M.Sc. Thesis
Master of Science in Engineering

**DTU Management Engineering**
Department of Management Engineering

# Optimization in Railway Timetabling for Regional and Intercity Trains in Zealand

## A case study of DSB

Bernardo Martin Iradi (s161075)

Kongens Lyngby 2018

**DTU Management Engineering**
**Department of Management Engineering**
**Technical University of Denmark**

Produktionstorvet
Building 426
2800 Kongens Lyngby, Denmark
Phone +45 4525 4800
braa@man.dtu.dk
www.man.dtu.dk

# Abstract

The Train Timetabling Problem is one of the main tactical problems in the railway planning process. Depending on the size of the network, the problem can be hard to solve directly and alternative methods should be studied.

In this thesis, the Train Timetabling Problem is formulated using a graph formulation that takes advantage of the symmetric timetabling strategy and assumed fixed running times between station. The problem is formulated for the morning rush hour period of the Regional and InterCity train network of Zealand.

The solution method implemented is based on a Large Neighborhood Search model that iteratively applies a *dive-and-cut-and-price* procedure. An LP relax version of the problem is solved using Column Generation considering only a subset of columns and constraints. Each column corresponds to the train paths of a line that are found by shortest paths in the graphs. Then, violated constraints are added by separation and an heuristic process is applied to help finding integer solutions. Last, the passengers are routed on the network based on the found timetable and the passenger travel time calculated. The process is repeated taking into account the best transfers from the solution found.

A parameter tuning is conducted to find the best algorithm setting. Then, the model is solved for different scenarios where the robustness and quality of the solution is analyzed. The model shows good performance in most of the scenarios being able to find good quality solutions relatively fast. The way the best transfers are considered between timetable solutions does not add significant value in terms of solution quality but could be useful from a planning perspective. In addition, most of the real-life conflicts are taken into account in the model but not all of them. As a result, the model can still be improved in order to provide completely conflict-free timetables.

In general, the model appears to be useful for the timetabling planning process of DSB. It allows to test different network requirements and preferences easily. The model not only generates a timetable but also estimates the passenger travel time and the occupancy of the trains quite accurately. Also, any modification in the line plan can easily be included without affecting the core model.

# Statement of originality

I hereby, declare that this thesis is my own work and that, to the best of my knowledge and belief, it contains no material which has been accepted or submitted for the award of any other degree or diploma.

I also declare that, to the best of my knowledge and belief, this thesis contains no material previously published or written by any other person except where due reference is made in the text of the thesis.

Kongens Lyngby, June 29, 2018

Bernardo Martin Iradi (s161075)

# Preface

This master's thesis was prepared at the department of Management Engineering at the Technical University of Denmark in fulfillment of the requirements for acquiring a MSc in Engineering degree in Industrial Engineering and Management. The workload of the thesis is 35 ECTS and it was conducted in the period 2$^{\text{nd}}$ of January 2018 - 29$^{\text{th}}$ of June 2018 under the supervision of Professor Stefan Røpke and Federico Farina, DTU Management Engineering, and Esben Linde, DSB Langsigtet Planlægning.

I would like to start by expressing my warmest gratitude to all my supervisors for all the help, dedication and patient guidance. Thanks to Stefan for all the enlightening comments and constructive discussions and for being always available to comment on any issue or idea I had. I wish to thank Federico for all the valuable feedback and the assistance with the circular graphs. Those plots would definitely not have looked nice without his help. I am very thankful to Esben, for introducing me to all the aspects of the railway planning process and for the valuable discussions about all parts of the project. I would also like to extend my thanks to the colleagues of the Langsigtet Planlægning department at DSB for being very attentive with me. A big thanks goes to Johanna for her endless support and constant boost of motivation. Thanks to Valentín and Gorka for their comradeship during all the hours in the library. Also, thanks to Filippo for his support and last minute proofreading. Finally, I wish to thank my family for being a constant source of motivation and optimism and encourage me to give my best.

Kongens Lyngby, June 29, 2018

Bernardo Martin Iradi (s161075)

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# CHAPTER 1

# Introduction

In 1825, the first railway line was opened in the UK between Stockton and Darlington (see painting by John Dobbin in Figure 1.1). This event marked the beginning of the railway age. In order to make a transportation system attractive to the potential passengers, the available services should be announced in advance and this is usually done by publishing a timetable. In 1830, the Liverpool and Manchester railway was opened becoming the first entire double-track line and the first one running according to a timetable. The benefits that this railway produced in the society were notable. As it is collected in the *Annual Register* of 1832, before the establishment of the railway, the transport mean between those cities was either by coach or by boat along the canal. The new railway allowed to carry twice as many daily passengers in less than half of the time compared to the coach transportation. It reduced transportation costs drastically and reduced the canal traffic by 30 %. This extraordinary success could not have been possible without the use of a timetable. George Bradshaw initiated in 1839 a series of railway timetables and travel guide books that became the first compilation of railway timetables in the United Kingdom. They were known as the *Bradshaw's guide* and became of great importance in the 19$^{\text{th}}$ century expanding later to other countries. Figure 1.2 shows an example timetable from 1850's *Bradshaw's guide*.

In Denmark, 22 years after the first worldwide railway line was opened in the

Figure 1.1: The Opening of the Stockton and Darlington Railway, 1825 (Dobbin, 1888)

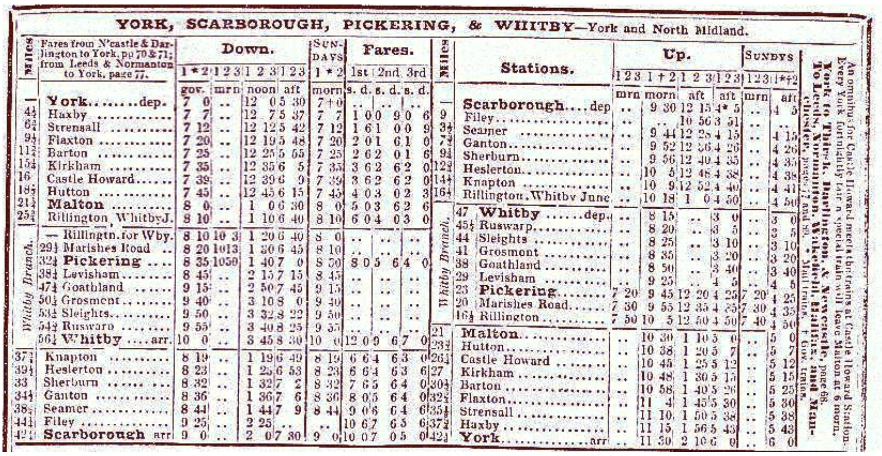


Figure 1.2: Timetable from the 1850 *Bradshaw's guide* (Bradshaw, 1850)

UK, in 1847, the first Danish railway line was constructed connecting the cities of Copenhagen and Roskilde. It was built by the privately owned company of *Det Sjællandske Jernbaneselskab* DSJ (lit.: The Zealandic Railway Company), that it was taken over by the state in 1880. A few years before, the state had already bought *Det Danske Jernbane-Driftsselskab* (i.e. the main operating company in Jutland and Funen) forming *De Jysk-Fynske Statsbaner*. In 1885, these two companies merged forming *De Danske Statsbaner* (DSB, 2018a), the main national railway operator that nowadays is still the largest passenger train operating company in Denmark.

Both the importance of having a good timetable in public transportation and the relevance of a public company such as DSB in the national railway sector form the motivational core of this master's thesis.

## 1.1 Motivation

The public transportation system of a country is a crucial part of the solution to the nation's economic, energy, and environmental challenges, helping to bring a better quality of life. The rapid growth of the modern cities asks for a reliable and efficient public transport system in order to counteract the incremental private transport usage. In Denmark, more than 40 % of its population currently lives in the archipelago of Zealand (StatBank, 2018) and it can be seen as a high transit area for daily commuters. One of the main transport systems used by people to travel around Zealand, apart from private ones, are both Regional and Intercity trains. These trains connect the different cities and towns of the archipelago through a multi-line network system. The travel distances and possible disturbances in this system are usually larger than the ones from a more urban railway system such as the metro and provide, in the same way, a greater margin of operation. Moreover, the process of finding a good timetable for the network system is currently being done manually to a large extent. Therefore, the study of improving the current timetable planning system for Regional and Intercity trains in Zealand by applying automatized techniques becomes an interesting case study. A reliable and efficient timetable for Regional and Intercity trains can help to provide a better and faster travel experience for commuters.

## 1.2 Aim of the thesis

This thesis addresses railway passenger optimization in the public transportation sector and it has been conducted in collaboration with Danske Statsbaner (DSB). Major improvements in public transportation are often a result of investments in infrastructure, rolling stock or new technology systems that allow faster transit and lower operational costs. However, there are other aspects of the public transportation that can be further improved and do not require expensive investments. Improving the timetables is one of them. As mentioned at the beginning of this chapter, a good timetable can lead to great benefits for both the passengers and the operating companies. This thesis aims at improving public transportation by focusing in the train timetable planning process. More specifically, the main goal of this project is to design an optimization tool that can be used in the timetable planning process of DSB. The tool consists in an optimization model able to generate timetables that optimize the train paths and passenger routing while considering different real-life constraints. The model is tested with DSB's data for a long-term scenario considering some, currently under construction, infrastructure improvements.

## 1.3 Thesis structure

This thesis is divided into nine main chapters.

Chapter 1 is a brief introduction to the thesis where the main motivations are stated as well as a historical overview of timetabling history and railway transport in Zealand.

Chapter 2 gives an overview of the whole railway planning process and the tasks that it decomposes into. It also explains how railway liberalization affected the danish railway planning system and introduces the main stakeholders of the danish railway system.

Chapter 3 covers the train timetabling process more specifically. The different characteristics of a timetable are explained as well as the methods to generate them that are listed through an extensive literature review. In addition, how the national railway timetable is planned according to European legislation is described and the planning process inside the danish largest operator DSB is also explained.

Chapter 4 describes the case study of this thesis, which information was collected and used in the study as well as the scope covered.

In Chapter 5 the train timetabling problem is formulated where each part of the problem is cautiosly described.

Chapter 6 describes the solution method used to solve the problem. Each of the steps in the algorithm and how they interact together are carefully explained.

Chapter 7 summarizes the computational results obtained from the different tests. First, a parameter tuning is done for the parameters affecting the algorithm behaviour based on three core instances. Later, different instance scenarios are tested with the best algorithm setting found.

In Chapter 8 an extensive discussion is conducted based on the results obtained. Here, the performance of the algorithm is analyzed as well as the advantages and limitations of it and which parts of it can be further improved.

Finally, Chapter 9 concludes giving an overview of the model implemented and the case study analyzed.

CHAPTER 2

# The railway planning process

The planning process of railway companies is very complex and is usually categorized into three main levels: *strategic*, *tactical* and *operational* (Bussieck et al., 1997). These levels conform a hierarchical process used as a decision-making tool where each of the levels includes different problems whose solution is used as an input for the problems at the subsequent level. The *strategic* level stands at the top level and takes care of the long-term planning problems such as defining the infrastructure of the network (Network Planning Problem (NPP)) or defining the lines and their frequencies along the network and their stopping patterns (Line Planning Problem (LPP)). Next, the subsequent level is the *tactical* level. The problems at this level cover the medium-term planning processes such as generating the timetables for the lines (Train Timetabling Problem (TTP)), allocating the trains to the different tracks in the network (Train Routing Problem (TRP)), or specifying the assignment of the rolling stock and crew to the trains (Rolling Stock Scheduling Problem (RSSP) and Crew Scheduling Problem (CSP)). Finally, there is the *operational* level. At this phase, the short-term planning processes are managed covering real-time operations such as train re-scheduling or delay management. Figure 2.1 depicts the main problems usually solved at each level and the information flow from one problem to another.

In this thesis, the focus is mainly on the tactical level of the planning process. As mentioned before, the strategic level provides a network and the lines that are supposed to run on the network. Then, the problems at the tactical level assume that the infrastructure is fixed and try to allocate the available resources in the most efficient way. From Figure 2.1, the first task of the tactical level is generating the timetables for the lines. This task consists in, for each train, determining the arrival and departure time at each of the stations the line visits. Chapter 3 makes a further explanation about this process.

## 2.1 Railway liberalization

During the 80's, the amount of passengers using railway transportation decreased noticeably in all Europe causing the railway companies to strongly depend on public

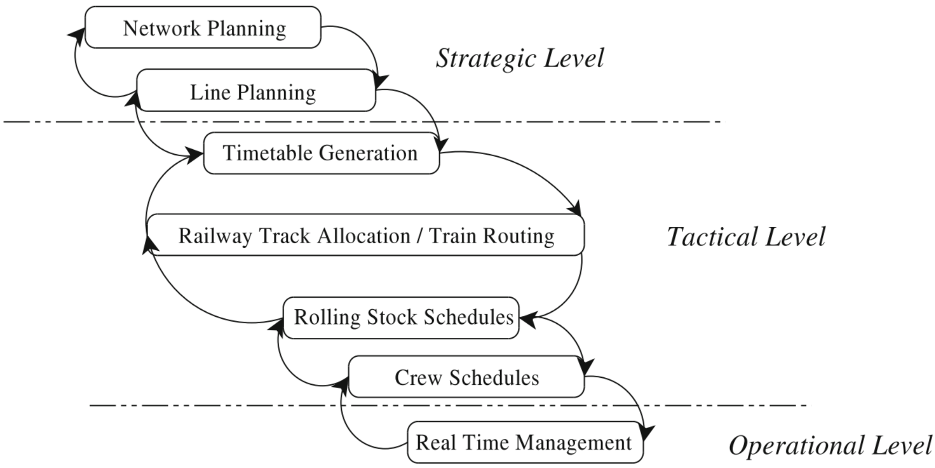


Figure 2.1: Railway Planning Process (Lusby et al., 2011)

funding (Alexandersson and Hulten, 2008). As a result, governments were under pressure to reform the national railway networks system. Both Sweden and UK were the pioneers in rail transport liberalization. In 1988, the Swedish government adopted the *Transport Policy Act* (Alexandersson and Hulten, 2008). It consisted in creating a govern-dependant Infrastructure Manager (IM) and separating it from the former incumbent monopoly train operator (SJ AB), that became a mere service operator that started paying for the infrastructure usage. This new legislation had a successful outcome, resulting in lower operation costs and, therefore, lower subsides, as well as a price reduction for using the operating lines (Alexandersson and Hulten, 2008). In the following years, other European countries started adopting similar measures and expanding the liberalization of the rail transport in the European Union (EU). Finally, in 1991, the European Comission (EC) announced the *91/440 Directive* (European Commission, 1991) that does not allow the operating activities and infrastructure manager to be part of the same company unless additional measures are taken to ensure independence.
The advantages of railway liberalizing are numerous. A recent publication by the European Comission (European Commission, 2013) concluded that the liberalized markets of Sweden and UK have improved in average, considering ten Key Performance Indicators that cover a broad scope of analysis such as quality or efficiency of the transport systems. It also showed that the performance difference between countries with a liberalized network and countries without. is significant in all studied aspects.

Regarding the Danish railway market, it is classified as a *Quasi-liberalised market*

(European Commission, 2013). This cluster considers that there is open access to the whole market but that there is no effective competition in it. There is a complete separation between infrastructure manager and operations (vertical separation). Moreover, passenger transport and freight transport are handled by different companies (horizontal separation). According to the Rail Liberalization index (LIB) developed by IBM (IBM, 2011), Denmark is among the leaders in railway liberalization progrees and it is considered *Advanced* in this aspect. According to Schittenhelm (2013), the current Danish railway timetabling process has been influenced by the liberalization of the European railway market.

### 2.1.1 Actors in the Danish railway transport planning

The full separation between Infrastructure Manager (IM), Train Operating Company (TOC) and government mentioned in section 2.1 creates a set of stakeholders for the Danish railway sector. All the stakeholders can be classified in four main groups. First, as government representative, the Danish Transport Authority (DTA) is the government agency in charge of regulating and planning public transport in Denmark. Next, the TOCs can be divided into freight TOCs or passenger TOCs that can also be private or state-owned companies. Finally, the IM is the one in charge of presenting the annual valid timetable. Figure 2.2 shows the main TOC using the Danish rail network. This thesis studies the main network of Zealand and part of Funen, therefore, only the TOCs that use part of the network could have the potential to affect a timetable. From the TOCs mentioned in Figure 2.2, both freight TOCs and DSB use the studied network. The corridor connecting Copenhagen's central station with the airport and Sweden is not considered and, therefore, SJ AB can also be discarded. Finally, it worth mentioning that DSB operates two different transport systems, one is the Regional/Intercity system and the second is the S-train system, the urban rail transport. The focus of this thesis is on the Regional/Intercity transport system.

