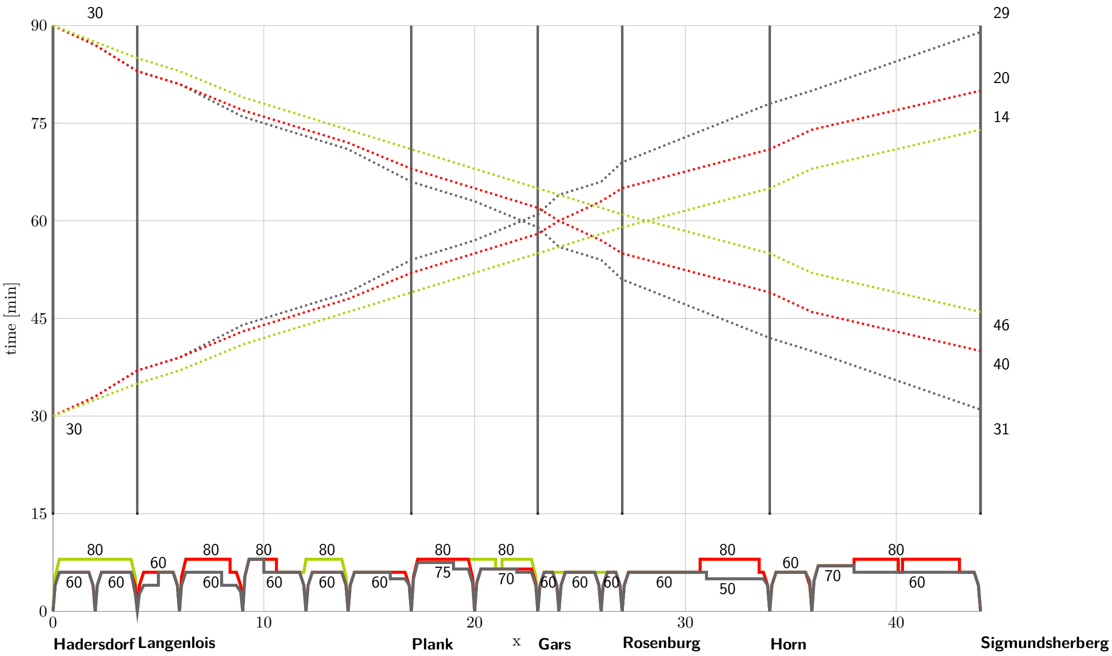


Figure 7.19: Variations of speed band and riding time for *Kamptalbahn*

At first, we evaluate the current speed band for possible improvements. Figure 7.19 shows both the current riding time and the current speed band. Furthermore, it shows the possibilities of increasing the speed between the stops, be it by safeguarding level crossings, adjusting the superelevation, or increasing uncompensated lateral acceleration (Walter and Fellendorf 2015: 40). As can be seen, using all methods can account for 9 min riding time reduction. The figure also shows both the modified riding time and the modified speed band for this case. However, these measures alone cannot account for enough riding time decrease so as to allow for the 75 min of target riding time. Finally, the last possible measure is to skip intermediate stops. More than indicated beforehand, the train needs to skip four stops in order to achieve a riding time of 74 min, which is enough to achieve the intended target riding time. Both the modified riding time and the modified speed band for this case are also shown in this figure.

For the costs of upgrading the intermediate sections, we can use the experiences documented by Veit, Walter, et al. 2014, as well as Veit, Fellendorf, et al. 2016 for regional railways. The *Kamptalbahn* features 65 level crossings on a 43 km track length. About 50 % of these can be expected to be lifted, while the others have to be safeguarded. Each safeguarding roughly amounts to €400,000, so in total we can expect €8,400,000 for these measures. All measures considering superelevation and uncompensated lateral acceleration are unaccounted for in this context, since they can be obtained during usual maintenance actions (Veit, Walter, et al. 2014, 101).

Next, we need to obtain the sectional target riding times. Figure 7.20 shows the modified timetable layout needed for finding sectional target riding times. The thin, black trajectories denote the modified trajectories as found beforehand. Train R61 could also serve the stations skipped by train REX61, since it does not have to obey the target riding time imposed by the hub at *Sigmundsherberg*. These slower trajectories are depicted in black, dotted lines.

Additionally, only those stations that come into consideration as crossing stations are highlighted: *Altenhof* and *Rosenburg*. Both are currently just single-track stops, while *Rosenburg* at least features one siding. Therefore, we will, following the values from Veit, Walter, et al., set these measures to cost €5.000.000 each (Veit, Walter, et al. 2014, 80ff. Veit, Fellendorf, et al. 2016: 53ff.).

In order to obtain valid crossings, the sectional riding times have to be met between these crossing stations. As can be seen in the black trajectories, this is not the case yet. Therefore, either a longer stretch must be double-tracked or the speed must be increased even further and one more stop must be skipped (denoted as orange sections in the speed band). A speed increase above 80 km/h requires a signalling system instead of the current direct traffic control. This measure can be estimated to cost approx. €7,000,000, spread across the whole line (Veit, Fellendorf, et al. 2016: 57). The figure shows the final trajectories following this speed increase in thick lines. As can be seen, the speed increases on the outer sections need to be compensated by extra waiting time on the central section.

Also, it can be seen that the option of serving additional stops with train R61 (dotted black lines) is to be dropped completely, since it would further increase the need for speed increases or double-track sections.

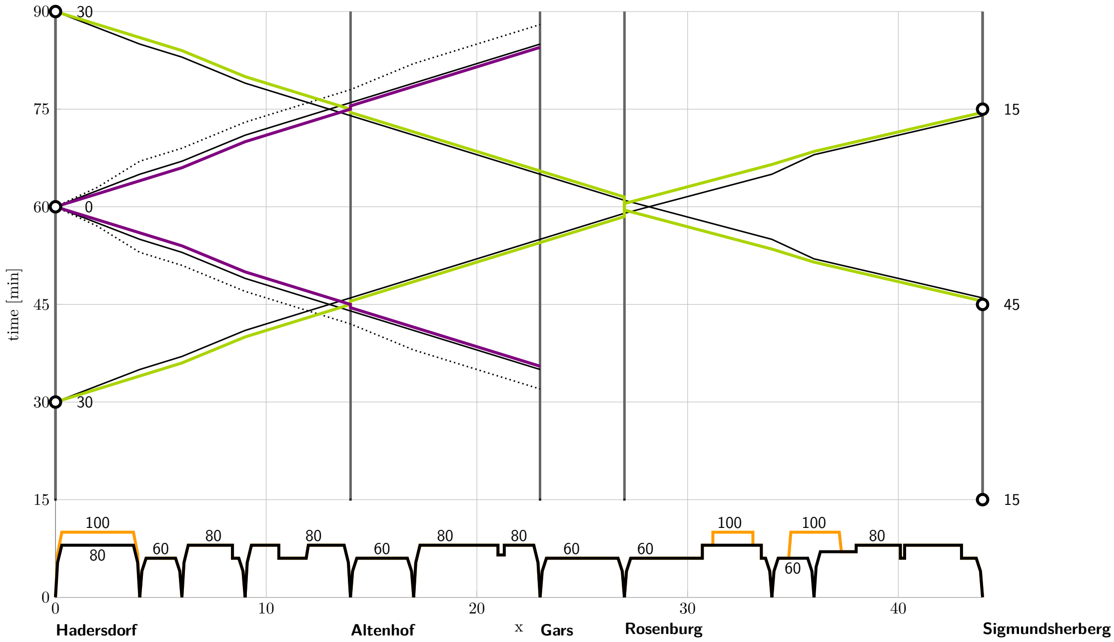


Figure 7.20: Sectional riding times for *Kamptalbahn*

Version 2

As noted before, the alternative to speeding up beyond 80 km/h is longer double-tracked sections. Since the theoretical crossing point of R61 and REX61 is at track kilometre 13 and *Altenhof* lies at kilometre 13.5, this amounts to 2 kilometres of double tracking. Since this section is comparable to the section between *Schadendorf* and *Söding* on the *Graz-Köflach* railway, we can use the values obtained there and estimate the costs at *Altenhof* to €25,000,000 (Veit, Walter, et al. 2014, 81).

At *Rosenburg*, the theoretical point of crossing is at track kilometre 28, which means that one bridge has to be rebuilt. Furthermore, this section runs along a street and the *Taffa* river, so it can be compared to the section east of *Köflach* on the *Graz-Köflach* railway, so we estimate the costs to be €40,000,000.

Version 3

Finally, we can take a look at the option of not increasing the speed and breaking the journeys in *Horn*. The crossing station in *Altenhof* must be upgraded in any case, and if the trajectories between *Gars* and *Horn* are joined at least once a day as shown in figure 7.18, the *Rosenburg* crossing must also be built[62]. Apart from that, no other measures are necessary in this timetable version[63]

7.3.3 Measure Selection

Summing up the three variants, we receive the costs as listed in table 7.4. Version 3 ranks least costly, while version 2 features considerably more costs than the other versions. From this point of view, version 3 is to be preferred.