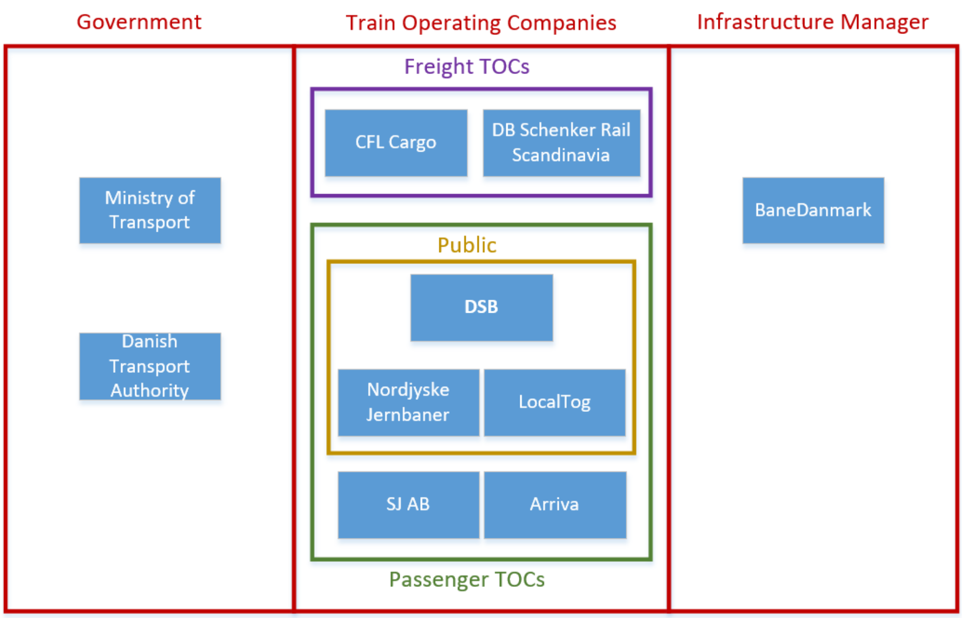


Figure 2.2: Main stakeholders in the Danish railway sector from Schittenhelm (2013)

# CHAPTER 3
# Train timetabling process

As passengers, when they think about a timetable the first thing they picture would probably be a table with different lines and the departure times from each station as the one in Figure 3.1. At first sight, constructing a timetable may look a rather simple task to do, nevertheless, the complexity behind a good timetable can be huge as there are different additional requirements to take into account that are hidden from the passenger's eye.

In the railway planning industry, timetables are depicted using the so-called *Time-Space Diagram*. These diagrams consider time in one of the axis and distance in the

**50 Odense - Ringsted - Roskilde - København**

| Køredage | | ①-⑤ | ①-⑤ | | | ①-⑤ | | | | ①-⑤ |
|---|---|---|---|---|---|---|---|---|---|---|
| Odense | | | 5.54 | | 6.09 | | | | 6.20 | |
| Langeskov | | | \| | | \| | | | | 6.28 | |
| Nyborg | | | 6.09 | | 6.26 | | | | 6.37 | |
| Korsør | | | 6.22 | | \| | | | | 6.50 | |
| Slagelse | | | 6.31 | | 6.47 | | | | 7.00 | |
| Sorø | | | 6.40 | | \| | | | | 7.09 | |
| Ringsted ○ | | | 6.48 | | \| | | | | 7.17 | |
| Ringsted | 6.37 | | 6.49 | | \| | 6.51 | | 7.06 | 7.18 | |
| Borup | \| | | \| | | \| | 7.00 | | \| | \| | |
| Viby Sjælland | \| | | \| | | \| | 7.05 | | \| | \| | |
| Roskilde ○ | 6.53 | | 7.04 | | \| | 7.12 | | 7.24 | 7.33 | |
| Roskilde | 6.54 | 6.59 | 7.05 | 7.06 | \| | 7.13 | 7.19 | 7.25 | 7.34 | 7.34 |
| Trekroner | \| | \| | \| | 7.09 | \| | 7.16 | \| | \| | \| | 7.38 |
| Hedehusene | 7.00 | \| | \| | \| | \| | 7.21 | \| | \| | \| | \| |
| Høje Taastrup ○ | 7.05 | 7.07 | 7.12 | 7.16 | 7.21 | 7.25 | 7.28 | 7.32 | 7.41 | 7.44 |
| Ny Ellebjerg ○ | \| | \| | \| | \| | \| | 7.36 | \| | \| | \| | \| |
| Valby ○ | 7.16 | \| | 7.22 | 7.26 | \| | \| | \| | 7.43 | 7.51 | 7.55 |
| København H ○ | 7.22 | 7.24 | 7.28 | 7.32 | 7.35 | 7.43 | 7.42 | 7.49 | 7.57 | 8.01 |
| Nørreport ○ | 7.26 | 7.29 | 7.37 | 7.40 | | 7.47 | 7.50 | 7.57 | | 8.06 |
| Østerport ○ | 7.31 | 7.34 | 7.42 | 7.45 | | 7.52 | 7.55 | 8.02 | | 8.11 |
| **Tognummer** | 4208 | 1208 | IC 808 | 2508 | Lyn 206 | 4108 | 3508 | 2208 | IC 108 | 4508 |

Figure 3.1: Published timetable example for the train lines going from Odense to Copenhagen through Ringsted and Roskilde during morning rush hour available for the passengers (DSB, 2018c)

other (i.e. the position of the train). The path of the train is represented through a line that defines the arrival and departure times at the stations along the way. This diagram is useful to check the track occupancy level and ensure that trains cross or overtake each other only at allowed points in the network. Figure 3.2 shows an example of a time-space diagram. The timetable represented in the diagram is the one of the corridor between Copenhagen central station and Kalundborg between 16:45 and 18:15.

## 3.1 Timetabling strategies

When planning a timetable, one of the first steps includes choosing a proper strategy. To do so, one should clarify the scope and desired structure of the timetable. In this case, four different timetabling strategies have been distinguished.

- **Non-cyclic Timetabling:** If each train path is scheduled individually, the structure of the timetable and dependencies between trains do not follow any clear pattern. This type of timetabling is categorized as *non-cyclic timetabling* or *individual trips timetabling* and it is mainly used where the train departures are different during the time period and/or the purpose is to maximize the track capacity by scheduling as many trains as possible where cancelling trains is allowed.

- **Cyclic Timetabling:** If the timetable is repeated over time and one train path of a line in one direction is scheduled, all the remaining paths for the line in that direction become determined. The time interval between two consecutive departures of the line is defined as the *period*. If more than one line is considered in

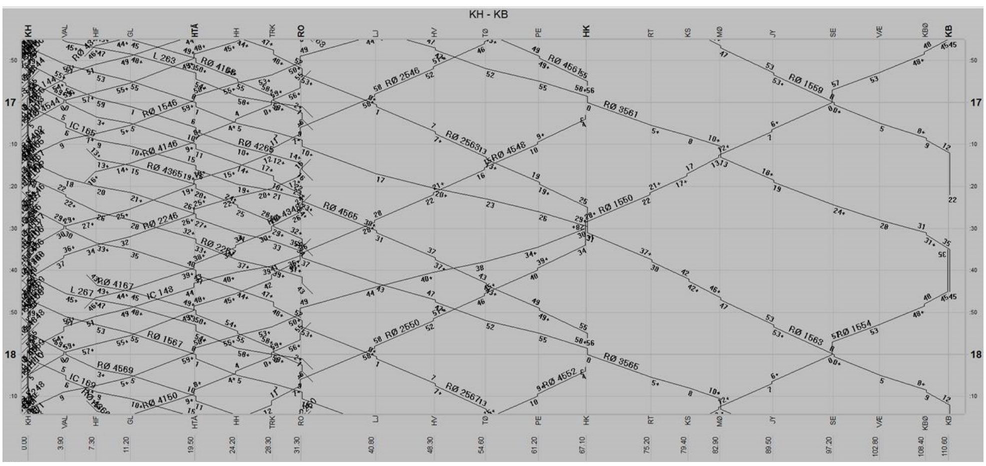


Figure 3.2: Timetable example represented as a time-space diagram (DSB, 2018b)

the study, including different trip frequencies, a common period that comprises all the lines should be chosen. This type of timetabling is categorized as *cyclic timetabling* or *periodic timetabling*. This timetabling strategy adds a more clear structure to the timetable and it is used more often in passenger rail transport than the non-cyclic one, as periodic timetables are easier to remember.

- **Symmetric Timetabling:** If the periodic timetable also shares a common symmetry axis, the timetable can be categorized as *symmetric timetable*. This means that if the train path of a line is scheduled, then, all the remaining paths of the line in both directions are automatically determined. This strategy can be used mainly when the running times of the line are identical in both directions. It has been shown by Liebchen (2004) that adding symmetry requirements in a Mixed Integer Problem (MIP) model for periodic timetabling allows to reach sub-optimal solutions faster. Moreover, in a symmetric timetable, any transfer between trains in one direction has the same transfer in the opposite direction with the same waiting time. This is highly appreciated from the passenger's point of view as it allows to do the same trip in the reversed way.
- **Integrated Fixed-Interval Timetabling:** This strategy defines immediate structural dependencies between different lines in some stations. At this stations, so-called *premium* transfers can be done. These transfers allow to change from any line in one direction to any line in the opposite direction in a very short period of time. This transfer places are known as *zero hub*. As it is mentioned next in Section 3.1.1, the Hour Model is a national railway timetable goal for Denmark that would like to resemble an IFIT timetable where the main cities (i.e. Copenhagen, Odense, Aarhus and Aalborg) operate as zero hubs for most of the trains with one hour travel times between them.

An application example of the four strategies is displayed in Figure 3.3. It can be seen that, for any cyclic timetable, if a train path is scheduled in one direction, it will cross a train from the same line travelling in the opposite direction exactly twice if the total path length is greater than the period time.

### 3.1.1 Hour Model

The Hour Model aims to introduce high-speed railway transportation in Denmark by decreasing travel time between main cities. As it is depicted in Section 3.4, the goal is to accomplish a travel time of one hour in Copenhagen-Odense, Odense-Aarhus, Odense-Esbjerg and Aarhus-Aalborg corridors and the timetabling strategy behind this target model seeks to resemble an Integrated Fixed-Interval Timetabling (IFIT).

Out of all this corridors, the new line between Copenhagen and Odense should to be first one to be completed. The new 250 km/h high-speed rail line between Copenhagen and Ringsted is expected to be finished in 2019 while the upgrade to 200 km/h of the existing line between Ringsted and Odense is not expected to be

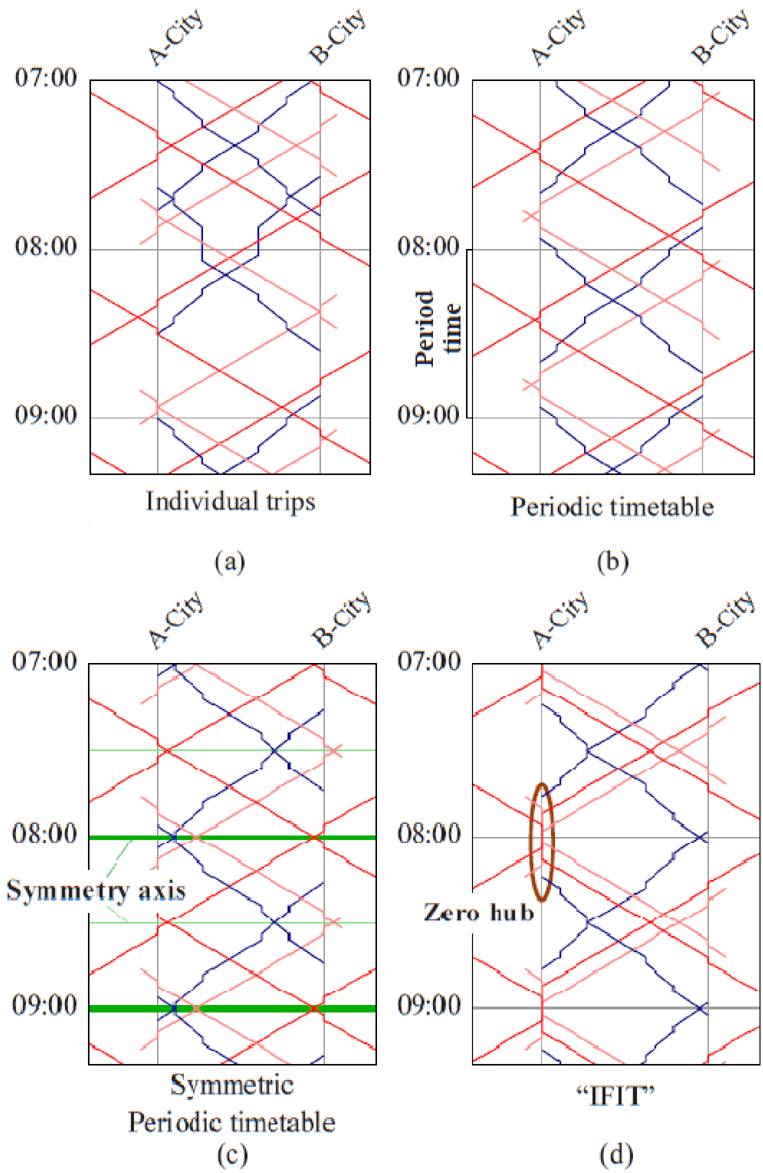


Figure 3.3: Time-space diagrams for a track showing the four timetabling strategies. (a) Non-cyclic timetabling, (b) cyclic timetabling, (c) symmetric timetabling and (d) integrated fix-interval timetabling. From Liebchen (2007)

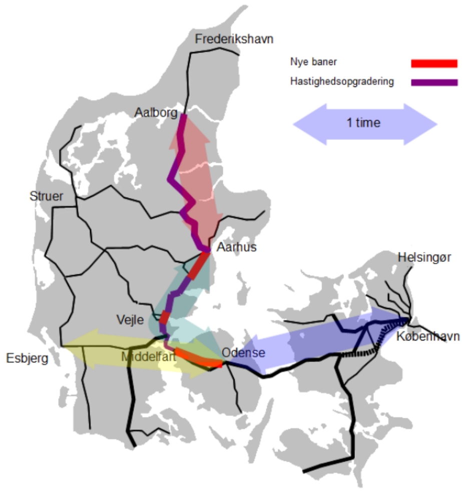


Figure 3.4: Hour Model travel time goals. The bidirectional arrows represent a one hour travel time segment (Transportministeriet, 2013)

completed before 2020. In Figure 3.5 the two mentioned lines are displayed. The rest of the infrastructure improvements shown in Figure 3.4 have a later expected date of completion and are not considered in this thesis.

## 3.2 Timetabling patterns

Most TOCs do not just plan a timetable for a specific period and repeat it along the entire day or week. Usually, the day (or the week) is divided into different time slots and a timetable is planned for each time slot. Each of the planned timetables corresponds to a pattern and can be combined together. Usually the differences between patterns consider starting/shunting different lines in the network and therefore, a small transition period is required between time slots with different patterns. Most commonly the following patterns are used along a weekday period:

- Morning rush hours
- Afternoon rush hours
- Day-time hours
- Evening hours
- Night hours

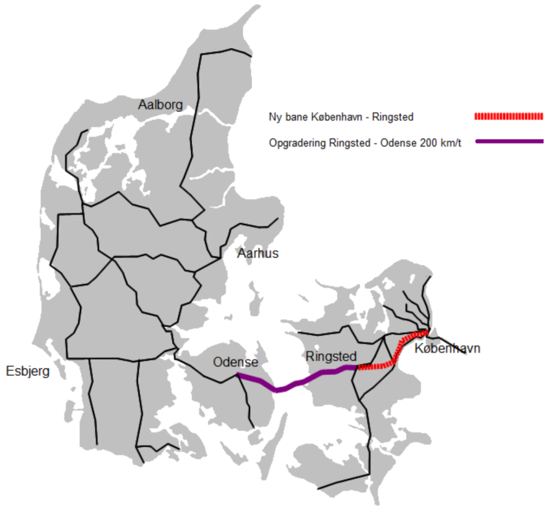


Figure 3.5: Routes of the, currently under construction, new lines in Zealand and Funen (Transportministeriet, 2013)

Usually, the morning and afternoon rush hours correspond to the time slots when most passengers travel and usually represent the time gap when people attend to and return from the working place respectively. Additional rush hour trains are usually scheduled in the direction to or from the working areas. Day-time hours are those covering the daylight time (usually from 5 am to 6 pm) that do not correspond to rush hours. The lines and frequencies operating at this time slot are similar to the ones of the rush hours but excluding the rush hour trains. Finally, evening hours (i.e. 6pm-12am) and night hours (12am-5am) are the time slots with less passenger flow and both time slots form the so-called *weak traffic time* (Liebchen, 2007).

This partition responds to the need to keep the passengers demand homogeneous at each time slot. The patterns do not need to have any similarity between them and they can differ in all aspects such as period or operating lines. However, passengers and operators prefer to have a *core* timetable with unalterable departure times. This means that is preferable to use a unique period time for all patterns and just remove or add a few lines between pattern, keeping the rest of the operating lines identical.

## 3.3 Train timetabling process in Denmark

As it was mentioned in Section 2.1, railway liberalization allows multiple Train Operating Company (TOC) to make use of the railway infrastructure in order to maximize the utility of the resources available. Therefore, railway liberalization creates different groups of interest that have common or conflicting interests. In this situation, creating a timetable that fully satisfies the interests of each TOC can be very difficult. The IM (BaneDanmark) is in charge of coordinating the national railway timetable and combining it with the ones of other European countries that may be affected. The timetable in Denmark is prepared following the European Union legislation and therefore follows the same process as the one from RailNetEurope (RNE) that is displayed in Figure 3.6. Its main goals are funded in political decisions that define the allocation of financial resources for planned investments. An example of a national timetable goal is the Hour Model explained in section 3.1.1 that aims to resemble an IFIT timetable. As it is depicted in Figure 3.6, the time span from the initial planning steps on the future timetable until the implementation day is up to 48 months. The timetable becomes effective the 2$^{\text{nd}}$ Saturday in December the year before the timetable actual year. The same date is used as a common date in all Europe in order to facilitate the timetable implementation.

Tenders for public service traffic are decided by the Danish Transport Authority (DTA). No railway tender project is alike and therefore, a close collaboration between the TOCs and the DTA is needed. The DTA needs to ensure that biding TOCs can create a feasible timetable. After TOCs present their bids, all proposals are evaluated by the DTA and external consultants. Finally, the chosen winning operators apply for infrastrcture capacity to the IM.