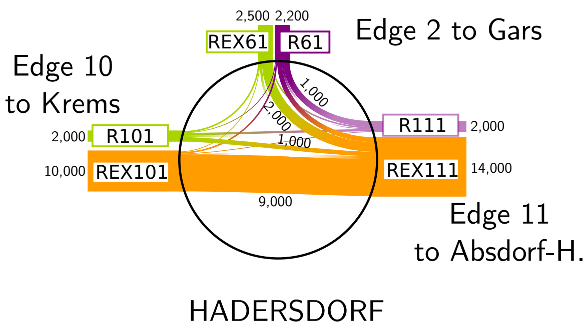
Measure	Costs		
	Version 1	Version 2	Version 3
level crossings	8,400,000		0
crossing stations	10,000,000	65,000,000	10,000,000
signalling system	7,000,000	0	0
sum	**25,400,000**	**73,400,000**	**10,000,000**

Table 7.4: Upgrade measure sets for *Kamptalbahn*

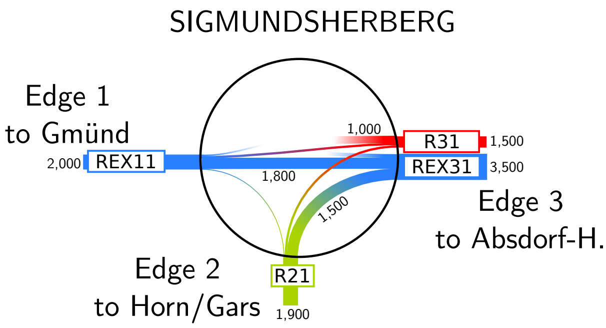
If we take a closer look at the demand, we get a clear picture of which version to choose. As noted already, we do not need to penalise the broken trajectory in *Horn*. Therefore, we can only weight the versions by their individual travel time benefit. The differences between versions 1 and 2 are too minor to be assessed. Version 1/2 and 3 differ in riding time: Passengers on the southern branch are faster by 5 minutes in versions 1/2 and by 5 minutes on the northern branch. Since we do not have dynamic demand modelling at hand in this example, we can take the current passenger flows at *Hadersdorf* and *Sigmundsherberg*, respectively, to judge. There is no need to weight the passenger numbers with k_{sr}, since we do not compare different train systems. In doing so, we can find out that 3,700 passengers on the southern branch and 1,900 passengers on the northern branch gain 5 minutes each, which amounts to 28,000 passenger minutes in total. Compared to other transfers in the hubs beyond *Hadersdorf* and *Sigmundsherberg*, amounting to significantly more passengers (see figures7.21a and 7.21b), this amount of time is unlikely to change mobility behaviour at large, so it is a feasible decision to stick to version 3 in this example, i. e. *not* to upgrade the line.

[62] Trains between *Gars* and *Horn* might also wait for the crossing with R21 at *Horn*. In this case *no* additional crossing station is needed

[63] Note that it is likely that the level crossings have to be safeguarded in any case for legal reasons (Walter 2016: 86), but this is out of scope here.



(a) Node flows for *Hadersdorf*



(b) Node flows for *Sigmundsherberg*

Figure 7.21: Node flows for comparison

8 Synthesis

8.1 Results

In this dissertation, a methodology has been developed to design a long-term strategy for infrastructure development in passenger railway networks. Based upon the objectives presented in the problem statement in section 1.3, the results and thus the contribution of this work shall be summarised.

Long-term infrastructure strategy: Starting with the Service Intention Phase, the long-term nature of this approach is set by a correspondingly long planning horizon. The Target Timetable Phase and the Feasible Timetable Phase both focus on obtaining a solution space broad enough to allow for intermediate strategy modifications without major damage.

Timetable-based infrastructure design: Every phase in this methodology comprises timetable design. However, instead of a purely timetable-based infrastructure design, an iterative approach has been found to jointly develop infrastructure and timetable.

Mixed-traffic timetable model: The incorporation of *several* intervals and *several* riding times per edge, i. e. mixed passenger traffic, is accomplished by the (i) calculation of riding times per train, (ii) the hub type conversion, and (iii) the equal inclusion of all train systems upon hub type conflict resolution.

Spot and rank infrastructure measures: In the Feasible Timetable Phase, a set of infrastructure measures per edge is developed. Within this set, bundles of infrastructure measures are developed to form functional alternatives. Since the current target timetable is fixed upon measure evaluation, every single measure can be traced back to its benefit for both the timetable and the demand. When measures prove impossible, the migration to a different target timetable can also be justified by both timetable and demand.

Provide alternatives: In both the Target Timetable Phase and the Feasible Timetable Phase, the provision of alternatives is implicitly included: The hub type conflict resolution in the Target Timetable Phase already provides a set of alternatives to relieve hub type conflicts. Target timetables differ in hub types per station and therefore each provide alternatives to each other. The creation of a feasible timetable, finally, incorporates the creation of measure bundles for each edge, so as to allow for a transparent comparison of upgrade alternatives.

Track network-wide consequences: Whenever a target timetable on an edge proves impossible, there are two possibilities: modifying the adjacent hub types and edges, and migrating to a different target timetable version. Either makes it possible to track the consequences of singular infeasibilities on an edge across the whole network.

8.2 Methodology

Since the design methodology is the key contribution of this work, it shall be summarised to give an overview. For hurried readers, figure 4.9 is placed here as figure 8.1 again to facilitate understanding.

The contribution of this work is to be found in the *Target Timetable Phase* and the *Feasible Timetable Phase*. The other Phases are sketched, but not touched upon in greater detail.

Service Intention Phase

First, a *service intention* is established. For a complete setup, we require a network of *hubs* and *edges*. On the network, we require a set of *lines* with respective *routes* and *intervals*. Furthermore, each line is to be associated with a corresponding *train system*, assigning each a *maximum* and a *minimum riding speed*.

A service intention cannot, from scratch, contain all definite determinations about line network and intervals right at the beginning. Therefore, a first iterative process featuring (i) *service intention* design, (ii) *line planning*, and (iii) *demand estimation*, must be carried out. Demand estimation can, during the Service Intention Phase, be carried out by means of a *headway-based transport assignment*. When stable results of the demand estimation are obtained, the resulting *node flows* can be retrieved.

Line network, *service intention*, and *node flows* are subsequently passed on to the Target Timetable Phase.

Target Timetable Phase

For adequate transfer conditions, the transfer hubs need to be served in the best possible manner by all train systems. Therefore, all train systems within a hub as well as all hubs within a network need to be compatible.