Each TOC needs to submit an application for infrastructure capacity with the IM. The deadline of this application determines the timetable planning process of each TOC. Afterwards, each TOC receives a preliminary timetable from the IM only including their respective train paths. Then, a margin of two months approximately is given to the TOCs to consider the preliminary national timetable. In the next step, all TOCs gather together with the IM to suggest changes and negotiate any potential

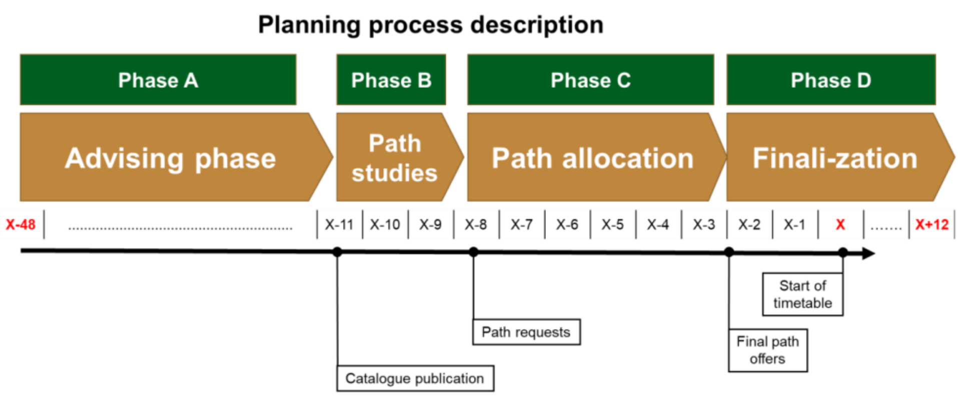


Figure 3.6: Main phases of the RNE general timetabling process (RailNetEurope, 2018)

conflict of the timetable. In the end, the final timetable is agreed and, after being assessed by a quality control, it is implemented in the railway system.

### 3.3.1 The timetabling process at DSB

The timetable planning process at DSB consists of five main phases and starts more than one year before of the timetable deployment date (Schittenhelm, 2013). The main steps of the process are the following:

1. Timetable ideas
2. Project timetable preparation
3. Detailed timetable preparation
4. Rolling stock preparation
5. Rostering plans for the crew preparation

The first step considers suggestions or proposals for minor adjustments in the existing timetable. The deadline for presenting ideas is usually approximately one and a half year before the timetable implementation date.

Next, larger changes are suggested to the existing timetable. This may consider, for example, travel time or train connections and are evaluated according to customers. This process is concluded roughly one year before the implementation date.

The following step assesses the project timetable for train path conflicts and measures its robustness. As a result, a detailed timetable is created where information from other TOCs and the IM is also taken into account. This process lasts approximately until eight months before the implementation date

The detailed timetable serve as a basis for estimating the rolling stock and turnaround times for the trains. This estimation is based on the occupancy levels calculated by manual passenger counting. This step needs to be concluded before the path requests deadline to the IM that is approximately 8 months before the implementation deadline (see Figure 2.1).

Finally, as a last step, the crew is scheduled for the trains as well as the maintenance service times. Crew scheduling is not strictly required in the path request from the IM and the deadline for presenting it can be relaxed a bit.

It can be noted, that steps 2-5 are highly dependent one on another and are done in parallel to a large extent although their deadlines are different.

## 3.4 Train timetabling generation

The process of generating a timetable for a given network of lines is formulated as the *Train Timetabling Problem* (TTP). Its main goal is, as mentioned before, to determine the arrival and departure times at the stations for each of the train lines in the network.

### 3.4.1 Train timetabling constraints

The departure and arrival times are subjected to multiple track capacity constraints and specific requirements from the railway operating company. An obvious example of a type of track capacity constraints is that two trains cannot be in the same track segment at the same time, whereas the requirements form the operating company can be very diverse (i.e. from forcing specific lines to synchronize at specific stations to spreading lines with similar stopping pattern along the timetable period). In order to avoid having two trains at the same track segment at the same time, a *headway* is defined. The headway refers to the minimum time interval between two consecutive train movements. The headway is defined by the signalling system along the track. These signals define the so-called *blocks* and enforce that only one train can be in a block at a time. Likewise, a headway may be defined for both departures and arrivals of consecutive trains along the same track segment. Moreover, dwelling restrictions may be applied, requiring the train to stop a minimum time interval at stations. A minimum dwell time is necessary to allow passengers to get in and out the train as well as changing drivers at some specific stations. In the same way, minimum running times between two stations may be enforced mainly due to the train speed, acceleration or breaking capabilities and track segment specifications. Moreover, always an additional buffer time is always considered to define the minimum running times called *timetable margin*.

### 3.4.2 Train timetabling objectives

Several objectives can be considered when creating a timetable. In general, these objectives can be classified in three main groups: Customer satisfaction, robustness and cost-efficiency. These objectives may be conflicting in most cases (see Figure 3.7). For instance, passengers would prefer to have always direct connections to their destinations at a high frequency, however, this would incur in an enormous

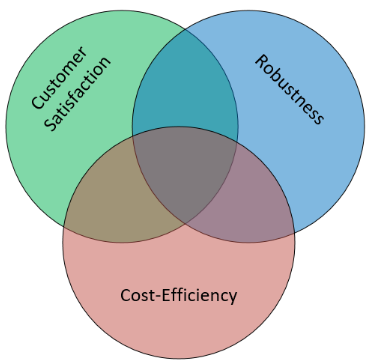


Figure 3.7: Main objectives of a train timetable

operational cost for the TOC, assuming a feasible timetable exists. Therefore, a compromise between conflicting objectives should be found.

From the passenger's point of view, minimizing the travel time or the transfer and waiting times at stations are good examples of objective functions. Also, the availability of seats and comfort at the train, or the ticket fares are factors that affect customer satisfaction.

On the other hand, train operators may be more interested in a robust timetable or timetables that do not have a high operational cost. A good example of objectives for a robust timetable can be maximizing headway between consecutive trains. For cost-effectiveness, minimizing rolling stock circulation and crew scheduling can be seen as attractive objectives that are directly related to the timetable.

## 3.5 Literature review

The literature about train scheduling is very extensive. The different publications apply a wide range of methods to different cases. Some of them consider just a corridor or a junction whereas other study a whole network. Moreover, the nature of the resulting timetable (i.e. cyclic or non-cyclic) also affects in the algorithm proposed. There are a few interesting publications that survey different TTP models (see Desrosiers et al. (1995), Cordeau et al. (1998), Caprara, Kroon, et al. (2007), Hansen (2009), Lusby et al. (2011), Cacchiani and Toth (2012) or Harrod (2012)).

Most of the studies that model a network assuming the periodicity of the timetable (cyclic timetable) are based on the Periodic Event Scheduling Problem (PESP) first introduced by Serafini and Ukovich (1989). Odijk (1996) proposes a cutting plane algorithm to solve the PESP and considers an Integer Linear Problem (ILP) formulation that is rather weak where integer variables are used to ensure the travel intervals are respected and continuous variables to determine the arrival and departure times modulo the period. Peeters (2003) and Liebchen (2008) propose a new ILP formulation where the integer variables are removed in exchange of a larger set of constraints. This formulation provides a stronger LP relaxation that leads to a significant speed up in the solution times. Given the effectiveness of the PESP, these type of models have been used to solve many network cases, whereas non-cyclic approaches are used more often to model single-line corridors or congested networks where it may not be possible to schedule all trains in an efficient way.

Szpigel (1973) presented one of the first ILP formulation for the non-cyclic TTP. The formulation is regarded as a job-shop scheduling problem where jobs (trains) need to be assigned to machines (track segments). Szpigel (1973) solved it using branch-and-bound applied to a Brazilian single-track line. Jovanovic and Harker (1991) proposed an Mixed Integer Linear Problem (MILP) formulation where the arrival/departure times are defined with continuous variables and the order of trains with binary variables and tries to find a reliable timetable. Cai and Goh (1994) designed a greedy constructive heuristic that allows to find conflict-free solutions to given single-track in a quick time but far from optimal. Carey and Lockwood (1995) proposed a mix of heuristic and branching procedure to solve a similar MILP as the one presented by Jovanovic and Harker (1991) in a one-way corridor, and Carey (1994) extended it to a two-way corridor showing that no additional constraints are needed. Higgins et al. (1997) tested different meta-heuristic algorithms such as Tabu search or genetic algorithms on a similar MILP.

Furthermore, Brannlund et al. (1998) introduced a pure ILP formulation where the time was discretized and therefore, the formulation could be represented as a graph where the nodes represent the arrival and departure time instants to each station. This new formulation is referred to as *time-space graph formulation* but cannot be directly applied to large instances due to the large amount of binary variables. As a result, further studying the LP relaxation of the model becomes more attractive and different methods have been developed based on it. The ILP formulation proposed by Caprara, Fischetti, et al. (2002) defines a variable for each node in the graph and it is solved using Lagrangian relaxation combined with sub-gradient optimization. Caprara, Monaci, et al. (2006)) extended the model to include different real-life constraints. Cacchiani, Caprara, et al. (2008) proposed a formulation where the variables refer to whole paths instead, and solved it applying column generation together with separation techniques. Cacchiani, Caprara, et al. (2010b) extended the formulation presented by Caprara, Fischetti, et al. (2002) to be applied in a network considering both passenger and freight trains and solved it using a similar procedure. Min et al. (2011) proposed a method for solving the train-conflict resolution problem with a column-generation based algorithm that takes advantage of the separability of the problem. Using an heuristic for the pricing problem, the method is able to get near optimal conflict-free solutions in a few seconds. Cacchiani, Caprara, et al. (2013) applied dynamic programming to solve the clique constraints that arise in the graph formulations and developed an exact method whose performance is compared with various heuristics in Cacchiani, Caprara, et al. (2010a). Robenek et al. (2014) proposed a method for competitive railway with different train operators. The ideal timetable is considered from each train operator and an overall weighted ideal timetable is calculated using the passenger demands that serves as input for the traditional TTP. Liu and Han (2017) showed, using a branch-and-price algorithm, that considering different types of headway, more trains can be scheduled in a one-way corridor. In general, most of the models proposed for solving non-cyclic timetables are used for scheduling multiple competing timetables from different operators.

The aforementioned studies are summarized in Table 3.1. Each row represents one study or method to solve the TTP and each column one characteristic of it. The first column states the authors of the proposed method in a chronological order. The second column specifies which type of infrastructure has been considered in the study; three main groups have been distinguished here: on one side there are studies covering a single line or *corridor* that can consist in a *single-track* where trains can travel in opposite directions or a *double-track* where each track is only used by trains traveling in one direction and, on the other side, there are studies covering a *network*. The group *network* comprehends in this case, for example, from a whole railway infrastructure of a country to a junction station where different tracks intertwine.

The third column specifies which mathematical problem is solved by the proposed method. The fourth column states the *nature* of the problem and is classified either as *cyclic* if it explicitly takes advantage of the periodicity or *non-cyclic* if it does not. The fifth column describes shortly the solution method used to solve the problem. The two main groups here include either exact methods (i.e. branch-and-bound) or heuristic methods. As a sixth column, the instance where the method has been tested is defined. The study case can be either a real-world case (i.e. part of a railway system of a country) or an theoretical example. Last but not least, the seventh column states which aspect of the timetable has been optimized. The goal can be, for example, to minimize travel or delay times or simply just generate a feasible timetable.

Table 3.1: Literature Review on methods applied to the Train Timetabling Problem

| **Authors** | **Infrastructure** | **Mathematical Problem** | **Nature of Timetable** | **Solution Method** | **Case Study** | **Objective Function** |
|---|---|---|---|---|---|---|
| Szpigel, 1973 | Single-track line | Job-shop | Non-cyclic | Branch-and-bound | Single track railroad in Brazil | Minimize Travel time |
| Serafini and Ukovich, 1989 | - | PESP | Cyclic | Implicit enumeration type algorithm | Illustrative example | Feasible solution |
| Jovanovic and Harker, 1991 | Single-track line | Ad hoc MILP | Non-cyclic | Branch-and-bound | Illustrative example | Maximize reliability |
| Cai and Goh, 1994 | Single-track line | Ad hoc IP | Non-cyclic | Greedy heuristic | Illustrative example | Minimize total cost of dwelling and delaying in passing loops |
| Carey, 1994 | Single-track line | Ad hoc MILP | Non-cyclic | Heuristic + Branch-and-bound | British and European rail lines type | Minimize deviation from ideal schedule |
| Carey and Lockwood, 1995 | Double-track line | Ad hoc MILP | Non-cyclic | Heuristic + Branch-and-bound | British and European rail lines type | Minimize deviation from ideal schedule |
| Odijk, 1996 | Network | PESP | Cyclic | Constraint Generation Algorithm | Arnhem CS station (Netherlands) | Feasible solution |
| Higgins et al., 1997 | Single-track line | Ad hoc MINLP | Non-cyclic | Local Search,Genetic Algorithms,Tabu Search and Hybrid Algorithms | Illustrative example | Minimize weighted travel time |

Continued on next page

**Table 3.1 – continued from previous page**

| Authors | Infrastructure | Mathematical Problem | Nature of Timetable | Solution Method | Case Study | Objective Function |
|---|---|---|---|---|---|---|
| Brannlund et al., 1998 | Single-track line | Time-space graph formulation | Non-cyclic | Lagrangian Relaxation | Swedish railway | Maximize profit |
| Caprara, Fischetti, et al., 2002 | Double-track line | Time-space graph formulation | Non-cyclic | Lagrangian heuristic algorithm | Rete Ferroviaria Italiana | Maximize the profit |
| Peeters, 2003 | Network | PESP | Cyclic | Many: based on the Cycle Periodicity Formulation | Dutch railway | Multi-objective |
| Caprara, Monaci, et al., 2006 | Double-track line | Time-space graph formulation | Non-cyclic | Lagrangian heuristic algorithm | Rete Ferroviaria Italiana | Maximize profits |
| Cacchiani, Caprara, et al., 2008 | Double-track line | Time-space graph formulation | Non-cyclic | Heuristic and exact algorithms based on column generation + separation | Rete Ferroviaria Italiana | Maximize profit |
| Liebchen, 2008 | Network | PESP | Cyclic | generic MIP solve | Berlin's underground network | Minimize the transfer and train idle time |
| Cacchiani, Caprara, et al., 2010a | Double-track line | Time-space graph formulation | Non-cyclic | Exact method and heuristics based on column generation | Rete Ferroviaria Italiana | Maximize profits |
| Cacchiani, Caprara, et al., 2010b | Network | Time-space graph formulation | Non-cyclic | Lagrangian heuristic algorithm | Austrian-Italian railway | Maximize profits |

Continued on next page

**Table 3.1 – continued from previous page**

| Authors | Infrastructure | Mathematical Problem | Nature of Timetable | Solution Method | Case Study | Objective Function |
|---|---|---|---|---|---|---|
| Min et al., 2011 | Network | Train-conflict resolution problem | Non-cyclic | Column generation + heuristic algorithm | Seoul metropolitan railway network | Minimize the total weighted deviation from original timetable |
| Robenek et al., 2014 | Network | Ideal Train Timetabling Problem | Non-cyclic | Generic MIP solve | - | Minimize the passenger cost |
| Liu and Han, 2017 | Double-track line | Time-space graph formulation | Non-cyclic | Branch-and-price | Chinese high-speed railway | Minimize the weighting sum of total dwell time and deviation of the earliest departure time |

CHAPTER 4

# Case study of DSB

The case study of this thesis aims at defining a timetable for a long-term network scenario of Zealand. More specifically, the scope covered corresponds to a period of one hour during morning rush hour.

## 4.1 Stations and lines notation

In the following chapters, the stations and lines are referenced using the identifiers and abbreviations listed in this section.

| Abbreviation | Station | Abbreviation | Station |
|---|---|---|---|
| **BO** | Borup | **NEL** | Ny Ellebjerg |
| **EK** | Eskilstrup | **NG** | Nyborg |
| **GZ** | Glumsø | **NF** | Nykøbing Falster |
| **HZ** | Haslev | **NÆ** | Næstved |
| **HH** | Hedehusene | **NÆN** | Næstved Nord |
| **HF** | Herfølge | **NV** | Nørre Alslev |
| **HK** | Holbæk | **OD** | Odense |
| **OL** | Holme-Olstrup | **RT** | Regstrup |
| **HV** | Hvalsø | **RG** | Ringsted |
| **HTA** | Høje Taastrup | **RO** | Roskilde |
| **JY** | Jyderup | **SG** | Slagelse |
| **KB** | Kalundborg | **SO** | Sorø |
| **KBØ** | Kalundborg Øst | **SE** | Svebølle |
| **KS** | Knabstrup | **TRK** | Trekroner |
| **KØ** | Korsør | **TH** | Tureby |
| **KH** | København H | **TØ** | Tølløse |
| **KJ** | Køge | **VAL** | Valby |
| **KJN** | Køge Nord | **VY** | Viby Sjælland |
| **LV** | Langeskov | **PE** | Vipperød |
| **LJ** | Lejre | **VO** | Vordingborg |
| **LU** | Lundby | **VÆ** | Værslev |
| **MØ** | Mørkøv | **ØLB** | Ølby |

| Line ID | Route | Type | Main corridor | Frequency (trains/hour) |
|---|---|---|---|---|
| **1xx** | Odense ↔ København H | InterCity | Old | 1 |
| **2xx*** | Odense ↔ København H | InterCityLyn | Old | 1 |
| **8xx** | Odense ↔ København H | InterCity | New | 1 |
| **12xx** | Nykøbing Falster ↔ København H | Regional | New | 1 |
| **15xx** | Kalundborg ↔ København H | Regional | Old | 1 |
| **22xx** | Nykøbing Falster ↔ København H | Regional | Old | 1 |
| **24xx/34xx** | Næstved ↔ København H | Regional | New | 2 |
| **25xx/45xx** | Holbæk ↔ København H | Regional | Old | 2 |
| **35xx*** | Kalundborg ↔ København H | Regional | Old | 1 |
| **41xx/43xx** | Ringsted ↔ København H | Regional | Old | 2 |
| **42xx*** | Nykøbing Falster ↔ København H | Regional | Old | 1 |
| **Lyn** | Odense ↔ København H | InterCityLyn | New | 1 |

* indicates that the line only operates during rush hours

## 4.2 The network

This network scenario covers the Regional, Intercity and IntercityLyn lines running in the entire Zealand excluding the *Kystbanen* (Coast line) that connects Copenhagen's central station with Helsingør and the *Kastrupbanen* (Kastrup line) that connects Copenhagen's central station with the airport and continues towards Malmö. The lines heading towards Jutland or Germany are studied until Odense in the west and Nykøbing Falster in the south. As mentioned in Section 3.1.1, the new corridor between Copenhagen's central station and Ringsted should be completed by 2019. Therefore, the new corridor is as well considered. Figure 4.1 shows the network considered in this case study.

The network shows 15 lines that cover 43 passenger stations. Each line in Figure 4.1 represents a line covered by one train per hour and direction, except the dashed lines that represent the rush hour lines that only run in one direction (i.e. towards København H if it is morning rush hour). In this case, the lines covering the same route has been considered a single line where the trains run strictly half an hour apart from each other. The network has been divided in six main corridors as shown in Figure 4.2 and will serve as a reference to the different parts of the network throughout the study.

The amount of tracks and the direction of trains running along them varies along each corridor. In order to be coherent throughout the whole report, three different types of track segment are defined. A track segment connecting two stations formed by only one track is referred as *single-track*. Along this type of track trains can circulate in both directions but there can only be one train on the segment at a time.

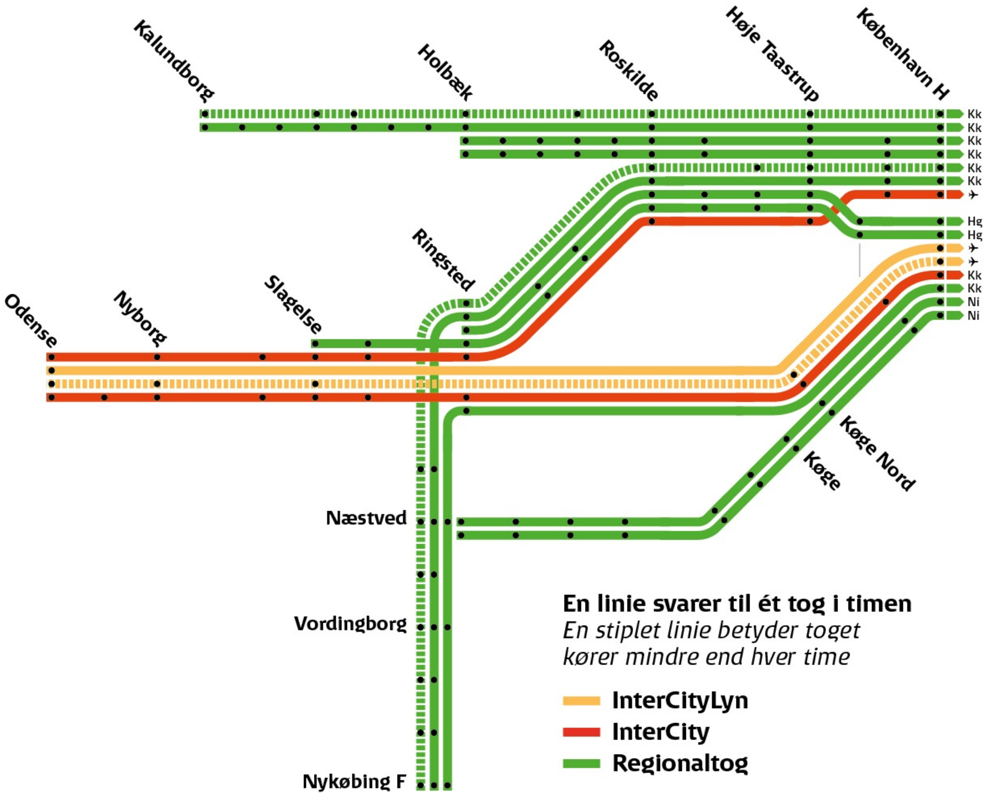


Figure 4.1: Network considered in the case study. Each line represents a frequency of one train per hour and direction and the dashed lines represent trains only running during rush hours (DSB, 2018b)

If two tracks connect two stations allowing trains to travel in both directions (one track per direction), it is denoted as *double-track*. Finally, if there are four tracks between two consecutive stations and trains can travel in both directions (two tracks per direction), it is denoted *quadruple-track*. These type of tracks allow two trains going in the same direction to overtake each other along the segment. These terms will be used in the following sections to refer to the different track segments connecting two stations along a corridor. In the network considered, there are two main single-track segments: the segment between Holbæk and Kalundborg and the segment connecting Køge Nord and Næstved along the Small-South corridor. Currently, part of the Large-South corridor is a single-track segment (i.e. between Vordingborg and Nykøbing F). However, the construction to upgrade that segment to a double-track is in progress and, therefore, the segment is considered as double-track. The rest of the network is

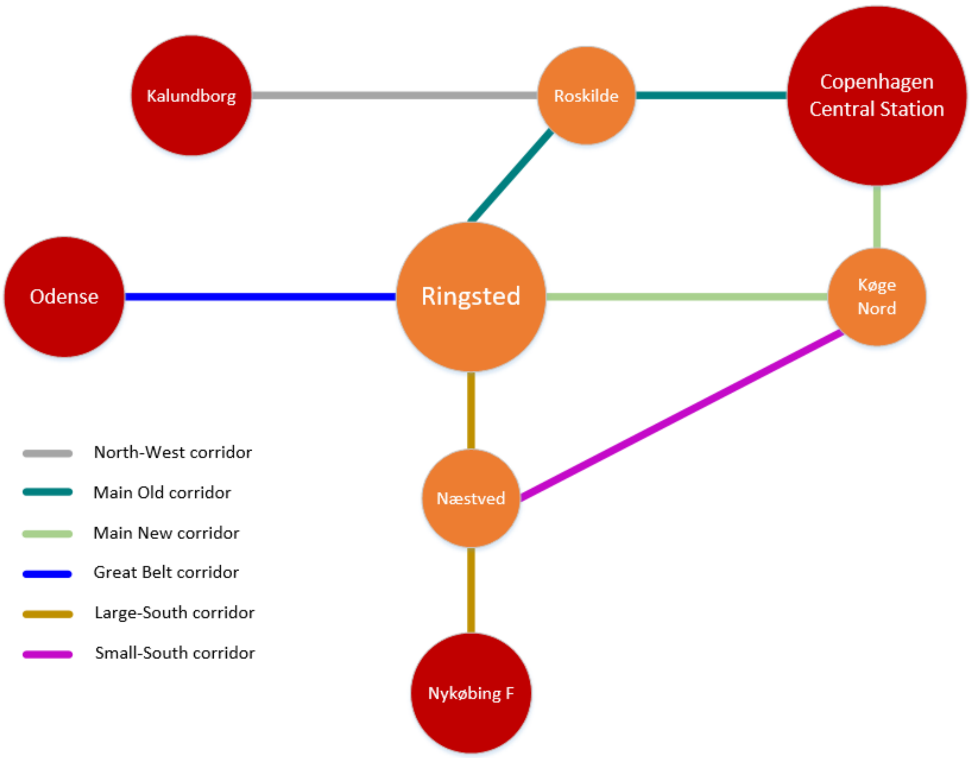


Figure 4.2: Division of the studied network in six corridors

connected by double-track segments with the exception of the segment between Høje Taastrup and Roskilde where there is a quadruple-track segment. Figure 4.3 displays the amount and type of track segments between the stations in the network.

## 4.3 Input data from DSB

In order to fulfill a realistic study, DSB provided different useful information to be used. Next, the different data parameters are listed.

- **Network lines:** All the lines forming the network as well as the stopping patterns are provided. The lines are running in both directions (towards Copenhagen and out of Copenhagen). However, a specific set of lines only run during rush hour and they only do it in one direction, towards Copenhagen if it is morning rush hour or out of Copenhagen if it is afternoon rush hour.

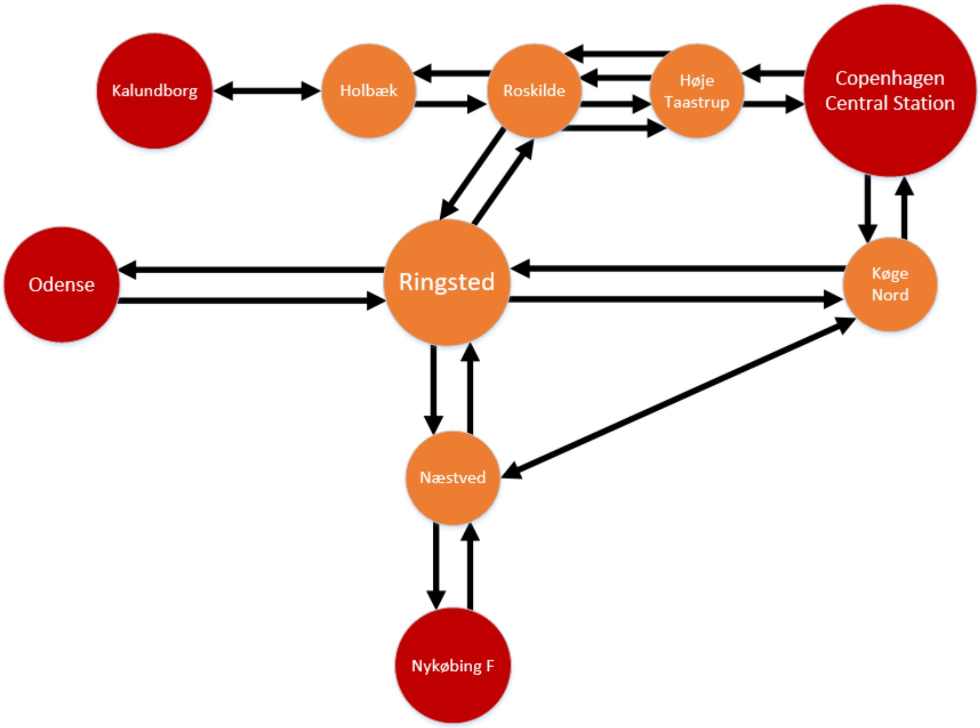


Figure 4.3: Number and type of track between specific main stations. Double or quadruple-track segments are depicted with mono-directional arrows whereas single-track segments are illustrated using bi-directional arrows.

- **Minimum running time:** This parameter states the minimum required time for a train to travel between two specific stations. This time interval is usually depending on the rolling stock type and the speed limits on the track segment. This value is given for every track segment connecting two consecutive stations in each line and direction.
- **Minimum dwelling time:** This parameter states the minimum required time for a train to dwell at a specific station. This time is usually the time required by the passengers to board and leave the train. This value is given for every station visited by each line and each direction.
- **Frequency lines:** There are specific pairs of lines that have similar or identical routes. It is a requirement from DSB's perspective to have those lines as separated as possible in the time-line. As an example, if both lines have a frequency of one train per hour, the departures along the same stations should ideally be half an hour apart from each other. There are three pair of these lines

considered in this case study. The normal and rush hour lines that travel the North-West corridor (København H-Kalundborg), the normal and rush hour line running between København H-Nykøbing Falster through the Main-Old corridor and the two IC lines connecting København H and Odense. The latter pair run through different corridors between København H and Ringsted (i.e. Main-Old and Main-New corridors) and the separation is only enforced in the segment they share (Ringsted-Odense).

- **Minimum headway between trains:** The headway is seen as the minimum time separation between two consecutive trains. In this case study, three different types of headway are considered. Headway between two consecutive departing trains in the same track segment and direction, headway between two consecutive arriving trains in the same track and direction and headway between two consecutive trains arriving from single-tracks in opposite directions.

- **Technical stations:** Some track segments between two consecutive stations may be too long. In the case of a train breakdown or disruption in the network, it is desirable to have an intermediate point where the train can either switch direction or track. Moreover, along the single-track track segments they might also be used to allow two trains to cross each other. These points are called technical stations or sidings. There are 20 of these stations along the network considered. However, 19 of them are located in double-track segments and only one in a single-track segment. In this case study, no disruption or breakdown of trains is considered and, therefore, only the technical station located in the single-track is taken into account in the study. This station is located between Kalundborg Øst station and Svebølle station and it is denoted as *Værslev* station.

- **Origin-Destination matrix:** This matrix defines the amount of passengers travelling between each pair of stations. In this case, the matrix reports an annual forecast done by a external company. For most stations, the historical data can help to make a more accurate forecast, however, this is not possible for the stations affected by the new corridor and the ones currently under construction.

- **Manual planned timetable:** The timetable planned manually by DSB is useful as a benchmark tool. The expected results can be compared and, although there are some modifications considered in this study, the solution can be used as a reference for the priorities and requirements of DSB.

- **Single-platform stations:** Some stations along the single-track segments have only one platform meaning that the station can only host one train at a time and a crossing between two trains is not allowed. In this case, there are two single-platforms in each single-track corridor, Knabstrup St. and Kalundborg Øst St. in the Holbæk-Kalundborg corridor and Ølby St. and Næstved Nord St. in the Køge Nord-Næstved corridor. It is assumed that, for the rest of stations in the network, any train arriving from an adjacent track segment has an available arriving platform.

## 4.4 Timetable evaluation by DSB

In order to measure the performance of a given process, companies usually use Key Performance Indicators (KPIs). Depending on the nature of the process and the interests of the company, the KPIs could be very different. Schittenhelm (2013) presented a extensive study about quantitative methods to asses railway timetables. He contacted different TOCs to survey the preferences and priorities each of them had when evaluating a timetable. According to Schittenhelm (2013), DSB criteria to evaluate a timetable is listed as follows:

1. Robustness of the timetable
2. Fast, frequent and direct connections
3. Scalability of the timetable

DSB committed itself, through the contract with the Danish Ministry of Transport, to achieve a customer punctuality level of 86.1 % by 2020 (Transportministeriet and DSB, 2015). This means that 86.1 % of all passengers should arrive on time. Both IM and DSB are responsible for achieving this level. In addition, DSB committed to achieve a train punctuality level of 94.4 % (Transportministeriet and DSB, 2015). This means that 94.4 % of all regional, IC and ICLyn trains must arrive on time. It was agreed that for a passenger or a train to be considered on-time, they should arrive with a delay of less than 3 minutes. In order to achieve this, DSB needs a realistic timetable that avoids small delays to be propagated along the whole network.

Another requirement from the government forces DSB to provide a minimum service level at all stations. This level is measured by the number of trains stopping at the given station per direction. Therefore, DSB needs to define a timetable that is able to comply with the contractual requirements and fulfill the minimum service levels at all stations.

Furthermore, a timetable that provides fast and efficient transfer at station is desirable for DSB in order to make it attractive to the potential passengers. From the passenger perspective, the need to do fewer transfers to reach their destination is highly appreciated.

Finally, the scalability of the timetable also needs to be taken into account. A timetable with good scalability allows to accommodate different scenarios without major changes. The scenarios can vary from increasing or decreasing the frequency of some lines to shorten or lengthen the route of a line.

All this aspects are crucial for defining a good timetable and have been taken into consideration when analyzing the results of this case study.

CHAPTER 5

# Problem formulation

This chapter covers the formulation of the mathematical problem. First, the notation is stated. Then, the graph formulation implemented is described. Next, an Integer Linear Problem (ILP) problem is formulated and the different constraints carefully explained. Finally, a brief comparison of the formulated problem with the previously presented work is done, highlighting the novel features of the modelling framework proposed.

## 5.1 Symbols and mathematical notation

In this section, the notation used for all the formulation presented in the chapter is summarized.