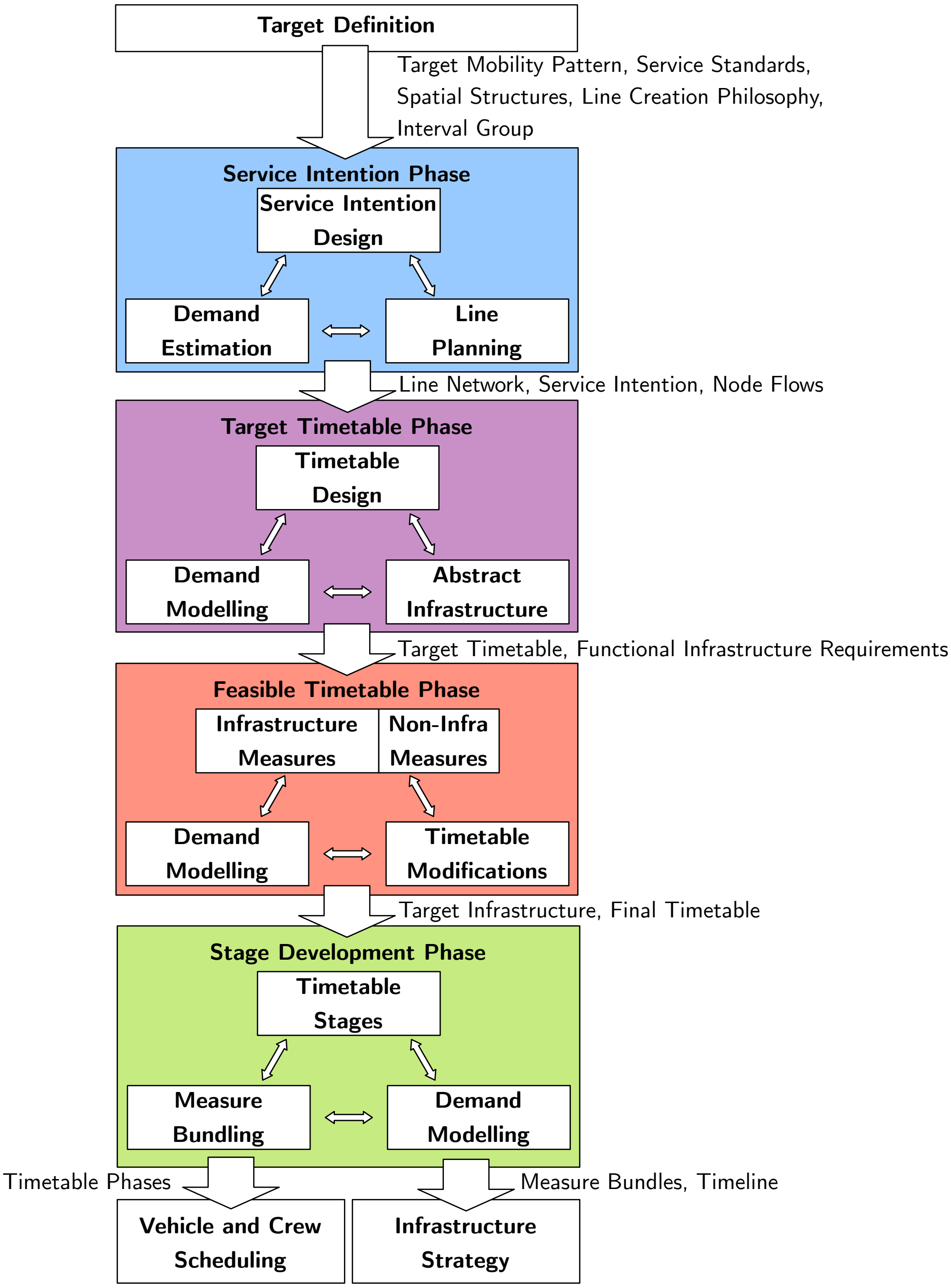


Figure 8.1: Mixed Sequential-Iterative Design Model

First, possible transfer conditions are classified into *hub types* that differ in their respective type of service within the hub. The resulting set of six hub types per occurring interval is then used as framework hinges for the setup of an array of possible target trajectories.

Next, the *minimum* and *maximum riding speeds* of each train system are used to create feasible trajectories by connecting the individual hub hinges. The resulting *hub type combinations* per interval define the possible target trajectories.

To obtain a compatibility of hub services across different intervals, the hub type combinations are then *converted* to form the final set of hub type combinations at the *base interval*.

These combinations are then used to generate a *truth formula* representing a Boolean model of the hub type combination requirements. By means of the *Boolean satisfiability problem*, the satisifiability of the network-wide hub type compatibility can be checked with reasonable computing speed.

Since it is unlikely to obtain a completely satisfiable, network-wide hub type combination right away, the next step is a *hub type conflict diagnosis* aiming to retrieve *hub type conflict sets*, i. e. the set of hub type requirements that prevent satisfiability.

The conflicts returned by the conflict diagnosis can be directly used for a *manual hub type conflict resolution*—an automated conflict resolution would not yield suitable solving strategies due to the individual nature of each conflict.

By taking the *node flows* from the Service Intention Phase, the severity of the conflicts found can be assessed. By a modification of riding speed, interval, or line service, solutions for the individual conflicts can be constructed. At this stage, it can also be decided to drop some hub transfers completely.

This yields a set of network-wide *target timetables* to be passed on to the Feasible Timetable Phase together with *functional infrastructure requirements*, i. e. target riding times, the rough location of crossing and overtaking opportunities as well as parallel station gridiron requirements.

Feasible Timetable Phase

From the Target Timetable Phase, a *predimensioned infrastructure* for each *target timetable* can be derived for each edge. However, since infrastructure upgrade options, depending upon the local situation, differ greatly from edge to edge, this phase is carried out as a *guided manual design process*.

For each edge, the *target riding times* and the *functional infrastructure requirements* are taken. Starting from the existing infrastructure, a *feasible timetable* as close to the target timetable as possible is constructed. Iteratively, *route conflicts* between target trajectories and the actually feasible timetable are removed by designing *infrastructure*, *vehicle*, *operational*, or *timetable measures*. Competing design cases are assessed in a demand model with *dynamic demand* and a *timetable-based demand assignment*, thus replacing the static demand from the Target Timetable Phase; competing options of *route conflict resolution* measures are to be assessed in terms of cost.

Iteration within this phase is continued until (i) a *feasible timetable* that meets the requirements from the Target Timetable Phase can be obtained or (ii) all efforts within the iteration fail to achieve such a feasible timetable under the given circumstances, triggering a *migration* to the next *target timetable* instance.

Finally, a set of *infrastructure*, *vehicle*, *operational*, and *timetable measures* is passed on to the Stage Development Phase.

Stage Development Phase

The set of *measures* is then taken to create a stepwise upgrade process to gradually reach a target infrastructure. Measures are bundled into *functional stages* to be completed in comparable timeframes, such that each completion of a bundle forms partial efficacy, thus avoiding stranded investments.

Each of these phases is a small-scale instance of the Feasible Timetable Phase, though with smaller planning areas and smaller-scale demand investigations. Additionally, existing re-investment requirements and medium-term infrastructure requirements need to be interwoven into the process.

Finally, the information gained in this Phase forms an *Infrastructure Upgrade Strategy*. This means that every measure design, timetable alteration, and time of investment can be justified all the way from the initial *Target Definition*.

8.3 Further Research

Within this work, several research questions have been opened up for further research.

At first, the Service Intention Phase, as partly developed by Walter 2010 and partly expanded upon in this work, could make for a thorough update to seamlessly work within the methodology framework presented here, such that the creation, modification and evaluation along the demand model are directly integrated into the design process instead of the currently manual data transfer.

A medium-term amendment to the Service Intention Phase could see a reverse application of the trajectory construction and matching algorithm used in the Target Timetable Phase: From the possible hub types and riding time ranges, the very location of hubs could be evaluated also in a timetable view, providing a preliminary design basis for the evaluation of station locations.

The Stage Development Phase, as only sketched within this work, is to be soundly elaborated, keeping the feasible timetable as a long-term goal, but incorporating maintenance, re-investment, and intermediate efficacies in the design process.

In the Feasible Timetable Phase, it is thinkable to process the feasible timetable by the use of PESP rather than the guided manual process as described in this work. With the approaches of Schlechte et al. and Sewcyk 2004, a timetable design with fixed hub types and the preliminary infrastructure dimensioning presented in section 5.5 should yield the possibility of assigning a purpose to every track element and thereby constructing a target infrastructure. This process should, nevertheless, feature the possibility of feedback loops to the set of target timetables and retain manual intervention interfaces in case of hub type or route conflicts.

As noted in several places throughout this work, freight traffic has not been incorporated in this methodology. Freight traffic cannot be directly incorporated within this methodology, for it is crucial for freight trains *not* to be present at a timetable hub during the hub time, which in turn affects the truth function. Furthermore, we cannot consider freight trains to feature similar (i) reliability, (ii) homogeneity of riding times, and (iii) periodicity to passenger traffic at the moment. However, following the approaches of a *path catalogue* as presented in sections 2.3.7 and 3.1.1, we can use the findings of *Boolean interval parting* in section 5.4.3: By constructing both a passenger network truth function and a freight network truth function, a coupling function between them could make it possible to work out catalogue paths for freight trains. However, taking the findings of Pöhle et al. and Nachtigall, Noll, et al., it seems, for the time being, sufficient to model passenger traffic first and only then assess the possibilities of incorporating a freight traffic path catalogue (Pöhle et al. 2012 and Nachtigall, Noll, et al. 2014).

Within the construction, matching, and conflict detection algorithm, several tweaks are thinkable to improve the accuracy of the Target Timetable Phase: First, the minimum and maximum speed could be defined per edge and per train system individually, so as to account for different boundary conditions, such as high-speed lines, mountainous lines, or lines with short station spacing. Second, the maximum allowable transfer time could be changed to a range, rather than one discrete value, depending on the node flows. Third, an automatic feedback between trajectory construction and the number of lines within a hub could automatically check for possible interval partings and accumulations of parallel station entries/exits as early as upon preprocessing. Finally, the installation of directional transfers of type C could be handled more flexibly when only one interval is in place, moving the hub location rather than changing the riding times.

Finally, there are minor, only partly scientific, issues to be tackled in consecutive works: (i) interval parting with different intervals as described in section 5.4.3; (ii) an integrated design of peak and off-peak timetables, rather than the design of separate timetable models as pursued in this work; (iii) the programming of a graphical user interface for easier data input; and (iv) the possibility of directly exporting the results of the (MATLAB and PYTHON) algorithms to the (MATLAB) script for the creation of three-dimensional network and timetable graphs.

Bibliography

Ahrens, Gerd-Axel, Rico Wittwer, Stefan Hubrich, Sebastian Wittig, and Frank Ließke (2015a). *Sonderauswertung zum Forschungsprojekt "Mobilität in Städten—SrV 2013." SrV-Stadtgruppe: Mittelzentren, Topografie: flach.* Tech. rep. TU Dresden, Fakultät Verkehrswissenschaften "Friedrich List", Institut für Verkehrsplanung und Straßenverkehr.

Ahrens, Gerd-Axel, Rico Wittwer, Stefan Hubrich, Sebastian Wittig, and Frank Ließke (2015b). *Sonderauswertung zum Forschungsprojekt "Mobilität in Städten—SrV 2013". SrV-Stadtgruppe: Mittelzentren, Topografie: flach.* Tech. rep. TU Dresden, Fakultät Verkehrswissenschaften "Friedrich List", Institut für Verkehrsplanung und Straßenverkehr.