**Network**

| | |
|---|---|
| $S$ | set of stations in the network |
| $\hat{S}^e \subseteq S$ | set of stations that are endpoint of a single track segment |
| $A$ | set of double-track segments (arcs) |
| $E$ | set of single-track segments (edges) |
| $N = (S, E \cup A)$ | Network multi-graph |
| $\delta_N^+(i) \subseteq E \cup A$ | set of incident edges/arcs leaving station $i \in S$ |
| $\delta_N^-(i) \subseteq E \cup A$ | set of incident edges/arcs entering station $i \in S$ |
| $d(i, e)$ | minimum headway for departures in same direction for $(i, h) = e \in E \cup A$ |
| $a(i, e)$ | minimum headway at arrivals from same direction for $(h, i) = e \in E \cup A$ |
| $h(i)$ | minimum headway at arrivals from opposite directions for $i \in \hat{S}^e$ |

**Lines and trains**

| | |
|---|---|
| $L$ | set of lines |
| $D$ | set of possible directions of line (i.e. 2) |
| $F^l$ | frequency of the line $l \in L$ (trains/hour and direction) |
| $\Upsilon = L \times D$ | set of trains in a line and direction $L \times D$ |
| $T$ | set of time intervals in the cycle time |
| $d^j \in D$ | direction of train $j \in \Upsilon$ |
| $l^j \in L$ | line of train $j \in \Upsilon$ |
| $S^j \subseteq S$ | set of stations of train $j \in \Upsilon$ |
| $f_j$ | first station of train $j \in \Upsilon$ |
| $e_j$ | last station of train $j \in \Upsilon$ |
| $A^j \subseteq A \cup E$ | set of (directed) tracks for $j \in \Upsilon$ |
| $N^j = (S^j, A^j)$ | auxiliary network of train $j \in \Upsilon$ |
| $\phi_j(a)$ | running time in arc $a \in A^j$, $j \in \Upsilon$ |
| $\omega_j^{min}(i), \omega_j^{max}(i), \omega_j(i)$ | min, max and actual dwell time in station $i \in S^j$, $j \in \Upsilon$ |
| $\Xi$ | set of Frequency line pairs in the network |
| $T_s$ | minimum time interval between departures from Frequency lines |

**Graph formulation for train $j \in \Upsilon$**

| | |
|---|---|
| $V^l$ | set of nodes related to the graph of train $j \in \Upsilon$ |
| $V := V^1 \cup ... \cup V^L$ | the overall set of nodes |
| $R^j$ | set of arcs of the graph of train $j \in \Upsilon$ |
| $G^j = (V^j, R^j)$ | graph for train $j \in \Upsilon$ |
| $\sigma^j$ | artificial source node |
| $\tau^j$ | artificial sink node |
| $W_i^a$ | set of departure nodes from $i \in S^j \setminus \{e_j\}$ and arc $a = (i, h) \in A^j$ |
| $U_i^a$ | set of arrival nodes at $i \in S^j \setminus \{f_j\}$ and arc $a = (h, i) \in A^j$ |
| $\theta(v)$ | time instant associated with node $v \in V^j$ |
| $\Delta(u, w)$ | time interval between nodes $u \in V^j$ and $v \in V^j$ |

**Symmetric graph formulation for line $l \in L$**

| | |
|---|---|
| $\hat{V}^l$ | set of nodes related to the graph of line $l \in L$ |
| $\hat{V} := \hat{V}^1 \cup ... \cup \hat{V}^L$ | the overall set of nodes |
| $\hat{R}^j$ | set of arcs of the graph of line $l \in L$ |
| $\hat{G}^j = (\hat{V}^j, \hat{R}^j)$ | graph of line $l \in L$ |
| $\hat{\sigma}^l$ | artificial source node |
| $\hat{\tau}^l$ | artificial sink node |
| $\hat{V}_a^l$ | set of nodes for arc $a \in A^j$ where $j \in \Upsilon, l^j = l, d^j = 1$ |
| $K$ | maximum symmetry gap allowed in $\pm$ minutes |

**Paths and Line Paths**

| | |
|---|---|
| $P^j$ | Set of paths for train $j \in \Upsilon$ |
| $PT^p$ | Path length (time) of path $p \in P^j$ |
| $Q^l = \{Q^l_1, ..., Q^l_J\}$ | Set of line train paths for line $l \in L$ |
| $Q := Q^1 \cup ... \cup Q^L$ | Overall set of line train paths |
| $Q^l_v \subseteq Q^l$ | Subset of line train paths of $l \in L$ that visit node $v \in V$ |
| $Q_v = Q^1_v \cup ... \cup Q^L_v$ | Set of line train paths visiting node $v \in V$ |
| $c_q$ | "cost" of line paths $q \in Q$ |

**Decision variable of the original problem**

$\lambda_q \in \{0, 1\}$ 1 if line paths $q \in Q$ is chosen in solution, 0 otherwise

# 5.2 Railway network notation

The following network notation is stated for the network defined in Section 4.2. The notation is based on the one from Cacchiani, Caprara, et al., 2010b. Let $S = \{1, ..., s\}$ denote the set of stations in the network where each one is uniquely represented by a number of the set (i.e. 13 = København H (KH) and 24 = Kalundborg (KB)). Throughout the rest of the problem formulation, the stations will be referenced with their abbreviations (see Section 4.1). The network can be represented as a mixed multi-graph $N = (S, E \cup A)$ where each vertex $i \in S$ represents a station in the network and each edge $e = \{h, i\} \in E$ represents a single-track segment between two stations with no intermediate stations in between that is used by trains travelling in both directions (i.e. from $h$ to $i$ and from $i$ to $h$). Finally, each arc $a = \{h, i\} \in A$ represents a double-track segment between station $h$ and $i$ with no intermediate stations that can be used only by trains travelling in one direction (i.e. from $h$ to $i$). The reason why it is called a "multi-graph" is due to the fact that multiple arc/edges can connect the same two stations. For instance, there are four tracks between Høje Taastrup (HTA) and Roskilde (RO) (two in each direction. Therefore, the adjacent stations in between can be connected with four arcs (two in each direction) in the multi-graph. In the same way, if there is a single track, for example, between Holbæk (HK) and KB, and trains are travelling in both directions, the adjacent stations in between can be connected with an edge in the multi-graph. For convenience, for each station $i \in S$, let denote $\delta^+_N(i) \subseteq E \cup A$ the set of edges incident to $i$ and arcs leaving $i$, and $\delta^-_N(i) \subseteq E \cup A$ the set of edges incident to $i$ and arcs entering $i$.
Furthermore, for both mono and bi-directional tracks, minimum time intervals between departures/arrivals (i.e. headway) on the same track are required. Therefore, for each $e \in E \cup A$ and station $i$ of $e$, let denote:

- $d(i, e)$: minimum time interval between consecutive departures of trains travelling in the same direction from $i$ on the track segment $e$.

- $a(i, e)$: minimum time interval between consecutive arrivals of trains travelling in the same direction at $i$ on the track segment $e$.

Moreover, in the case of single-tracks, additional time interval requirements need to be set for trains travelling in opposite directions. Therefore, for each edge $e \in E$ and station $i$ of $e$ where $i \in \hat{S}^e$, let denote:

- $f(i,e)$: minimum time interval between an arrival at $i$ on $e$ and a departure from $i$ on $e$ of trains travelling in opposite directions.
- $g(i,e)$: minimum time interval between a departure from $i$ on $e$ and an arrival to $i$ on $e$ of trains travelling in opposite directions.
- $h(i)$: minimum time interval between an arrival to $i$ and an arrival to $i$ of trains travelling in opposite directions.

In this case study, due to safety requirements, a minimum value of $d(i,e)$, $a(i,e)$ and $h(i)$ is defined, whereas there is no minimum time for $f(i,e)$ and $g(i,e)$. However, the value of $g(i,e)$ is implicitly given by

$$\text{minimum travel time from } i \text{ to } h \text{ on } e + \text{minimum travel time from } h \text{ to } i \text{ on } e,$$

where $h$ is the other endpoint of $e$.

## 5.3 Lines and timetables notation

The different lines link two major stations with a number of intermediate stations in between. Let $L = \{1,...,l\}$ denote he number of operating lines in the network space (i.e.KH - HK) and $D = \{1,2\}$ the direction of the line, $D = 1$ for direction out of Copenhagen and $D = 2$ for direction towards Copenhagen. Let $\Upsilon$ be the set of trains that cover the $L$ lines and $D$ directions. For each train $j \in \Upsilon$ let denote $f_j$ the starting station and $e_j$ the ending station. Let $S^j := \{f_j,...,e_j\} \subseteq S$ be the ordered set of stations visited by train $j$ (stopping or not). Let $N^j = (S^j, A^j)$ be the auxiliary network for each train $j \in \Upsilon$ where each arc in $A^j$ is either an arc in A or an edge in E with an orientation, corresponding to the unique travel direction of $j$ along the single-track. A timetable for each train is given by the departure time at $f_j$ and the arrival time at $e_j$, and the arrival and departure times for the intermediate stations $f_j+1,...,e_j-1$. Let $\phi_j(a)$ denote the running time along arc $a \in A^j$ of train $j \in \Upsilon$. Let $\omega_j^{min}(i)$ denote the minimum dwell time at station $i$ for train $j \in \Upsilon$ where $i \in S^j \setminus \{f_j, e_j\}$. In the same way there is an upper bound in the dwell time (i.e. $\omega_j^{max}(i)$) in form of an additional percentage of the minimum dwell time ($\omega_j^{max}(i) \propto \omega_j^{min}(i)$). Note that, for a line containing N stations, there are N-1 minimum running times and N-2 minimum dwell times defined in one direction. The mentioned parameters above are defined for each train meaning that the station, running and dwell times sets are defined independently for trains in different directions for the same line, as they may differ. Finally, the time horizon is defined as $T = \{1,...,t\}$ referring to a whole hour discretized into time instants of half a minute ($|T| = 120$ time instants) and each line has an associated running frequency $F^l$ indicating how many trains per hour cover each direction of that line.

## 5.4 A graph representation

The problem can be defined using graphs to represent the possible timetables (from now on referred as *train paths*). A graph can be defined for each train $j \in \Upsilon$. Let $G^j = (V^j, R^j)$ be a directed and acyclic space-time graph in which the nodes represent the arrivals or departures at a station at a given time instant. The node set has the form

$$V^j = \{\sigma^j, \tau^j\} \cup \bigcup_{a=\{h,i\}\in A^j} (U_i^a \cup W_h^a)$$

where $\sigma^j$ and $\tau^j$ are the *artificial source node* and *artificial sink node* respectively and the sets $W_h^a$ for $h \in S^j \setminus \{e_j\}$ and $U_i^a$ for $i \in S^j \setminus \{f_j\}$ represent the set of time instants where a train can depart from or arrive to station $h$ or $i$ on the track represented by arc $a \in A^j$ respectively (also called *departure* and *arrival* nodes). Let $u, w \in V^j$ be nodes of the node set and let $\theta(u)$ be the time instant associated with node $u$. Furthermore, let $\Delta(u,w) := \theta(w) - \theta(u)$ denote the time interval between nodes $u$ and $w$ if $\theta(w) \geq \theta(u)$ and $\Delta(u,w) := \theta(w) - \theta(u) + T$ otherwise. Due to the cyclic nature of the time horizon $T$, it is said that node $u$ *precedes* or *coincides* with node $w$ (i.e. $u \preceq w$) if $\Delta(w,u) \geq \Delta(u,w)$ as it is assumed that all the time intervals used in this study case are far from the time horizon of one hour. Table 5.1 illustrates the time interval calculation with one example. For convenience, for each

Table 5.1: Example of the time interval calculation between two nodes with a cycle time $T = 60$

| $\theta(u)$ | $\theta(w)$ | $\Delta(u,w)$ |
|---|---|---|
| 10 | 15 | 5 |
| 15 | 10 | 55 |

station $i \in S^j$, let denote $\delta^+_{N^j}(i) \subseteq A^j$ the set of edges incident to $i$ and arcs leaving $i$, and $\delta^-_{N^j}(i) \subseteq A^j$ the set of edges incident to $i$ and arcs entering $i$.

The arc set $R^j$ for each graph can be defined by four main type of arcs.

- **Starting arc set**. These arcs connect the *artificial source node* with the set of nodes for the departure of first station in the line. There is one arc $r = (\sigma^j, w)$ for each $w \in (\bigcup_{a\in\delta^+_{N^j}(f_j)} W^a_{f_j} \cap V^j)$. These arcs have a null cost (free arcs).

- **Segment arc set**. These arcs connect the nodes related to the departure time from one station to the the nodes related to arrival time to the next station in the line (i.e. $r = (w, u)$) where

$$w \in (\bigcup_{a=(i,h)\in\delta^+_{N^j}(i)} W_i^a \cap V^j) \text{ and } u \in (\bigcup_{a=(i,h)\in\delta^-_{N^j}(h)} U_h^a \cap V^j)$$

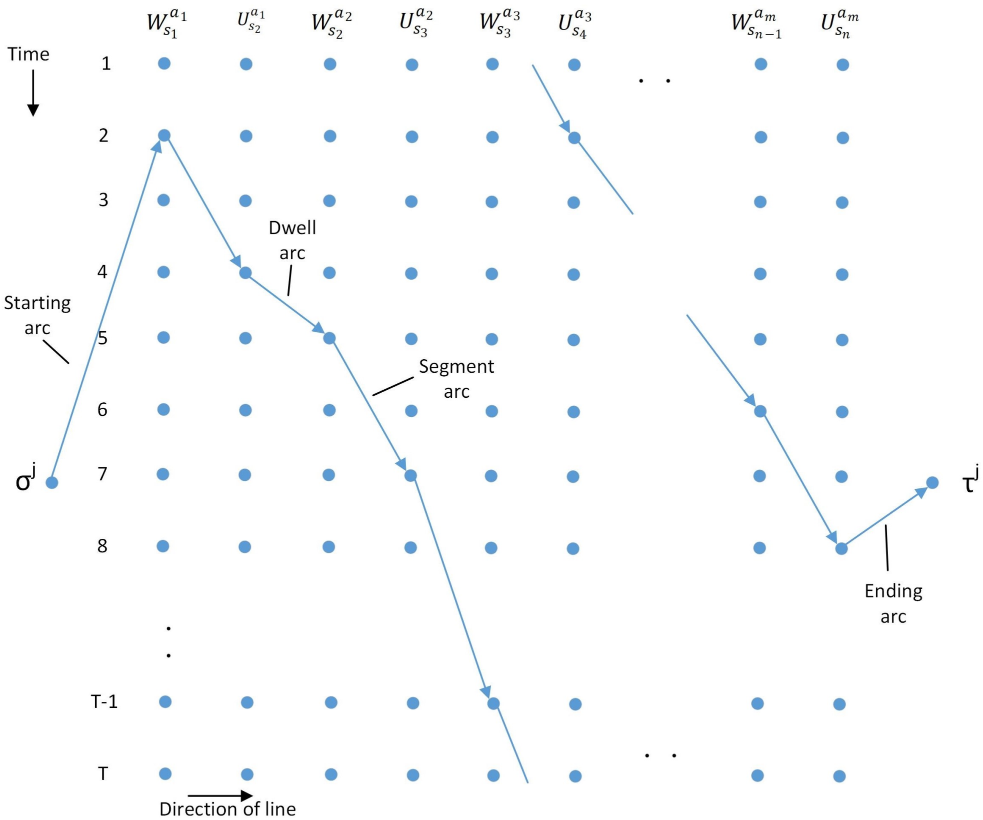


Figure 5.1: Graph representation of a train path

Furthermore, the arc needs to satisfy that $\Delta(w, u) = \phi_j(a)$ where $\phi_j(a)$ denote the travel time for arc $a \in A^j$. The cost of the arc corresponds to the *travel time* between the departure and arrival instants in the respective sets.

- **Dwell arc set**. These arcs connect the nodes related to the arrival time to one station with the the nodes related to departure time from the same station in the line (i.e. $r = (u, w)$) where

$$u \in \left( \bigcup_{a \in \delta^-_{N^j}(i)} U_i^a \cap V^j \right) \text{ and } w \in \left( \bigcup_{a \in \delta^+_{N^j}(i)} W_i^a \cap V^j \right) \text{ and}$$

Furthermore, the arc needs to satisfy that $\Delta(u, w) \in [\omega_j^{min}(i), ..., \omega_j^{max}(i)]$ for $i \in S^j \setminus \{f_j, e_j\}$. The cost of the arc corresponds to the *dwell time* between the arrival and departure instants in the respective sets. Let $\omega_j(i)$ denote the dwell time for the actual path at station $i \in S^j \setminus \{f_j, e_j\}$ (i.e. $\Delta(u, w)$).

- **Ending arc set**. These arcs connect the set of nodes of the arrival to the last station in the line with the *artificial sink node* . One arc $a = (u, \tau^j)$ for each $u \in (\bigcup_{a \in \delta^-_{N_j}(e_j)} U^a_{e_j} \cap V^j)$. These arcs have a null cost (free arcs).

Figure 5.1 shows an example of a train path represented using the a time-space graph. The timetable for train $j \in \Upsilon$ is defined by any path from the artificial source node $\sigma^j$ to the artificial sink node $\tau^j$. In the next chapters of this thesis these graphs will be referred as Train graphs.

## 5.4.1 Main assumptions

The final graph formulation presented in this thesis is based on the assumption that the travel time of each train along each track segment joining two stations is fixed. In other words, it is not possible to slow down the train along the track segment and, therefore, the departure time from one station uniquely determines the arrival time at the next station. Even if slowing down is something that has to be done at the operational level, this assumption is supported by the fact that, in practice, slowing down a train between two stations in most cases is equivalent to forcing the train to stop in an endpoint station of the track segment for a longer time and then to travel at the regular speed along the track. This statement is not true in general but it holds for realistic cases. In particular, experimental results performed by Caprara, Monaci, et al. (2006) show that the solution values found by heuristic procedures are marginally affected by this additional constraint, whereas the corresponding running time per iteration is widely reduced, since the graph $G$ turns out to be much smaller (for each train, the number of segment arcs between two stations is equal to the number of departure nodes). Furthermore, the above assumption simplifies the mathematical representation of the problem, yielding simpler and stronger overtaking or crossing constraints (see sections 5.5.3 and 5.5.4).

Another characteristic of the model assumed is the need for a symmetric timetable. When the train services are identical in both running directions it is easier to plan the timetable since the train path on one direction uniquely defines the path of the train in the opposite direction. Therefore, symmetric timetables are easier to plan and are more attractive to passengers as they provide equal transfer times in both directions. Nevertheless, this type of timetable reduces the degrees of freedom in the planning process and it is more suitable only when the passenger demands are similar in both directions.

As a result, these two main assumptions can lead to a new, more efficient, graph formulation. On one side, keeping the running times fixed reduces the number of nodes to the half since the arrival of a train is directly defined by the previous departure. On the other side, assuming symmetric paths for each line requires just creating one train path for a line, as the remaining line train paths are automatically defined. In a nutshell, ideally, the information of two Train graphs can be summarized in one graph formed by half of the nodes.

### 5.4.2 Symmetric Line graph

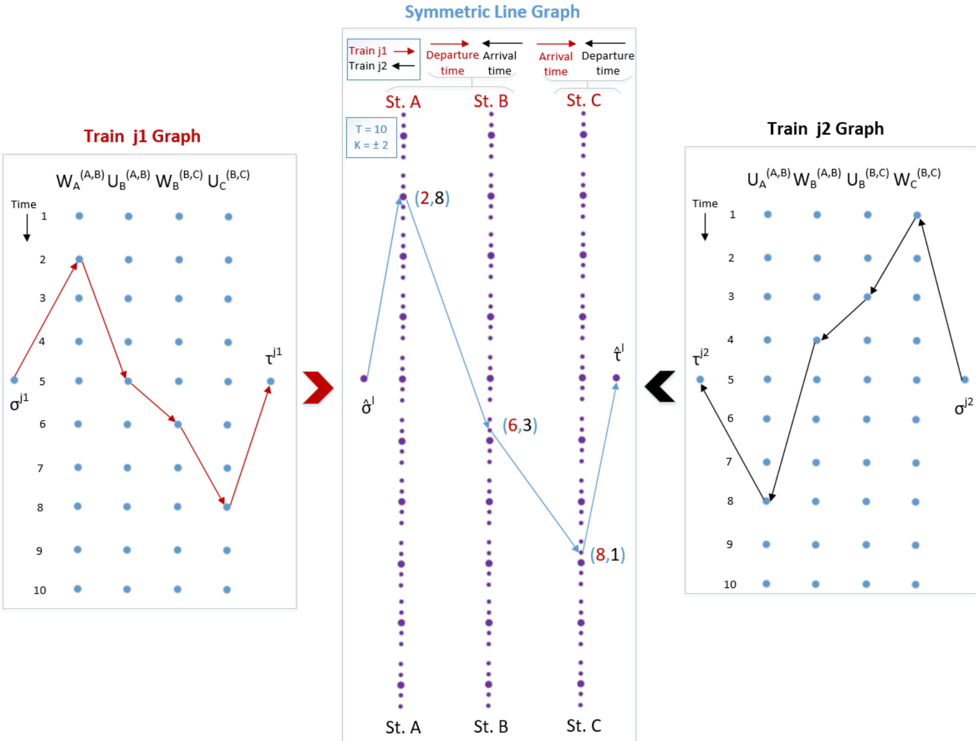


Figure 5.2: Representation of a path in the Symmetric Line graph as the combination of two paths in the respective Train graphs

The Symmetric Line graph formulation creates one graph for each line instead of one per train as stated initially, meaning that the number of graphs needed is equal to the number of lines considered. Ideally, each of the Symmetric Line graphs would include half of the nodes of one Train graph described initially (see section 5.4) due to the fixed running times and symmetric paths. Nevertheless, in practice, the running times of trains running in opposite directions along the same track segment are sometimes slightly different, meaning that two exactly symmetrical paths cannot be achieved. Therefore, a *maximum symmetry gap* $K$ is considered, meaning that the departure time of the train in one direction and the arrival time of the train in the other direction are considered *symmetrical* if the sum of both time instants sum up to a value that is within the range of the cycle time and the gap considered(i.e. $T \pm K$). A path in the Symmetric Line graph corresponds to two symmetric train paths in opposite directions as it can be seen in Figure 5.2. In this figure, the exactly symmetrical times at a station are depicted by larger nodes in the Symmetric Line graph and the

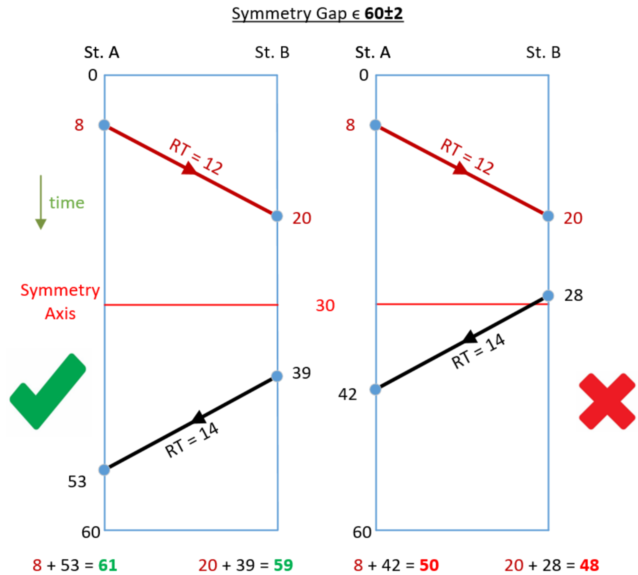


Figure 5.3: Example of line train paths in a track segment where $T = 60$ and a maximum symmetry gap $K \pm 2$ is allowed. The line train paths in the left are symmetrical in both ends of the segment whereas the line train paths in the right are not symmetrical in any end of the segment

symmetric instants that are within the gap considered $(K)$ are depicted with smaller nodes. In addition, Figure 5.3 shows an example of symmetric and non-symmetric train paths along a track segment.

A unique graph is created for each line $l \in L$. Let $\hat{G}^l = (\hat{V}^l, \hat{R}^l)$ be a directed and acyclic space-time graph in which each node represents the departure and arrival of two symmetrical train paths of the same line along a track segment. In other words, one node from the Symmetric Line graph notation is equivalent to four nodes of the Train graph notation. Let $w^i_j, u^h_j, w^h_k, u^i_k$ be these four nodes where, for a track segment $(i, h) = a \in A \cup E$, $w^i_j \in W^a_i \cap V^j, u^h_j \in U^a_h \cap V^j$ are the departure node from $i$ and arrival node to $h$ for train $j \in \Upsilon$ repectively and $w^h_k \in W^a_i \cap V^k, u^i_k \in U^a_i \cap V^k$ are the departure node from $h$ and arrival node to $i$ for train $k \in \Upsilon$ repectively, where $l^j = l^k$ and $d^j \neq d^k$. The four nodes belong to a single node in the Symmetric Line graph if $\theta(u^h_j) = \theta(w^i_j) + \phi_j(a)$ and $\theta(u^i_k) = \theta(w^h_k) + \phi_k(a)$ and $\theta(w^i_j), \theta(u^i_k)$ are symmetrical. Note that, symmetry is only checked in one of the stations of the arc because it is assumed that the running times of both trains along the same track are very similar (if not identical) and, therefore, a slight deviation in the symmetry in the other endpoint station can be rescheduled adapting the dwell time of one of the trains in accordance. Figure 5.4 shows, for a track segment, the nodes related to trains 1 and 2, belonging to the same line, that are equivalent to one node in the new

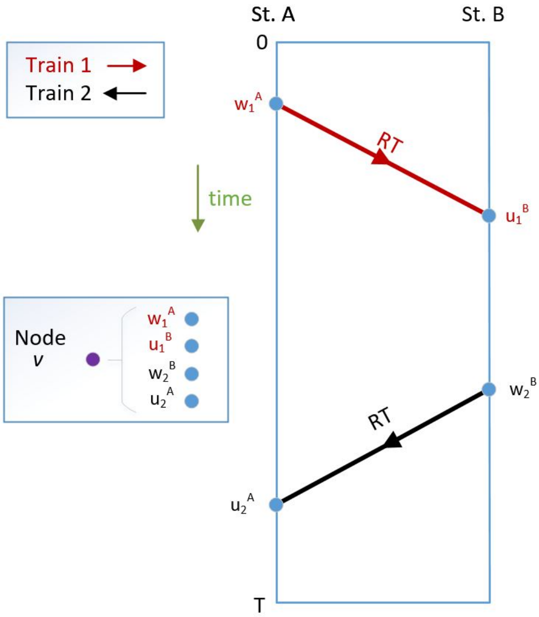


Figure 5.4: Representation of the Train graph nodes (blue) associated with one node of the Symmetric Line graph formulation (purple)

formulation.

Due to the symmetry gap allowed, each time instant in one direction has a range of symmetrical time instants in the opposite directions. As a result, the number of nodes increases proportionally to the range of symmetry allowed. Figure 5.5 shows a generic example of the distribution of the nodes for a line with trains running in both directions. In the example, the maximum symmetry gap is $K = \pm 2$ meaning that the range symmetry covers five time instants (i.e. $[T-K, ..., T+K]$), and therefore, the number of nodes in the graph is five times larger than the graph version imposing perfect symmetry between train paths. In addition, the station assigned to each node in Figure 5.5 represents the station of the segment where the symmetry is checked. This station is always the departing station for the train in direction out of KH (i.e. $D = 1$). Additionally, the symmetry is checked at both end stations of the line, meaning that a final set of trivial nodes is added to the graph that measure the arrival of the train in $D = 1$ to its last station and the equivalent departure from first station of the train in $D = 2$ (depicted as the last column of nodes in Figure 5.5).

The node set $\hat{V}^l$ has the following form for each $l \in L$:

$$\hat{V}^l = \{\hat{\sigma}^l, \hat{\tau}^l\} \cup \bigcup_{a \in A^j} \hat{V}^l_a \quad \text{where } j \in \Upsilon, l^j = l, d^j = 1$$

where $\hat{\sigma}^l$ and $\hat{\tau}^l$ are the *artificial source node* and *artificial sink node* respectively.

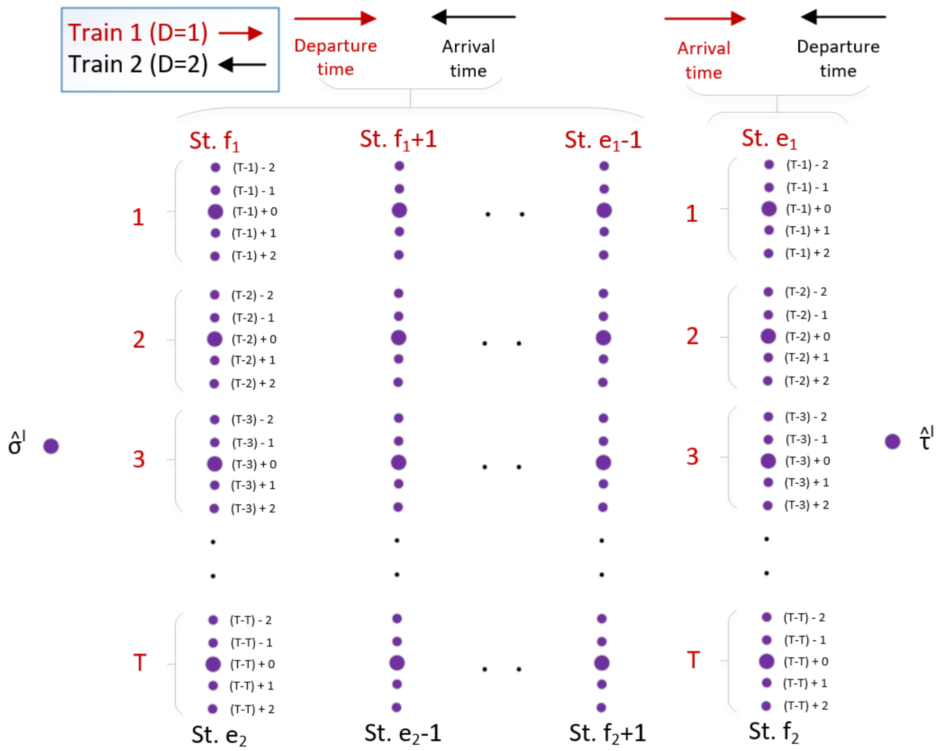


Figure 5.5: Graph representation with the node distribution for a normal line running in both directions where the larger dots represent exactly symmetrical departure/arriving times and the smaller ones represent symmetrical departure/arrivals within the stated gap (i.e. $\pm 2$ symmetry gap allowed in this example)

The subset $\hat{V}_a^l$ represents the nodes related to the track segment $a \in A^j$ where $j$ is a train from line $l$ in direction out of KH ($D = 1$). Relating to the initial type of graphs described, let trains $j, k \in \Upsilon$ where $l^j = l^k$ and $d^j \neq d^k$ be the two trains in opposite directions of the same line and a track segment $a = (i, h) \in A^j$. The set $\hat{V}_a^l$ is formed by the sets $\{(W_i^a \cup U_h^a) \cap V^j\}$ and $\{(W_h^a \cup U_i^a) \cap V^k\}$.

Regarding the arc set $\hat{R}^l$, the fact of assuming fixed running times allows to merge the *segment* and *dwell* arc in a single *segment+dwell* arc. The weight of this arcs is given by the sum of running time and dwell for both trains. Figure 5.6 shows an example illustration of what this type of arc represents in the actual line train paths.

The addition of a trivial set of nodes at the end of the line to check symmetry forces to have a redundant segment arc linking the nodes related to the segment covering this end station and the trivial set of nodes (i.e. the last two columns in Figure 5.5). The weight of these arcs is given by the sum of running time for both

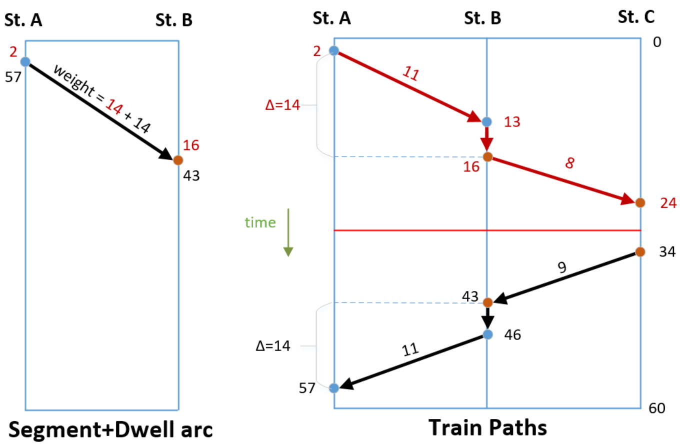


Figure 5.6: A Segment+Dwell arc example and the equivalent line train paths segments

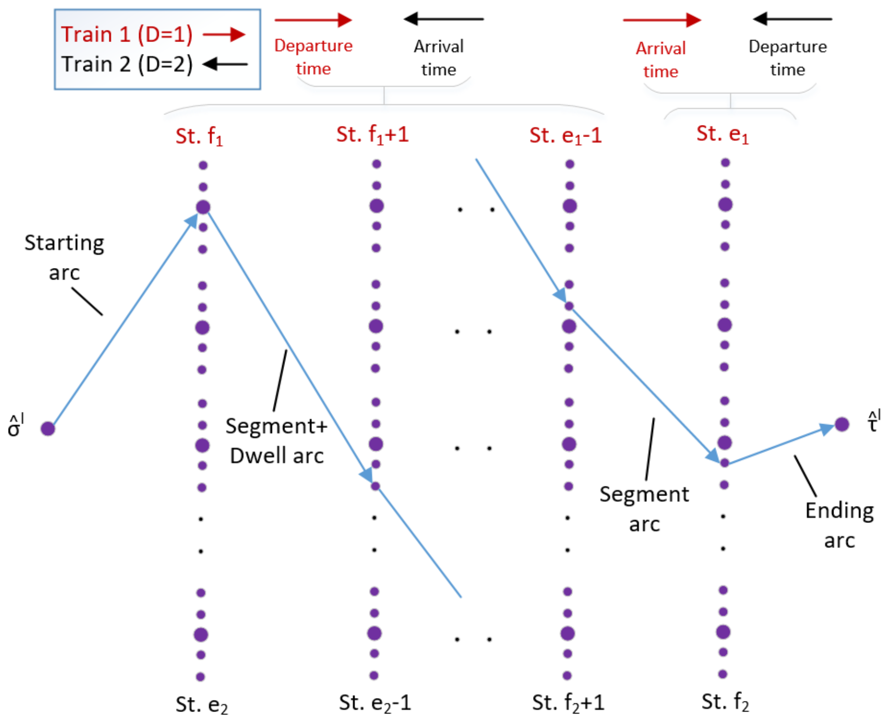


Figure 5.7: Graph representation of line train paths

trains. The *starting arc* and *ending arc* sets are used in the same way as in the train graphs.

Figure 5.7 shows the different arcs that compose a path representing the timetable of the line train paths.

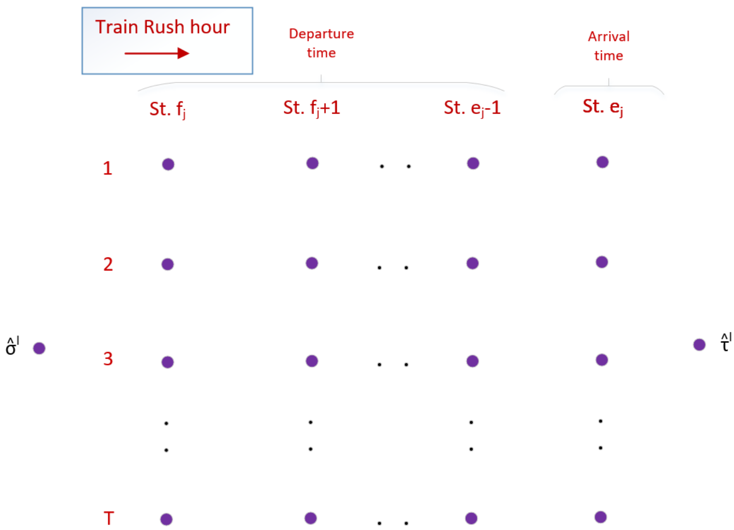


Figure 5.8: Graph representation with the node distribution for a rush hour line running in one direction where the blue dots represent the departure time instants from each station in the line and the arrival time instants to the last station in the line

In the rush hour lines, trains run only in one direction and, therefore, only one train path is needed. As a result, the symmetry requirements are not necessary in this case and the nodes only denote the time instant of the single train path. The resulting graph is almost the same as the Train graph but with half of the nodes because the assumption of fixed running time has already been applied. Figure 5.8 shows a generic example of the distribution of the nodes for a rush hour line with trains only running in one direction.