Ahrens, Gerd-Axel, Rico Wittwer, Stefan Hubrich, Sebastian Wittig, and Frank Ließke (2015c). *Sonderauswertung zum Forschungsprojekt "Mobilität in Städten—SrV 2013". SrV-Stadtgruppe: Oberzentren 500.000 und mehr EW (ohne Berlin), Topografie: flach.* Tech. rep. TU Dresden, Fakultät Verkehrswissenschaften "Friedrich List", Institut für Verkehrsplanung und Straßenverkehr.

Ahrens, Gerd-Axel, Rico Wittwer, Stefan Hubrich, Sebastian Wittig, and Frank Ließke (2015d). *Sonderauswertung zum Forschungsprojekt "Mobilität in Städten—SrV 2013". SrV-Stadtgruppe: Oberzentren bis unter 500.000 EW, Topografie: flach.* Tech. rep. TU Dresden, Fakultät Verkehrswissenschaften "Friedrich List", Institut für Verkehrsplanung und Straßenverkehr.

Ahrens, Gerd-Axel, Rico Wittwer, Stefan Hubrich, Sebastian Wittig, and Frank Ließke (2015e). *Sonderauswertung zum Forschungsprojekt "Mobilität in Städten—SrV 2013". SrV-Stadtgruppe: Oberzentren bis unter 500.000 EW, Topografie: hügelig.* Tech. rep. TU Dresden, Fakultät Verkehrswissenschaften „Friedrich List",Institut für Verkehrsplanung und Straßenverkehr.

Ahrens, Gerd-Axel, Rico Wittwer, Stefan Hubrich, Sebastian Wittig, and Frank Ließke (2015f). *Sonderauswertung zum Forschungsprojekt "Mobilität in Städten—SrV 2013". SrV-Stadtgruppe: Unter-/Grund-/Kleinzentren/ländliche Gemeinden, Topografie: flach.* Tech. rep. TU Dresden, Fakultät Verkehrswissenschaften "Friedrich List", Institut für Verkehrsplanung und Straßenverkehr.

Ahrens, Gerd-Axel, Rico Wittwer, Stefan Hubrich, Sebastian Wittig, and Frank Ließke (2015g). *Sonderauswertung zum Forschungsprojekt "Mobilität in Städten—SrV 2013". Stadtgruppe: SrV-Städtepegel.* Tech. rep. TU Dresden, Fakultät Verkehrswissenschaften "Friedrich List", Institut für Verkehrsplanung und Straßenverkehr.

Baudyš, Karel, Vít Janoš, and Jiři Pospíšil (2009). Railway Timetable in Czech Republic. In: *Proceedings of the 6th International Scientific Conference TRANSBALTICA 2009*, pp. 7–10.

BAV (2016). *Zukünftige Entwicklung der Bahninfrastruktur (ZEB)*. online; last accessed: January 9, 2016. URL: `http://www.bav.admin.ch/zeb/`.

Berghold, Armin (2011). "Die Gleislagequalität in Dauerlangsamfahrstellen". German. MA thesis. Technische Universität Graz.

Bickel, Thomas, Eric Cosandey, and Hansruedi Akerman (2010). Deciding for the Right Timetable Production System. In: *EURAILMag Business and Technology* 22, pp. 202–205.

Boehm, B. W. (1988). A spiral model of software development and enhancement. English. In: *Computer* 21.5, pp. 61–72.

Bopp, Bernd (2004). "Inverse Kapazitätsermittlung: Ermittlung der Infrastruktur unter der Vorgabe eines Zielfahrplans, Beispiel: Querspange Erbach–Ehingen". Diploma thesis. Karlsruhe Institute of Technology.

Borza, Viktor, Laszlo Kormanyos, and Bela Vincze (2005). "New regular interval timetables in operation on the suburban lines of the Hungarian State Railways". In: *Proceedings of ŽEL 2005*. Žilina: University of Žilina, p. 57.

Brezina, Tadej and Hermann Knoflacher (2014). Railway trip speeds and areal coverage. The emperor's new clothes of effectivity? In: *Journal of Transport Geography* 39, pp. 121–130.

Bundesnetzagentur (2016a). *Durchschnittliche Reiseweite im deutschen Schienenpersonenfernverkehr von 2002 bis 2009 (in Kilometer)*. online; last accessed: July 26, 2016. URL: `http://de.statista.com/statistik/daten/studie/168414/umfrage/reiseweite-im-schienenpersonenfernverkehr-in-deutschland-seit-2002/`..

Bundesnetzagentur (2016b). *Durchschnittliche Reiseweite im deutschen Schienenpersonennahverkehr von 2002 bis 2009 (in Kilometer)*. online; last accessed: July 26, 2016. URL: `http://de.statista.com/statistik/daten/studie/168416/umfrage/reiseweite-im-schienenpersonennahverkehr-in-deutschland-seit-2002/`..

Bussieck, Michael (1998). "Optimal Lines in Public Rail Transport". Dissertation. Technische Universität Braunschweig.

Cacchiani, Valentina and Paolo Toth (2012). Nominal and robust train timetabling problems. In: *European Journal of Operational Research* 219.3, pp. 727–737.

Caimi, Gabrio (2009). "Algorithmic decision suport for train scheduling in a large and highly utilised railway network". Dissertation. ETH Zürich.

Caimi, Gabrio, Marco Laumanns, Kaspar Schüpbach, Stefan Wörner, and Martin Fuchsberger (2011). The periodic service intention as a conceptual framework for generating timetables with partial periodicity. English. In: *Transportation Planning and Technology* 34.4, pp. 323–339.

Chen, L., C. Roberts, F. Schmid, and E. Stewart (2015). Modeling and Solving Real-Time Train Rescheduling Problems in Railway Bottleneck Sections. In: *IEEE Transactions on Intelligent Transportation Systems* 16.4. ID: 1, pp. 1896–1904.

Chierici, Alessandro, Roberto Cordone, and Roberto Maja (2004). The demand-dependent optimization of regular train timetables. In: *Electronic Notes in Discrete Mathematics* 17, pp. 99–104.

Clever, Reinhard (1997). Integrated Timed Transfer. A European Perspective. In: *Transportation Research Record 1571.*

Curtis, W., H. Krasner, V. Shen, and N. Iscoe (1987). "On building software process models under the lamppost". English. In: IEEE Computer Society Press, pp. 96–103. ISBN: 0270-5257.

Dewilde, Thijs, Peter Sels, Dirk Cattrysse, and Pieter Vansteenwegen (2013). Robust railway station planning: An interaction between routing, timetabling and platforming. In: *Journal of Rail Transport Planning & Management* 3.3, pp. 68–77.

Ellwanger, Gunther (2004). Hochgeschwindigkeitsverkehr in Europa weiter auf Erfolgskurs – Verkehrsprognosen 2020. In: *Eisenbahntechnische Rundschau* 7+8.53, pp. 416–423.

Felfernig, Alexander, Stefan Reiterer, Florian Reinfrank, Gerald Ninaus, and Michael Jeran (2014). Chapter 7 - Conflict Detection and Diagnosis in Configuration. In: ed. by Alexander Felfernig, Lothar Hotz, Claire Bagley, and Juha Tiihonen. Knowledge-Based Configuration. Boston: Morgan Kaufmann, pp. 73–87. ISBN: 9780124158177.

Felfernig, Alexander and Monika Schubert (2010). "FastDiag: A Diagnosis Algorithm for Inconsistent Constraint Sets". In: *21st International Workshop on Principles of Diagnosis.* Portland, p. 31.

Franco, John and John Martin (2009). A History of Satisfiability. In: ed. by Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh. Handbook of Satisfiability. Amsterdam: IOS Press, pp. 3–74.

Friedrich, M., T. Hofsäss, and S. Wekeck (2001). Timetable-based Transit Assignment Using Branch & Bound Techniques. In: *Transportation Research Record 1852*, pp. 100–107.

Geerts, Jean-François (2013). *Vers une nouvelle offre pour 2020–2030.Etude d'orientation stratégique et première phase de mise en œuvre.* Presentation at it13.rail, Zürich.

Hantsch, Fabian, Xiaojun Li, and Ullrich Martin (2013). Methoden zur Engpassanalyse bei der Infrastrukturbemessung im Schienenverkehr. In: *Eisenbahntechnische Rundschau* 03, pp. 30–33.

Herrigel-Wiedersheim, Sabrina (2015). "Algorithmic Decision Support for the Construction of Periodic Railway Timetables". Dissertation. ETH Zürich.

Hesse, Wolfgang, Karel Baudyš, Vít Janoš, Jiři Pospíšil, and Michael Guckert (2010). Landesweiter Integraler Taktfahrplan für Tschechien - eine Studie. In: *Eisenbahn-Revue International* 11, pp. 596–599.