In the case that the frequency of the line (i.e. $F^l$) is two trains per hour and direction, the outlook of the graph does not get altered. As the train paths are exactly 30 minutes apart from each other, once a train path is defined in one direction, the rest of the frequency trains are uniquely defined. The only alteration affects the weight of the arcs. These do not sum only the time intervals of the two train paths but of the four of them.

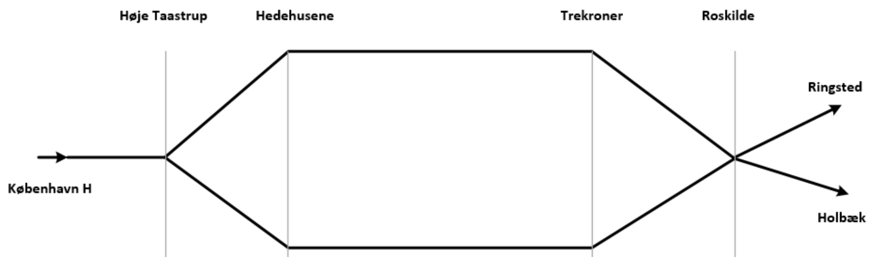


Figure 5.9: Illustration of the quadruple-track segment for the trains in direction out of KH

### 5.4.3 Quadruple-track management

As mentioned in Section 4.2, there is a quadruple-track segment between HTA and RO. This means that each train can choose between two tracks to travel along that track segment. It has been assumed that the train runs along the same track and cannot switch to the other track during the whole quadruple segment. Therefore, the formulation can easily be adapted adding an identical set of nodes for the quadruple-track segments and linking the sets in the stations where the track segment type changes (i.e. Høje Taastrup and Roskilde). Figure 5.9 shows that only two possible routes are allowed for a train travelling along the quadruple-track. With this requirement, trains travelling out of KH ($D = 1$) would need to choose which track they will use when arriving to HTA whereas trains travelling towards KH ($D = 2$) will need to take that decision when arriving to RO.

Only three track segments are quadruple-track, meaning that the number of nodes does not increase considerably. However, the combinations of train paths are double as an equal timetable can be found where the train chooses one or the other track segment.

## 5.5 ILP formulation

In this section the model is formulated as a Integer Linear Problem (ILP). In order to illustrate the different parts of the formulation, the notation of the Train graphs is used. As it is explained in Section 5.4.2, the set of nodes of the Symmetric Line graph are formed by combinations of node sets from the Train graph formulation.

### 5.5.1 Formulation without track capacity constraints

The problem can be formulated as a version of the Set Partitioning Problem (SPP) that aims to minimize the sum of total path lengths. The binary variable $\lambda_q \in \{0, 1\}$, $q \in Q$ defines if the group of line paths $q$ is included in the optimal solution. The parameter $c_q$ denotes the cost of choosing the group of line paths $q \in Q$ that is the sum of path lengths of the paths forming the line train paths group (i.e. $c_q = \sum_{p \in q} PT_p$). The formulation without the track capacity constraints is stated as follows:

$$min \quad \sum_{q \in Q} c_q \cdot \lambda_q \tag{5.1}$$

**s.t.**

$$\sum_{q \in Q^l} \lambda_q = 1 \qquad \forall l \in L \tag{5.2}$$

$$\lambda_q \in \{0, 1\} \tag{5.3}$$

The objective function minimizes the cost (path length) of the solution line train paths. Constraints 5.2 ensure that a group of line train paths is chosen to cover each line and constraints 5.3 state the binary property of the decision variable.

### 5.5.2 Headway constraints

One of the track capacity constraints deals with the minimum headway times between consecutive arrivals and departures at stations in the network.

$$\sum_{v \in U_i^a : v \preceq u, \Delta(v,u) < a_i} \sum_{q \in Q_v} \lambda_q \leq 1, \quad i \in S, a \in \delta_N^-(i), u \in U_i^a, \tag{5.4}$$

$$\sum_{v \in W_i^a : v \preceq w, \Delta(v,w) < d_i} \sum_{q \in Q_v} \lambda_q \leq 1, \quad i \in S, a \in \delta_N^+(i), w \in W_i^a, \tag{5.5}$$

$$\sum_{e \in \delta_N^-(i) \cap E} \sum_{v,u \in U_i^e : v \preceq u, \Delta(v,u) < h_i, \theta(u) = t} \sum_{q \in Q_v} \lambda_q \leq 1, \quad i \in \hat{S}^e, t \in T, \tag{5.6}$$

Constraints 5.4 and 5.5 enforce that the minimum headway distance between consecutive arrivals and departures respectively at each station, of trains in the same direction, is respected. Moreover, constraints 5.6 ensure that in the single-track segments the minimum headway between trains arriving in opposite directions is respected.

### 5.5.3 Overtaking constraints

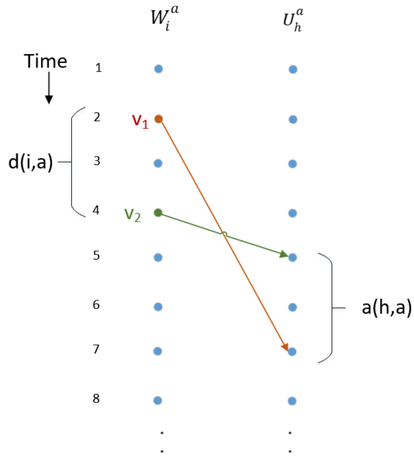


Figure 5.10: Illustration of a basic overtaking where $a(h, a) = 2$ and $d(i, a) = 2$.

It is not allowed that two trains travelling in the same direction on the same track overtake each other.

A basic example of an overtaking is shown in Figure 5.10 where both train departures are incompatible. The basic constraint corresponding to this overtaking would enforce that, at most, one orange train will depart from $v_1$ or one green train will depart from $v_2$. In this study, a stronger version of this basic constraint is formulated.

The following Constraints 5.7 are defined for every pair of trains $j, k$ along an edge/arc $a = (i, h)$ that is an arc in both auxiliary networks $N^j$ and $N^k$. Moreover, $j$ is considered the "slow" train and $k$ is the train that can actually overtake it. Therefore, the travel time of train $j$ should be greater than the one from train $k$ (i.e. $\phi^j(a) > \phi^k(a)$). For a constraint, we define an earliest possible departure from $i$ for trains $j$ and $k$. These departure nodes are denoted $v_1$ and $v_2$ respectively. Node $v_1 \in W_i^e \cap V^j$ and node $v_2 \in W_i^e \cap V^k$ correspond to departure nodes that are incompatible with each other (i.e. if train $j$ departs at $\theta(v_1)$, then train $k$ cannot depart at $\theta(v_2)$ and vice versa). The two trains $j, k$ are considered incompatible when either $\min\{\Delta(v_1, v_2), \Delta(v_2, v_1)\} < d(i, e)$, meaning that their departures are too close in time or $\min\{\Delta(u_1, u_2), \Delta(u_2, u_1)\} < a(i, e)$ where $u_1, u_2$ are the respective arrival nodes for $j, k$ corresponding to $v_1, v_2$, meaning that their arrivals to the next station are too close in time or $v_1 \prec v_2 \prec u_2 \prec u_1$ meaning that train $k$ overtakes train $j$ along the track.
Then, $v_3 \in W_i^e \cap V^{j,d}$ can be defined as the earliest possible departure of train $j$ that is compatible with $\theta(v_2)$ such that $v_1 \prec v_3$. Analogously, $v_4 \in W_i^e \cap V^{k,d}$ can be defined as the earliest possible departure of train $k$ that is compatible with $\theta(v_1)$ such

that $v_2 \prec v_4$. It can be seen that any departure of train $j$ from $[v_1, v_3)$ is incompatible with any departure of train $k$ from $[v_2, v_4)$.

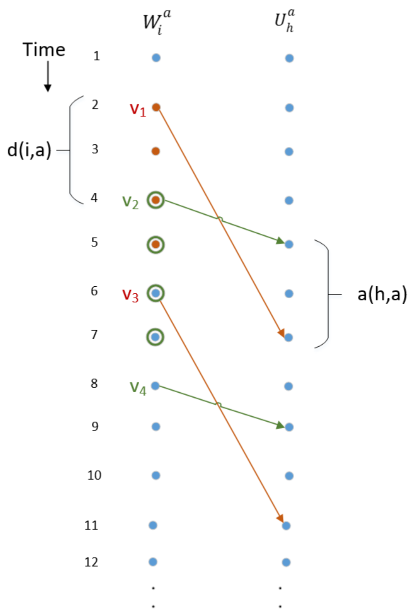


Figure 5.11: Illustration of an Overtaking constraint where $a(h, a) = 2$ and $d(i, a) = 2$.

In order to illustrate it more clearly, Figure 5.11 displays an example of an overtaking constraint. In the illustration, the departure nodes $v_1, v_2, v_3, v_4$ taking part in the constraint are shown. Note that in the illustration the minimum departure and arrival headways ($a(h, e)$ and $d(i, e)$) are respected for the trains at $v_1, v_2$ but they overtake each other along the track.

$$\sum_{w \in W_i^a \cap V^j : v_1 \preceq w \prec v_3} \sum_{q \in Q_w^{l^j}} \lambda_q + \sum_{w \in W_i^a \cap V^k : v_2 \preceq w \prec v_4} \sum_{q \in Q_w^{l^k}} \lambda_q \leq 1,$$
$$\forall j, k \in \Upsilon, v_1, v_2 \in W_i^a$$
$$(\text{where } l^j \neq l^k, d^j = d^k,\ i, h \in S^j \cap S^k, a = (i, h) \in (A^j \cap A^k)) \quad (5.7)$$

### 5.5.4 Crossing constraints

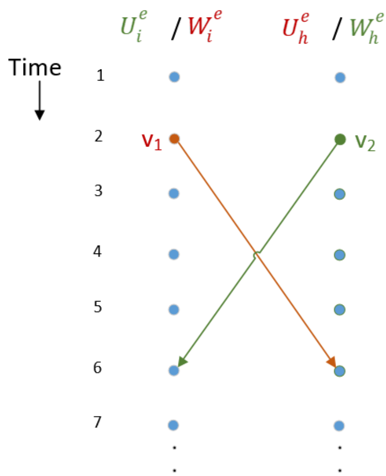


Figure 5.12: Illustration of a basic crossing

It is not allowed that two trains travelling in opposite directions are on the same single-track segment at the same time.

A basic example of a crossing is shown in Figure 5.12 where both departures are incompatible. The basic constraint corresponding to this crossing would enforce that, at most, one orange train will depart from $v_1$ or one green train will depart from $v_2$. In this study, a stronger version of this basic constraint is formulated.

The following Constraints 5.8 are defined in a similar way to Constraints 5.7. They are defined for every pair of trains $j, k$ travelling in opposite directions such that $e = (i, h)$ and $(h, i)$ are arcs in the auxiliary networks $N^j$ and $N^k$ respectively and correspond to the set of edges $E$ in the network. For a constraint, we define an earliest possible departure from $i$ and $h$ for trains $j$ and $k$ respectively. These departure nodes are denoted $v_1$ and $v_2$ respectively. Node $v_1 \in W_i^e \cap V^j$ and node $v_2 \in W_h^e \cap V^k$ correspond to departure nodes that are incompatible with each other (e.g. if train $j$ departs at $\theta(v_1)$, then train $k$ cannot depart at $\theta(v_2)$ and vice versa). The two trains $j, k$ are considered incompatible when either $u_2 \preceq v_1$ and $\Delta(u_2, v_1) < f(i, e)$ or $u_1 \preceq v_2$ and $\Delta(u_1, v_2) < f(i, e)$, meaning that arrival to and departure from same station are too close in time or $v_1 \prec u_2$ and $\prec v_2 \prec u_1$ meaning that train $j$ and train $k$ cross each other along the track.
Then, $v_3 \in W_i^e \cap V^j$ can be defined as the earliest possible departure of train $j$ that is compatible with $\theta(v_2)$ such that $v_1 \prec v_3$. Analogously, $v_4 \in W_h^e \cap V^k$ can be defined as the earliest possible departure of train $k$ that is compatible with $\theta(v_1)$ such that $v_2 \prec v_4$. It can be seen that any departure of train $j$ from $[v_1, v_3)$ is incompatible with any departure of train $k$ from $[v_2, v_4)$.

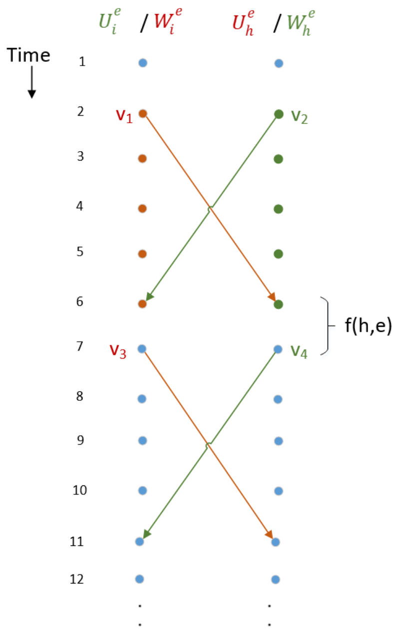


Figure 5.13: Illustration of a Crossing constraint where $f(h,e) = 1$

In order to illustrate it more clearly, Figure 5.13 displays an example of a Crossing constraint. In the illustration the departure nodes $v_1, v_2, v_3, v_4$ taking part in the constraint are shown. Also, for illustration purposes, the arrival and departure nodes at $i$ ($W_i^a$ and $U_i^a$) coincide in the figure, as well as the arrival and departure nodes at $j$ ($W_j^e$ and $U_j^e$).

Note that even if the minimum arrival headway ($f(h,e)$) is respected by the trains departing at $v_1, v_2$, they cross each other along the track.

$$\sum_{w \in W_i^e \cap V^j : v_1 \preceq w \prec v_3} \sum_{q \in Q_w^{l^j}} \lambda_q + \sum_{w \in W_h^e \cap V^k : v_2 \preceq w \prec v_4} \sum_{q \in Q_w^{l^k}} \lambda_q \leq 1,$$
$$\forall j, k \in \Upsilon, v_1 \in W_i^e, v_2 \in W_h^e$$
$$(\text{where } l^j \neq l^k, d^j \neq d^k,\ i, h \in S^j \cap S^k, e = (i,h) \in E^j, (h,i) \in E^k) \quad (5.8)$$

### 5.5.5 Frequency constraints

There are specific pairs of lines that share identical or similar first and last stations but have slightly different stopping patters. For instance, the line linking København H (KH) and Kalundborg (KB) during the day and the one linking the same stations during rush hour (see Figure 4.1) is a good example of the so-called *Frequency* lines. These pairs of lines should be spread along the cycle time as much as possible. In order to do so, the Frequency constraints behave in the same way as the departure headway constraints (5.5). Let $T_s$ denote the minimum time interval between consecutive departures of Frequency lines in one direction at each station. Finally let $\Xi := \{(m_1, n_1), ..., (m_k, m_k)\}$ denote the set of Frequency line pairs along the network where $m_k, n_k \in L$.

$$\sum_{v \in W_i^a : v \preceq w, \Delta(v,w) < K} \sum_{q \in \{Q_v^{l^j}, Q_v^{l^k}\}} \lambda_q \leq 1$$

$$\forall (l^j, l^k) \in \Xi, d \in D, w \in W_i^a$$

$$(\text{where } j,k \in \Upsilon, i \in S^j \cap S^k, a \in \delta_N^+(i) \cap (A^j \cap A^k) \quad (5.9)$$

Constraints 5.9 ensure that all the Frequency lines departures from any common station are spread at least a time interval of $T_s$ in each direction.

## 5.6 Comparison to existing models

The ILP and Symmetric Line graph formulation presented in this study are not created originally from scratch, and they are substantiated in different already existing formulations and models. The Train graph formulation described in Section 5.4 is based on the one presented by Caprara, Fischetti, et al. (2002). Moreover, Cacchiani, Caprara, et al. (2008) presents a formulation where the variables relate to whole paths and in which the formulation of this thesis is based on. Nevertheless, in this case, the formulation is extended to consider a group of line train paths per variable instead of just one train path. Additionally, Cacchiani, Caprara, et al. (2008) solves the problem using Column Generation and Separation techniques and it served as a basis for the core part of the solution method presented in this thesis. The studies aforementioned, however, modelled the problem only for a single corridor, meaning that the models were not applicable for a network like the one considered in this study. The network formulation and concept of crossing constraints has been inspired in the study presented by Cacchiani, Caprara, et al. (2010b).

All the studies that fundamentally affected the conception of the problem and solution method presented in this study, were originally created to model non-cyclic timetables. Unlike this study case, their goal is to create an specific timetable for a given time period. In this thesis, the goal of the resulting timetable is to be repeated over time taking advantage of the periodicity.

Another aspect that the formulation includes, is the symmetric timetabling strategy. As it is mentioned in Section 3.1, symmetric timetables increase customer satisfaction allowing identical transfers in both directions. The models in which the formulation is based on do not include the symmetry and it can be seen as a novel feature of the graph formulation proposed.