Hesse, Wolfgang, Michael Guckert, and Roland Hesse (2005). “OptiTakt – A tool for developing and evaluating periodic timetables”. In: *Proceedings of the 1st International Seminar on Railway Operations Modelling*. Ed. by International Association of Railway Operations Research, p. 1.

Hesse, Wolfgang, Michael Guckert, Johannes Scheider, and Andreas Schulz (2000). Werkzeuggestützte Entwicklung eines Integralen Taktfahrplans für Nordost-Bayern. In: *Internationales Verkehrswesen* 6, pp. 264–268.

Johnson, Daniel, Jeremy Shires, Chris Nash, and Jonathan Tyler (2006). Forecasting and appraising the impact of a regular interval timetable. In: *Transport Policy* 13.5, pp. 349–366.

Kang, Min-Wook, Manoj K. Jha, and Ramesh Buddharaju (2011). “A Rail Transit Route Optimization Model for Rail Infrastructure Planning and Design: Case Study of St Andrews, Scotland”. In: *TRB annual meeting 2012*. Washington, DC.

Kolonko, Michael and Ophelia Engelhardt-Funke (2002). Mathematische Optimierung in der Praxis. Genetisch optimierte Fahrpläne und Kosten-Nutzen-Analysen für Verkehrsnetze. In: *TUContact* 10, pp. 31–34.

Kormanyos, Laszlo (2007). “Building up a railway-based integrated public transport system in Hungary”. In: *From horse-drawn railway to high-speed transportation system*. Ed. by Bohumil Kubát. Praha: ČVUT, p. 121.

Kroon, Leo, Rommert Dekker, and Michiel Vromans (2004). Cyclic Railway Timetabling: A Stochastic Optimization Approach. In: *Railway Optimization* 4359, pp. 41–66.

Lai, Yung-Cheng and Po-Wen Huang (2011). “High Speed Route Improvement Optimizer”. In: *TRB annual meeting 2012*. Washington, DC.

Larman, Craig and Victor R. Basili (2003). Iterative and incremental development: A brief history. English. In: *Computer* 36.6, pp. 47–56.

Lee, Yusin and Chuen-Yih Chen (2009). A heuristic for the train pathing and timetabling problem. In: *Transportation Research Part B: Methodological* 43.8–9, pp. 837–851.

Li, Xiaojun and Ullrich Martin (2014). Einfluss des Betriebsprogramms und der Infrastrukturgestaltung auf die Entstehung von Engpässen. In: *Eisenbahntechnische Rundschau* 03, pp. 16–21.

Lichtenegger, Michael (1990). “Der Taktfahrplan; Abbildung und Konstruktion mit Hilfe der Graphentheorie ; Minimierung der Realisierungskosten”. Dissertation. Technische Universität Graz.

Liebchen, Christian (2004). Symmetry for Periodic Railway Timetables. In: *Electronic Notes in Theoretical Computer Science* 92, pp. 34–51.

Liebchen, Christian (2005). Fahrplanoptimierung im Personenverkehr – muss es immer ITF sein? In: *ETR - Eisenbahntechnische Rundschau* 11, pp. 689–702.

Liebchen, Christian (2006). "Periodic timetable optimization in public transport". English. Dissertation. Technische Universität Berlin. ISBN: 386624150X, 9783866241503.

Liebchen, Christian (2008). Planung im Nah- und Fernverkehr: Mathematische Optimierungsverfahren decken mehr Ziele ab. In: *Eisenbahntechnische Rundschau* 06, pp. 363–368.

Liebchen, Christian and Rolf Möhring H. (2002). A Case Study in Periodic Timetabling. In: *Electronic Notes in Theoretical Computer Science* 66.6, pp. 18–31.

Liebchen, Christian and Rolf H. Möhring (2007). "The modeling power of the periodic event scheduling problem: Railway timetables - And beyond". English. In: vol. 4359, pp. 3–40. ISBN: 354074245X, 9783540742456.

Liebchen, Christian, Rolf Möhring, and Frank Wagner (2004). "Infrastructure Update According to Schedule?" In: *3rd International Conference on Applied Infrastructure Research. Network Economics: Financing, Regulation and Capacity Allocation in Infrastructure Sectors*. Berlin: Berlin University of Technology.

Maxwell, Ross (1999). Intercity Rail Fixed-Interval, Timed-Transfer, Multihub System: Applicability of the Integraler Taktfahrplan Strategy to North America. In: *Transportation Research Record: Journal of the Transportation Research Board* 1691. http://dx.doi.org/10.3141/1691-01, pp. 1–11.

Melzow, Rainer (1997). Streckenstandards. In: *EI – Eisenbahningenieur* 48.7, pp. 36–40.

Michaelis, Mathias and Anita Schöbel (2009). Integrating line planning, timetabling, and vehicle scheduling: A customer-oriented heuristic. English. In: *Public Transport* 1.3, pp. 211–232.

Mills, H. D. (1976). Software Development. English. In: *IEEE Transactions on Software Engineering* SE-2.4, pp. 265–273.

Nachtigall, Karl, Oliver Noll, and Daniel Pöhle (2014). Ein innovatives Belegungsverfahren für den zukünftigen industrialisierten Fahrplanprozess. In: *Eisenbahntechnische Rundschau* 12, pp. 28–33.

Nachtigall, Karl and Jens Opitz (2008). Taktfahrlagen auf komplexen Infrastrukturen mittels iterativer lokaler Konfliktauflösung. In: *ETR* 6, pp. 369–373.

ÖBB (2011). *Zielnetz 2025+*. online; last accessed: September 3, 2016. Wien: ÖBB-Infrastruktur AG. URL: `http://www.oebb.at/infrastruktur/de/5_0_fuer_Generationen/5_4_`

Wir_bauen_fuer_Generationen/5_4_1_Schieneninfrastruktur/Zukunftbahn_Zielnetz_2025/__Dms_Dateien/_Downloads_Zukunftbahn.jsp.

Opitz, Jens (2009). “Automatische Erzeugung und Optimierung von Taktfahrplänen in Schienenverkehrsnetzen”. German. Dissertation. Technische Universität Dresden. ISBN: 9783834921284; 3834921289.

ÖROK (1992). *ÖROK-Empfehlungen zur Weiterentwicklung des NAT 91 zum IT 2000.* 1st ed. Wien: Österreichische Raumordnungskonferenz.

Pachl, Jörn (1999). Standardisierung der Infrastruktur kleiner Knoten. In: *EI – Eisenbahningenieur* 50.4, pp. 14–20.

Pachl, Jörn (2011). *Systemtechnik des Schienenverkehrs: Bahnbetrieb planen, steuern und sichern.* German. 6., überarbeitete Auflage. Wiesbaden: Vieweg+Teubner. ISBN: 3834814288; 9783834814289.

Peeters, Leon (2003). “Cyclic Railway Timetable Optimization”. Dissertation. Erasmus Universiteit Rotterdam.

Pöhle, Daniel, Reyk Weiß, Jens Opitz, and Karl Nachtigall (2012). Neuartiges automatisches Routing in realen Eisenbahnnetzen mit dem Programmsystem TAKT. In: *ETR - Eisenbahntechnische Rundschau* 10, pp. 59–65.

PTV Group (2016). *Schematic line diagram.* online; last accessed: August 16, 2016. URL: http://vision-traffic.ptvgroup.com/en-uk/lp/ptv-visum/use-cases/schematic-line-diagram/.

Reinold, Sittipan and Christian Blome (2012). Hamburg – Köln in 2,5 Stunden? Lösungskonzept zur Beschleunigung des SPFV. In: *Eisenbahntechnische Rundschau* 7+8, pp. 46–49.

Rittner, Michael and Karl Nachtigall (2009). Simultane Liniennetz- und Taktfahrlagenoptimierung. In: *EI – Eisenbahningenieur* 06, pp. 6–10.

Royce, W. (1987). “Managing the development of large software systems: concepts and techniques”. English. In: IEEE Computer Society Press, pp. 328–338. ISBN: 0897912160; 9780897912167.

Schaer, Thorsten (2013). *Eisenbahnbetriebswissenschaftliche Untersuchungen in der Langfristfahrplanung der DB Netz AG.* German. Presentation at FBS-Anwendertreffen 2013. Leipzig.

Scheidt, Martin (2016). Zyklus der Gestaltung von Eisenbahninfrastruktur. In: *Eisenbahntechnische Rundschau* 3, pp. 61–63.

Scheller, Albert and Frank Kutzner (2015). 2. Stammstrecke München. Zentrales Element des Bahnknoten-Konzeptes. In: *ETR - Eisenbahntechnische Rundschau* 5, pp. 16–19.

Schlechte, Thomas, Ralf Borndörfer, Berkan Erol, Thomas Graffagnino, and Elmar Swarat (2011). Micro–macro transformation of railway networks. In: *Journal of Rail Transport Planning & Management* 1.1, pp. 38–48.

Schröder, Michael and Ingmar Schüle (2008). Interaktive mehrkriterielle Optimierung für die regionale Fahrplanabstimmung in Verkehrsverbünden. In: *Straßenverkehrstechnik* 6, pp. 332–340.

Schwager, Dianne (2003). Transit Capacity and Quality of Service Manual. In: 2. Washington, DC: Transportation Research Board. Chap. 3, pp. 3.1–3.94.

Serafini, P. and W. Ukovich (1989). A Mathematical for Periodic Scheduling Problems. In: *SIAM J.Discret.Math.* 2.4, pp. 550–581. URL: http://dx.doi.org/10.1137/0402049.

Sewcyk, Bernd (2004). *Makroskopische Abbildung des Eisenbahnbetriebes in Modellen zur langfristigen Infrastrukturplanung.* Hannover: EurailPress.

Smoliner, Martin, Stefan Walter, and Martin Fellendorf (2015). "Light Rail Hubs in a Multimodal Transport Environment". In: *13th National Light Rail & Streetcar Conference "Transforming Urban Areas".* Ed. by APTA and TRB. Minneapolis: APTA; TRB.

Sparmann, Udo (2006). Abschied vom Taktfahrplan? In: *Der Nahverkehr* 3, pp. 12–20.

Stergidou, Anastasia, Konstantinos Panou, and Panagiotis Tzieropoulos (2013). "Critical Assessment and Benchmarking of Train Timetabling Methods". In: *Proceedings of the 13th WCTR.* Rio de Janeiro, p. 1.

Stohler, Werner (2008). Liberalisierung des internationalen Personenverkehrs – Konsequenzen für die Fahrplanung. In: *ETR - Eisenbahntechnische Rundschau* 3, pp. 107–112.

Stohler, Werner, Luigi Stähli, and Philipp Schröder (2012). Geschichte/Entwicklungsperspektiven der langfristigen Fahrplan-Planungsprozesse in Europa. In: *ETR - Eisenbahntechnische Rundschau* 7+8, pp. 30–36.

Streitzig, Constanze, Jens Opitz, and Karl Nachtigall (2010). Automatische Berechnung von Einzelfahrlagen in Taktfahrplänen. In: *EI – Eisenbahningenieur* 06, pp. 48–51.

Swartout, William and Robert Balzer (1982). On the inevitable intertwining of specification and implementation. English. In: *Communications of the ACM* 25.7, pp. 438–440.

Tron, David, Jean-Daniel Buri, and Panagiotis Tzieropoulos (2010). "From the reticular diagram to the 24-hours timetable". In: *Swiss Transport Research Conference.* Ed. by 21. Monte Verità/Ascona: EPFL Lausanne, p. 1.

Tzieropoulos, Panagiotis, Anne Curchod, and Yves Putallaz (2008). "FASTA: a new life for a former dinosaur adapting the modelling scope to new planning requirements". English. In: *Computers in Railways XI: Computer System Design and Operation in the Railway and Other*

Transit Systems. Ed. by J Allan, E Arias, CA Brebbia, C Goodman, AF Rumsey, G Sciutto, and N Tomii. Vol. 103. WIT Transactions on the Built Environment. 11th International Conference on Computer System Design and Operation in the Railway and other Transit Systems, Toledo, SPAIN, SEP 15-17, 2008. Wessex Inst Technol; Univ Castilla; WIT Transac Built Environm. Ashurst: WIT PRESS, pp. 15–24. ISBN: 978-1-84564-126-9. DOI: 10.2495/CR080021.

Uhlmann, Matthias and Karsten Mutschink (2004). Bemessung komplexer Eisenbahninfrastruktur – Die konstruktive Methode. In: *Eisenbahntechnische Rundschau* 7+8.53, pp. 506–514.

U.S. Department of Energy (2008). *Managing Design and Construction Using Systems Engineering*. Washington, DC.

Uttenthaler, Helmut (2010). “Grundlagen eines auf einem integrierten Taktfahrplan basierenden Eisenbahninfrastrukturausbaues am Beispiel Zentraleuropa”. Diploma thesis. Technische Universität Graz.

Vansteenwegen, Pieter and Dirk van Oudheusden (2006). Developing railway timetables which guarantee a better service. English. In: *European Journal of Operational Research* 173.1, pp. 337–350.

Veit, Peter, Martin Fellendorf, Martin Smoliner, Karl Hofer, and Stefan Walter (2016). *Verkehrskonzept Radkersburger Bahn. Final Report (in German)*. Tech. rep. Graz University of Technology.

Veit, Peter, Stefan Walter, and Martin Fellendorf (2014,). *GKB Weissbuch 2025+. Strategic Plan for GKB regional railway in Styria. Final Report (in German)*. Tech. rep. Graz University of Technology.

Walter, Stefan (2010). “Nachfrageorientierte Liniennetzoptimierung am Beispiel Graz”. German. MA thesis. Technische Universität Graz.

Walter, Stefan (2013). “The Truly Integrated Integrated Timetable”. In: *Proceedings of the 11th European Transport Congress*. Ed. by Miroslav Svítek. Prague: CVUT Praha, p. 292.

Walter, Stefan (2016). Langfristige nachfragebasierte Infrastrukturentwicklung für Regionalbahnen. In: *ZEVrail* 140.3, pp. 77–87.

Walter, Stefan and Martin Fellendorf (2015). Long-Term Upgrade Strategy for Light Rail and Regional Rail. In: *Transportation Research Record: Journal of the Transportation Research Board* 2534. http://dx.doi.org/10.3141/2534-06, pp. 38–47. URL: http://dx.doi.org/10.3141/2534-06.

Wardman, Mark, Jeremy Shires, William Lythgoe, and Jonathan Tyler (2004). Consumer benefits and demand impacts of regular train timetables. In: *International Journal of Transport Management* 2.1, pp. 39–49.

Weigand, Werner (2005). Dimensionierung der Eisenbahninfrastruktur. In: *EI – Eisenbahningenieur* 1.56, pp. 8–12.

Weis, Philipp (2005). "Konstruktionsprinzip eines Taktfahrplans; Eine Strategie für den Eisenbahnverkehr der Staaten Kroatien, Ungarn, Slowenien und Österreich". Diploma thesis. Technische Universität Graz, 126 Bl.

Wieczorek, Tobias (2006). "Inverse Kapazitätsbestimmung für die Eisenbahn-Infrastruktur". Dissertation. Technische Universität Karlsruhe. ISBN: 3-7771-0343-8.

Wittwer, Rico (2016a). *Sonderauswertung zum Forschungsprojekt „Mobilität in Städten – SrV 2013. SrV-Stadtgruppe: Oberzentren 500.000 und mehr EW (ohne Berlin). Special evaluation on behalf of Stefan Walter for long-distance trains.* Tech. rep. TU Dresden, Fakultät Verkehrswissenschaften „Friedrich List", Institut für Verkehrsplanung und Straßenverkehr.

Wittwer, Rico (2016b). *Sonderauswertung zum Forschungsprojekt „Mobilität in Städten – SrV 2013. SrV-Stadtgruppe: Oberzentren bis unter 500.000 EW - flach . Special evaluation on behalf of Stefan Walter for long-distance trains.* Tech. rep. TU Dresden, Fakultät Verkehrswissenschaften „Friedrich List", Institut für Verkehrsplanung und Straßenverkehr.

Yang, D., L. Nie, Y. Tan, Z. He, and Y. Zhang (2010). "Working out an incomplete cyclic train timetable for high-speed railways by computer". English. In: *Computers in Railways XII*. Vol. 114, pp. 889–900. ISBN: 1743-3509.

List of Acronyms Used

IID Iterative and Incremental Development

ITF Integrated Timetable

IC Intercity

SAT Boolean Satisfiability Problem

LRT Light Rail

HSDAG Hitting Set Directed Acyclic Graph

OD Origin-Destination

RJ Railjet

PESP Periodic Event Scheduling Problem

EC Eurocity

REX Regional-Express

RE Regio-Express

IR Interregio

CBD Central Business District

TSI Technical Specifications for Interoperability

ICE Intercity Express

TGV Train à Grande Vitesse

TEN-T Trans-European Transport Networks

Symbols in Use

Δt: the shift of directional transfer hubs (type C) towards the axis of symmetry as described in section 5.3.1.

i, j: $i, j \in \mathbb{N}$; index variables.

k_{sr}: relative weight for travel distances used for node flow weighting.

l_{edge}: length of an edge, usually in km.

l_{edge}: signal block length for use in the preliminary infrastructure dimensioning as described in section 5.5.

m, n: $m, n \in \mathbb{N}$; integers used for the expression of whole-number multiples.

$n_{\Delta t}$: number of different values of Δt in place in case of interval parting, as described in section 5.4.3.

n_{max}: maximum multiple of T considered upon trajectory construction.

n_{Tr}: the number of train systems in place for a given examination.

ν_i: divisor to describe denser intervals by writing $T_i = {}^{T}/_{\nu_i}$. Train systems with half the basic interval, i. e. $T_i = {}^{T}/_{2}$, therefore run at $\nu_i = 2$.

$p_{\text{reduction}}$: limit to the percentage of riding time reduction considered attainable, as used in section 6.1.1.

q: volumes of *node flows* in hubs, used to quantify passenger numbers transferring in a station.

q_{mod}: weighted passenger volumes in node flow as described in section 2.3.5.

$q_{\text{local}}, q_{\text{mod, local}}$: (weighted) passenger volumes in node flow bound for the hub itself.

$q_{tr,\text{lost}}, q_{tr,\text{mod, lost}}$: (weighted) transfer passenger volumes with lost transfers.

$q_{tr}, q_{\text{mod},tr}$: (weighted) sum of transfer passenger volumes in node flow.

$s_{\text{crossing}}, s_{\text{overtaking}}$: distane from a hub to the point of crossing or overtaking, as described in section 6.1.1.

s_r: average travel distance

T: the basic interval used for reference purposes.

T_{branch}: the interval on a branch line.

T_{chain}: the interval of a passenger in a longer travel chain.

$\boldsymbol{T}_{\textbf{dense}}$: the dense interval in selectively served hubs as described in section 5.1.3.

$\boldsymbol{T_i}$: the interval currently under examination.

$\boldsymbol{T}_{\text{joint}}$: combined interval in place on jointly served stretches with interval parting as described in section 5.2.1.

$\boldsymbol{t}_{\textbf{dwell}}$: *dwell time*, i. e. the time to allow for boarding and alighting of passengers.

$\boldsymbol{t}_{\textbf{crossing}}, \boldsymbol{t}_{\textbf{overtaking}}$: time from the start at a hub until a crossing or overtaking, as described in section 6.1.1.

$\boldsymbol{t_h}$: *stop time*, i. e. the time spent within a hub for each individual train.

$\boldsymbol{t}_{\boldsymbol{h},\textbf{prop}}$: *proportional hub stop time* attributed to an edge.

$\boldsymbol{t}_{\textbf{hub}}$: *hub time*, i. e. the point in time around which a hub transfer takes place, e. g. minutes .00 or .30 in a full hub.

$\boldsymbol{t}_{\textbf{offset}}$: offset between two trains for the calculation of crossings and overtakings, used for sectional target riding times as described in section 6.2.3.

$\boldsymbol{t_r}$: *riding time.* For riding time relations, see section 2.2.3.

$\boldsymbol{t}_{\boldsymbol{r},\textbf{edge}}$: *edge riding time* as explained in section 2.2.3.

$\boldsymbol{t}_{\boldsymbol{r},\textbf{net}}$: *net riding time*, also called *technical running time.*

$\boldsymbol{t}_{\boldsymbol{r},\textbf{gr}}$: *gross riding time.*

$\boldsymbol{t}_{\boldsymbol{r},\textbf{round trip}}$: riding time for one round trip, i. e. a trip from one terminus to the other and back.

$\boldsymbol{t}_{\boldsymbol{r},\textbf{status quo}}$: status quo riding time.

$\boldsymbol{t}_{\boldsymbol{r},\textbf{target}}$: *target riding time* as described in section 5.5.

$\boldsymbol{t_s}$: *hub spread time*, i. e. the total time between the arrival of the first train and the departure of the last train within a hub, as described in section 3.3.3.

$\boldsymbol{t}_{\textbf{shift}}$: initial value for a global shift of all trajectories in the case of inner-city trunk lines as described in section 5.4.3.

$\boldsymbol{t}_{\textbf{signal headway}}$: signal headway, i. e. the technical minimum for two trains to follow each other.

$\boldsymbol{t}_{\textbf{tr}}$: *transfer time*, i. e. the time to change between trains.

$\boldsymbol{t_{\mathrm{tr,max}}}$: *maximum allowable transfer time*, i. e. the maximum time we allow a transfer to take in order to be considered *satisfied*.

$\boldsymbol{v_i}$: riding speed (i. e. $l_{\mathrm{edge}}/t_{r,\mathrm{edge}}$) of a train system.

$\boldsymbol{v_{\mathrm{max}}}$: maximum riding speed per train system, used for trajectory construction as described in section 5.3.4.

$\boldsymbol{v_{\mathrm{min}}}$: minimum riding speed per train system.

$\boldsymbol{w}$: percentage *recovery time* added onto the *net riding time* $t_{r,\mathrm{net}}$ to calculate the *gross riding time* $t_{r,\mathrm{gr}}$.

Technical Algorithm Documentation

Overview

This algorithm has been written partly in Matlab R2009b (data collection, trajectory construction, and hub type conversion) and Python 3.5 64bit (conflict detection, conflict resolution). The following extra packages were used:

in Matlab : `str2adj`, `numnodes`, `numedges`, `cell2str`, `cell2num`[64].

in Python : `pyeda`, `math`, `time`

The algorithm can be split into three groups: (i) data collection and preprocessing, (ii) calculation and conversion, and (iii) conflict detection and resolution.

Data Collection

This section covers the input of data.

`fill_matrices` is a wrapper script to fill all basic data. It calls the following scripts sequentially:

`fillmhubs` fills the file `hubs.txt` with the following structure:

```
hub_name pos_x pos_y no_tracks id
```

where `hub_name` is the human-readable name of the hub, `pos_x` and `pos_y` are the coordinated for a (future) plotting of the network, `no_tracks` is the parameter for the number of platform tracks per station, to be used in the preliminary infrastructure dimensioning. Finally, `id` denotes the ID of the hub to be used for addressing it in the algorithm.

`fillmedges` fills the file `edges.txt` with the following structure:

```
id hub_1 hub_2 length no_tracks
```

where `id` is the edge ID, `hub_1` and `hub_2` denote the IDs of the adjacent hubs, `length` is the length of the edge im km; `no_tracks` finally denotes the number of tracks for the preliminary infrastructure dimensioning.

`fillmtrainsys` fills the file `trainsys.txt`[65], with the following structure:

```
traintype v_min v_max
```

where `traintype` is the name of the train system, and `v_minv_max` denote the minimum and maximum speed per train system.

[64]Note that several of these packages are included from Matlab 2014a.
[65]legacy naming before the clear separation of *train type* (RJ, IC, etc.) and *train system* (RJ1, S11, etc.).

`fillmadjmat` fills an adjacency matrix of edges and hubs to create an overview which hubs are connected by edges. This is done with the `accumarray` MATLAB function.

`fillmdistmat` creates a copy of the adjacency matrix, adding the distances between hubs as entries.

`fillmnotrains` collects the informations about existing hubs and train systems to fill the file `notrains.txt` with the following structure:

```
edge_id tt_1 tt_2 ... tt_n
```

where `edge_id` depicts the edge and `tt_1` to `tt_n` show the number of trains per train type per edge. An edge no. 7 with two IC, one REX, and two S-Bahn lines would therefore read:

```
edge_id tt_RJ tt_IC tt_REX tt_R tt_S
7       0     2     1      0    2
```

`fillmtrains` finally creates the file `trainlist.txt`, which is a list of all train systems, with the following structure:

```
train_id edge traintype nu
```

where `train_id` is a unique train system identifier, `edge` denotes the edge ID it rides on, `traintype` denotes its train type (RJ, IC, etc.), and `nu` denotes the ν_i of the train system.

Calculation and Conversion

The second set of scripts performs the calculation of hub types, riding times, and the hub type conversion needed for the hub type conflict resolution.

`calc` is a wrapper script that invokes the calculation procedures. It calls the following scripts sequentially:

`calcsearchspace` sets up the `searchspace` needed for riding time calculation (see section 5.3.4). `searchspace` is a cell structure in MATLAB in the following form:

	list of ν_i →
A	
B	
$C_{\Delta t1,\mathrm{low}}$	
$C_{\Delta t1,\mathrm{high}}$	
$\vdots$	
$C_{\Delta tn,\mathrm{low}}$	
$C_{\Delta tn,\mathrm{high}}$	
D_{low}	
D_{high}	

Each line contains one hub type (and two hub subtypes for directional transfers, see section 5.3.5) and one `searchspace` for each instance of ν_i. The two subtypes for directional transfers are named *high* and *low* rather than *1* and *2*, since their indexing depends upon the search direction, as described in section 5.3.5.

`calcubinfo` is a script that retrieves the information about hubs and their adjacent edges. It creates one file `hubinfo_hubname.txt` per hub, where `hubname` is replaced by the corresponding hub name. Each file has the following structure:

```
hub_id edges nu_list
```

where `hub_id` is the hub ID, `edges` is a list of all adjacent edges, and `nu_list` is the list of ν_i present within that hub.

`calcridingtimes` is the script for riding time calculation as described in section 5.3.4. It creates the file `hublist.txt` which describes all possible hub type combinations found on the network. It has the following structure:

```
edge  hub_id_a  hub_id_b  train  nu  hub_type_a  ...
      ...  t_a  hub_type_b  t_b  t_r  v_r  profile
```

where `edge` denotes the edge a train system rides on, `hub_id_a` and `hub_id_b` denote the ID of the adjacent hubs, `train` denotes the train system ID, `nu` denotes the ν_i of this train system, `hub_type_a` and `hub_type_b` denote the types found for the adjacent hubs, and `t_a` as well as `t_b` denote the hub times found per hub,

As described in section 5.3.4, the types found for the adjacent hubs stem from the matching of `searchspace` on either side with the minimum and maximum riding speeds per train system. Since needed for the hub type conversion process, `t_a` and `t_b` are also written out.

`t_r` and `v_r` denote the resulting riding time and riding speed. None of these is crucial for the algorithm, but facilitates the backtracing for the hub type conflict resolution process (see section 5.4.6). Finally, `profile` specifies the riding time profile. This is necessary, since at the initial value of ν_i, every line corresponds to one profile. This is needed for the hub type conversion afterwards.

`calchubtypetransformation` performs the hub type conversion as described in section 5.3.6. The `hublist` is evaluated per line, only the lines with $\max(\nu_i)$ are read. For each of these lines, ν_i is set to the next lower value ν_{i+1}. Then, `t_a` and `t_b` are extracted; for either, new lines with `\t_a + T_i` and `\t_b + T_i` are appended. All are then compared to the entries in `searchspace` for the corresponding new ν_i. This is iterated until for all lines $\nu_i = 1$, which is then passed on to the file `hublist_transf`, which has the following form:

```
nu train hub_id_a hub_type_a t_a hub_id_b hub_type_b t_b profile
```

which is just a modified order of the inital `hublist`, with modified column ordering, reduced amount of columns, and entries of $\nu_i = 1$ only.

Note that converted train systems will feature ν_i lines for each profile (see section 5.3.6). The grouping of blocks of lines into profiles is necessary, since several instances of one profile will be connected such that the hub type columns are first connected with *or* and only then with *and*, while train systems with one line per profile are connected linewise only.

Conflict Diagnosis

The third part is dedicated to hub type conflict diagnosis. This is, as described in section 5.4.5, carried out by the HSDAG-wrapped FastDiag algorithm. The basic concept of FastDiag as described by Felfernig and Schubert 2010 is as follows:

function FastDiag($C \subset AC$, $AC = \{c_1, \dots c_t\}$): diagnosis Δ
  if $C = \emptyset$ **or** inconsistent($AC - C$) **then**
    return $\emptyset$;
  else
    FD($\emptyset, C, AC$);
  end if
end function
function FD($D, C = \{c_1, \dots c_q\}, AC$): diagnosis Δ
  if $D \neq \emptyset$ **and** consistent(AC) **then**
    return $\emptyset$;
  end if
  if singleton(C) **then**
    return C;
  end if
  $k = \lceil q/2 \rceil$;
  $C_1 = \{c_1, \dots c_k\}$; $C_2 = \{c_{k+1}, \dots c_q\}$;

$D_1 := \mathrm{FD}(C_2, C_1, AC - C_2)$;
$D_2 := \mathrm{FD}(D_1, C_2, AC - D_1)$;
return $(D_1 \cup D_2)$;
end function

where $C \subset AC$ are the *foreground constraints*, i. e. those that can be altered. $AC\ C$ therefore names the *background constraints*, i. e. those parts of the truth formula that are fixed initially. $c_1 \dots c_t$ denote these constraints in detail. Δ is the name of a *diagnosis*, i. e. a conflict set.

FD is the call for a recursive splitting and diagnosis function. D is the part of the constraints of the *other* half of the splitting. D_1 and D_2 are the results of the splitting to be further treated with FD.

This basic idea has been modelled into a HSDAG-wrapper as described in Felfernig and Schubert 2010. Thanks to Martin Mödlinger for bringing this script into a runtime-saving form.

The `fastdiag` script extracts the information from `hublist_transf` and converts it into Python dictionaries. Then, (i) every *profile* of a train system (see section 5.3.6) is combined first column-wise with *or* and then linewise with *and*, (ii) the profiles of each train system are combined with *or*, and (iii) the train systems are combined with *and*.

Then, the interface to the SAT solver is implemented. This is done with the following lines:

```
def consistent(indices_list, tsc):
  constraint_list = tsc.GetConstraintList(indices_list)

  if (len(constraint_list) < 1):
    return True
  X = And(*constraint_list)
  Y = X.tseitin('z')
  sat_result = Y.satisfy_one()
  if (sat_result == None or len(sat_result) < 1):
    return False
  else:
    return True

def inconsistent(indices_list, tsc):
  return not consistent(indices_list, tsc)
```

Note that, for each run, the truth formula is converted with Tseitin (see Franco et al. 2009: 20), which changes the truth formula to an *equisatisfiable* form, speeding up the SAT solver significantly. This *cannot* be used for the final solution, but is a convenient method for consistency checks.

Then, the FD and the FastDiag functions are defined:

```
def FD(D, C, AC, indent, tsc):
  if len(D) > 0 and consistent(AC, tsc):
    return []
  if singleton(C):
    return C
  k = ceil(len(C) / 2)

  C1 = C[:k]
  C2 = C[k:]

  AC_without_C2 = difference_lists(AC, C2)
  D1 = FD(C2, C1, AC_without_C2, indent + 1, tsc)

  AC_without_D1 = difference_lists(AC, D1)
  D2 = FD(D1, C2, AC_without_D1, indent + 1, tsc)
  return union_lists(D1, D2)

def FastDiag(C, AC, indent, tsc):
  print('  ' * indent + "FastDiag()")
  if len(C) == 0 or inconsistent(difference_lists(AC, C), tsc):
    result = []
  else:
    result = FD([], C, AC, indent + 1, tsc)
  print("FastDiag returning, delta = " + str(result))
  return result
```

Finally, the HSDAG-wrapper FastDiagFull is defined. Note that first the list of Δ, i. e. the list of already found conflict sets, is checked for the use in the HSDAG search tree (see section 5.4.5).

```
def IsElementInListOfDeltas(d, list_of_deltas):
  for delta in list_of_deltas:
    if d in delta:
      return True
  return False


def FastDiagFull(C, AC, list_of_deltas, indent, tsc):
  delta = FastDiag(C, AC, tsc)

  if (len(delta) > 0):
    list_of_deltas_before = list_of_deltas[:]
    list_of_deltas.append(delta)
    for d in delta:
      if not IsElementInListOfDeltas(d, list_of_deltas_before):
        FastDiagFull(difference_lists(C, [d]), AC, list_of_deltas, indent + 1, tsc)

def PrintSatResult(sat_result):
  for key, val in sat_result.items():
    if (val > 0):
```

```
      print (str(key) + ": " + str(val))
```

Then, the truth formulas are constructed and an initial satisfiability test is invoked:

```
tsc = TrainsystemCollection()
tsc.ReadFromFile('hublist_transf_mod.txt')

print("Generating formulas...")
tsc.MakeTruthFormulaForAllTrains()
print("Generating formulas completed!");

num_of_trains = len(tsc.trainsystems)
print("Concatenating train formulas...");
formula = tsc.GetTruthFormula(0, num_of_trains)
print("Total Formula completed!")
print(str(formula.size))

print("Now converting with TSEITIN...")
formula_t = formula.tseitin('z')
print("TSEITIN completed!")
print("Now checking for satisfiability...")
sat_result = formula_t.satisfy_one()
```

Finally, all functions are put together to form the conflict diagnosis. Note that `E` creates a new truth formula, removing the conflict set `delta` from `C` to obtain a network-wide hub type combination for the given conflict set. `E.satisfy_all()` retrieves a list of all solutions found for the current conflict set, which is seldomly the case in practice, but can happen nevertheless (see the exaple of timetables 17 and 22 in the practical application in section 7.2.3).

Note that `E.restrict(fixedHubTypes)` allows to fix certain hub types for a faster resolution process.

```
if (sat_result == None or len(sat_result) < 1):
  print("NO SOLUTION!")
  print("Now invoking FastDiag...")

  C = []
  for i in range(0, num_of_trains):
    C.append(i)
  list_of_deltas = []
  FastDiagFull(C, C, list_of_deltas, 0, tsc)

  print("There are " + str(len(list_of_deltas)) + " deltas")
  di = 1
  for delta in list_of_deltas:
    s = "Delta " + str(di)
    print(s + ", Trains:")
    di += 1
    for d in delta:
```

```
      print("  " + tsc.GetTrainName(d))
    print(s + " Solving...")
    E = And(*tsc.GetConstraintList(difference_lists(C, delta)))
    E = E.restrict(fixedHubTypes)
    all_sat_results = list(E.satisfy_all())


    if (len(all_sat_results) < 1):
      print("No Solution.")
    else:
      for sat_result in all_sat_results:
        print(s + " SAT Result " + str(i) + "/" + c + ":")
    print(s + " SAT Result :")
    if not sat_result:
      print("No result!")
    else:
        PrintSatResult(sat_result)
        i += 1
else:
  print("SAT Result:")
  sat_result_filtered = [{v: val for v, val in sat_result.items() if v.name != 'z'}]
  PrintSatResult(sat_result_filtered[0])
  print("Counting all solutions...")
  sat_count = formula_t.satisfy_count();
  print("Solution Count: " + str(sat_count))
```