CHAPTER 6

# Solution method

This chapter covers the solution method used to solve the problem formulated in Chapter 5. First, an overview of the algorithm is given and next, the different steps of it are thoroughly described.

## 6.1 Overview

The implemented solution method is an iterative process that relies on a *dive-and-cut-and-price* procedure. The whole process runs with a time limit termination criterion and is summarized in Figure 6.1. The natural flow of the process is vertical, starting at the initialization of the problem and ending by returning the best encountered solution in the whole algorithm run. Nevertheless, between the main steps in the process, there are loops that should be taken into account. As mentioned, it runs with a time limit termination criteria, meaning that the whole procedure iterates while the time limit is not reached and the number of iterations can variate due to the complexity of the problem instance.

First, the problem is initialized. This consists in initializing the graphs for each of the lines as well as initializing the relaxation of the original ILP formulation considering just a subset of variables and the Headway constraints known as the Reduced Master Problem (RMP). Once this is done, the algorithm starts iterating through the innermost loop. This first loop concerns the Column Generation procedure. While the shortest path of any of the graphs provides a set of *line train paths* with a negative reduced cost, the process keeps iterating and adding columns to the RMP. This step is covered in Section 6.2.

When the Column Generation procedure is done, the next loop begins. This outer loop covers the separation procedure of the set of track capacity constraints not considered initially in the RMP (i.e. Overtaking, Crossing and Frequency constraints). If the LP solution violates any of the not included constraints, these are added to the RMP and the process iterates until the solution does not violate any track capacity constraint. At this point, the solution can be considered LP optimal. This step is further explained in Section 6.3.

Next, the heuristic loop begins and it iterates while the found solution is not integer. At this step, for a fractional solution, a fractionally used node is found among all the graphs and the node is enforced to be used as part of the final solution. This method is known as *dive heuristic* In this way, the solution space is reduced and

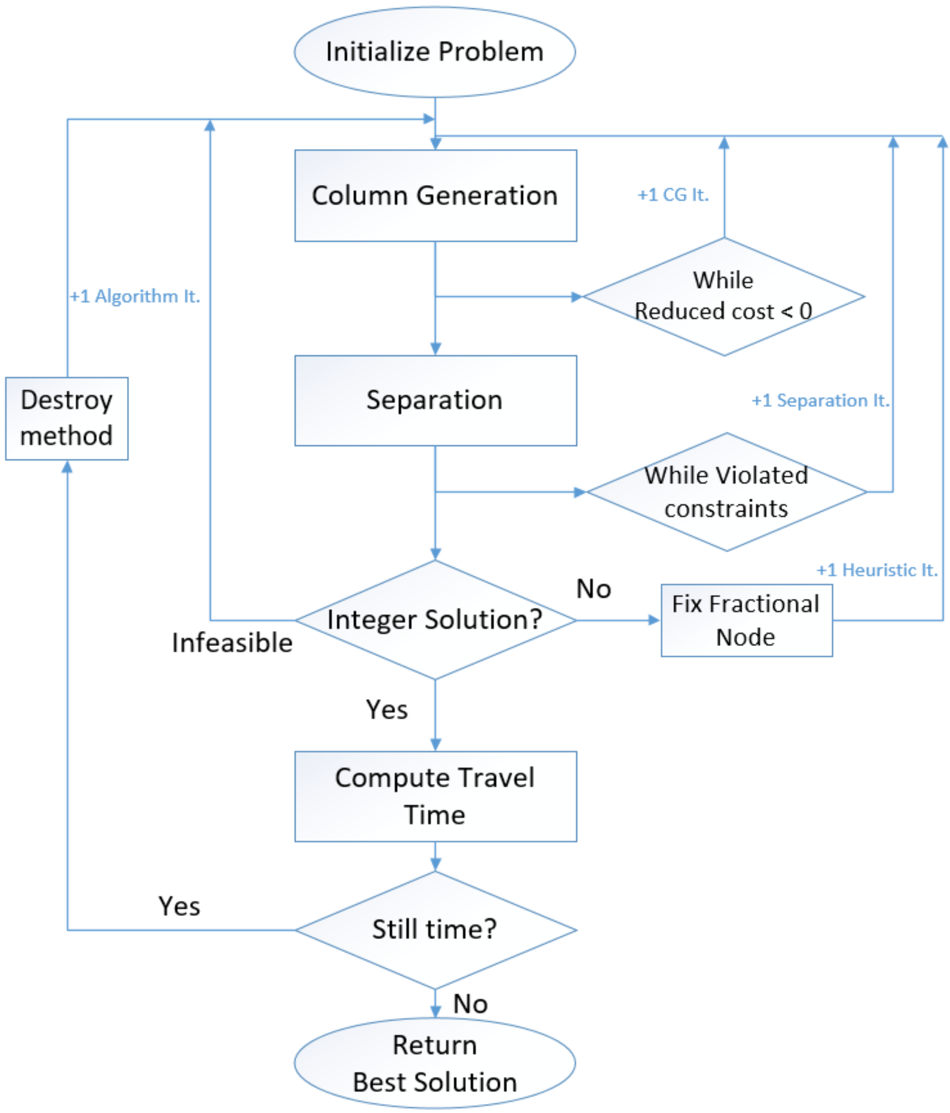


Figure 6.1: Algorithm diagram

avoids terminating the procedure with a fractional solution. However, this procedure has two different outcomes: either the solution is integer or it becomes infeasible. This step is explained in detail in Section 6.4.

When an integer feasible solution is reached, the total passenger travel time is calculated by computing the quickest routes for the passengers. This step splits in two main steps where, first, the best routes are found for the passengers by a shortest path heuristic and afterwards the Passenger Travel Time (PTT) is computed using an Origin-Destination (OD) matrix reflecting the passenger demand. Section 6.5 explain both parts in detail.

Finally, the outermost loop is reached that is performed while the time limit is not reached. It destroys the solution found by finding the most promising transfers and favouring their train connections in the following algorithm iteration procedure. Every time a solution is computed, it is compared with the best solution found so far and updated if the new one is better. This step is described in Section 6.6.

When the time limit is reached, the best solution found is returned, from which the whole timetable can be extracted as well as the occupancy levels of the trains along their paths and the passenger flow at the stations.

## 6.2 Column Generation procedure

Taking into account the cycle time, the size of the network and the symmetry gap allowed, the number of possible line train paths to be considered is extremely large. In order to handle that amount of variables efficiently, Column Generation (CG) techniques are necessary.

The column generation procedure relies on the *Dantzig-Wolfe decomposition* (Dantzig and Wolfe, 1960) where the original problem is relaxed and divided into one Master Problem (MP) and one or multiple Pricing Problems (PPs). The decomposition results in an stronger LP relaxation and a MP that typically has more variables and less constraints. For real life instances, the amount of variables is sometimes too large to solve directly and they should be added dynamically using *Column Generation*. Therefore, a reduced version of the MP is initially considered known as the Reduced Master Problem (RMP) that includes only a subset of the variables. These initial variables can just be a set of "dummy" artificial variables that satisfy the constraints of the RMP. For each line $l \in L$ a pricing problem is created (i.e. $PP^l$) that is in charge of providing line paths objects ($q \in Q^l$) that can potentially improve the current solution. Next, the formulation of both RMP and PPs are defined.

### 6.2.1 Reduced Master Problem

The formulation of the RMP is identical to the one of the original problem except for the relaxed version of the decision variable. The formulation is stated again in order to relate to the dual variables of each constraint introduced at the end of the section.

$$min \quad \sum_{q \in Q} c_q \cdot \lambda_q \tag{6.1}$$

**s.t.**

**Convexity Constraints**

$$\sum_{q \in Q^l} \lambda_q = 1 \quad \forall l \in L \tag{6.2}$$

**Headway Constraints**

$$\sum_{v \in U_i^a : v \preceq u, \Delta(v,u) < a_i} \sum_{q \in Q_v} \lambda_q \leq 1, \quad i \in S, a \in \delta_N^-(i), u \in U_i^a, \tag{6.3}$$

$$\sum_{v \in W_i^a : v \preceq w, \Delta(v,w) < d_i} \sum_{q \in Q_v} \lambda_q \leq 1, \quad i \in S, a \in \delta_N^+(i), w \in W_i^a, \tag{6.4}$$

$$\sum_{e \in \delta_N^-(i) \cap E} \sum_{v, u \in U_i^e : v \preceq u, \Delta(v,u) < h_i, \theta(u) = t} \sum_{q \in Q_v} \lambda_q \leq 1, \quad i \in \hat{S}^e, t \in T, \tag{6.5}$$

**Overtaking Constraints**

$$\sum_{w \in W_i^a \cap V^j : v_1 \preceq w \prec v_3} \sum_{q \in Q_w^{lj}} \lambda_q + \sum_{w \in W_i^a \cap V^k : v_2 \preceq w \prec v_4} \sum_{q \in Q_w^{lk}} \lambda_q \leq 1,$$
$$\forall j, k \in \Upsilon, v_1, v_2 \in W_i^a$$
$$(\text{where } l^j \neq l^k, d^j = d^k, \ i, h \in S^j \cap S^k, a = (i,h) \in (A^j \cap A^k)) \tag{6.6}$$

**Crossing Constraints**

$$\sum_{w \in W_i^e \cap V^j : v_1 \preceq w \prec v_3} \sum_{q \in Q_w^{lj}} \lambda_q + \sum_{w \in W_h^e \cap V^k : v_2 \preceq w \prec v_4} \sum_{q \in Q_w^{lk}} \lambda_q \leq 1,$$
$$\forall j, k \in \Upsilon, v_1 \in W_i^e, v_2 \in W_h^e$$
$$(\text{where } l^j \neq l^k, d^j \neq d^k, \ i, h \in S^j \cap S^k, e = (i,h) \in E^j, (h,i) \in E^k) \tag{6.7}$$

**Frequency Constraints**

$$\sum_{v \in W_i^a : v \preceq w, \Delta(v,w) < K} \sum_{q \in \{Q_v^{lj}, Q_v^{lk}\}} \lambda_q \leq 1$$
$$\forall (l^j, l^k) \in \Xi, d \in D, w \in W_i^a$$
$$(\text{where } j, k \in \Upsilon, i \in S^j \cap S^k, a \in \delta_N^+(i) \cap (A^j \cap A^k) \tag{6.8}$$

**Variable Linearity**

$$\lambda_q \geq 0 \tag{6.9}$$

Although initially only constraints (6.2)-(6.5) and (6.9) are considered, the separation procedures can potentially add any of constraints (6.6)- (6.8) and therefore are also included in the formulation.
Let the dual variables associated with constraints (6.2)-(6.8) be the following:

- Dual variables of constraints (6.2) $\rightarrow \pi_l \in \mathbb{R}$.
- Dual variables of constraints (6.3) $\rightarrow \alpha_u \leq 0$.
- Dual variables of constraints (6.4) $\rightarrow \beta_w \leq 0$.
- Dual variables of constraints (6.5) $\rightarrow \gamma_i^t \leq 0$.
- Dual variables of constraints (6.6) $\rightarrow \delta_{j,k,v_1,v_2} \leq 0$ satisfying the requirements of the constraint.
- Dual variables of constraints (6.7) $\rightarrow \mu_{j,k,v_1,v_2} \leq 0$ satisfying the requirements of the constraint.
- Dual variables of constraints (6.8) $\rightarrow \xi_w^{(l^j,l^k)} \leq 0$.

Let $\zeta_u := \sum_{v \in U_i^a : u \preceq v, \Delta(u,v) < a(i,e)} \alpha_v$ be the sum of dual variables $\alpha_v$ affecting the arrival of a train to node $u$ for $i \in S, a \in \delta_N^-(i), u \in U_i^a$.
Let $\eta_w := \sum_{v \in W_i^a : w \preceq v, \Delta(w,v) < d(i,e)} \beta_v$ be the sum of dual variables $\beta_v$ affecting the departure of a train from node $w$ for $i \in S, a \in \delta_N^+(i), w \in W_i^a$.
Let $\vartheta_u := \sum_{v \in U_i^e : u \preceq v, \Delta(u,v) < h(i,e)} \gamma_i^t$ be the sum of dual variables $\gamma_i^t$ affecting the arrival of a train to node $u$ for $i \in \hat{S}^e, e \in \delta_N^-(i) \cap E, u \in U_i^e : \theta(u) = t$.
Let $\kappa_{l,w}$ and $\psi_{l,w}$ be the sum of all variables $\delta_{j,k,v_1,v_2}$ and $\mu_{j,k,v_1,v_2}$ respectively over all constraints for which $l$ is either $l^j$ or $l^k$ and node $w$ is one of the nodes in the summations for train $j$ or $k$ in (6.6) and (6.7) (i.e. $v_1 \preceq w \prec v_3$ if $l = l^j$ or $v_2 \preceq w \prec v_4$ if $l = l^k$).
Finally, let $\varepsilon_{l,w} := \sum_{v \in W_i^a : w \preceq v, \Delta(w,v) < T_s} \xi_v^{(l^j,l^k)}$ be the sum of dual variables $\xi_v^{(l^j,l^k)}$ affecting the departure of a Frequency line $l$ from node $w$ for which $l$ is either $l^j$ or $l^k$ in $(l^j, l^k) \in \Xi$ for $i \in S^j \cap S^k, a \in \delta_N^+(i) \cap (A^j \cap A^k), w \in W_i^a$.

For a path $p \in P^j$ let $U_p$ and $W_p$ be the set of arrival and departure nodes visited by path $p$. The reduced cost $\hat{c}_p$ of path $p$ is given in Equation 6.10.

$$\hat{c}_p = PT_p - \pi_l - \sum_{u \in U_p} (\zeta_u + \vartheta_u) - \sum_{w \in W_p} (\eta_w + \kappa_{l,w} + \psi_{l,w} + \varepsilon_{l,w}) \tag{6.10}$$

Analogously, for a line train paths group $q \in Q^l$ let $U_q$ and $W_q$ be the set of arrival and departure nodes visited by line train paths $q$. The reduced cost $\hat{c}_q$ of line train paths $q$ is given by Equation 6.11.

$$\hat{c}_q = c_q - \pi_l - \sum_{u \in U_q} (\zeta_u + \vartheta_u) - \sum_{w \in W_q} (\eta_w + \kappa_{l,w} + \psi_{l,w} + \varepsilon_{l,w}) \tag{6.11}$$

### 6.2.2 Pricing Problem

The goal of the Pricing Problem (PP) is to find new promising train paths for the RMP. There is one PP per line and their function is to create a group of *line train paths* with the potential to improve the objective function. For example, if line 2 has a frequency of 2 train per hour, the related pricing problem is in charge of creating 4 train paths (2 in each direction) that are feasible between them. Here is where the Symmetric Line graph formulation described in section 5.4.2 becomes relevant. The use of a single graph for all the train paths of a line reduces the PP to a single *shortest path problem*. The graph is directed acyclic (see Section 5.4.2). From the fact that all the dual variables affecting the graph are non-positive and they are subtracted from the original weights of the edges (see Equation 6.11), it can be concluded that the graph has always non-negative edge weights. Therefore, this problem can be solved using Dijkstra's algorithm (Dijkstra, 1959). This algorithm can find the shortest path between two nodes in the graph in a polynomial time that can be approximated to the upper bound of $O(|V^2|)$ (Cormen, 2009). However, for less dense graphs with fewer than $V^2$ edges, the upper bound can be tighten more using different data structure methods. It follows an iterative process that is summarized in Algorithm 1.

The PP needs to take into account the dual values from the constraints in the RMP. In order to do that, the dual values related to either departure or arrival nodes are applied to the respective outgoing arcs in the Symmetric Line graph. These values are the ones from all constraints except 6.2 that is independent of the train path. Let $(m, n) = a \in \hat{R}^l$ be an arc in the Symmetric Line graph formulation where $m, n \in \hat{V}^l$. Therefore, the dual values of the departure and arrival nodes (from Train graphs) covered by node $m$ (from the Symmetric Line graph) affect all the outgoing arcs from node $m$. Each Symmetric Line graph node $m \in \hat{V}^l_a$ has four Train graph formulaiton nodes associated (see Figure 5.4). Let $w^m_1, w^m_2, u^m_1, u^m_2 \in V$ be the four nodes of the Train graphs that correspond to node $m$ in the Symmetric Line graph formulation, where the superscript denotes which node in the Symmetric Line graph it belongs to, the subscript denotes the direction of the train and $w, u$ define that is a departure/arrival node respectively. Following this nodes, $\Delta(w^m_1, w^n_1)$ defines the sum of running and dwell time of train in direction out of KH ($d = 1$) for arc $a$. Analogously, $\Delta(u^n_2, u^m_2)$ defines the sum of dwell and running time of train in direction towards KH ($d = 2$) for arc $a$. In this way, the updated cost $c_a$ of the arc for $a = (m, n) \in \hat{R}^l$ can be calculated as follows:

$$c_a = \Delta(w^m_1, w^n_1) + \Delta(u^n_2, u^m_2) - \sum_{d \in D} (\eta_{w^m_d} - \kappa_{l,w^m_d} - \psi_{l,w^m_d} - \varepsilon_{l,w^m_d} - \zeta_{u^m_d} - \vartheta_{u^m_d})$$

**Algorithm 1** Dijkstra's algorithm

**procedure** DIJKSTRA(Graph, source, target)
  Create vertex set $Q$
  **for all** $v$ in Graph **do** ▷ Initialization
    $dist(v) \leftarrow \infty$ ▷ Initial distance from source to all the nodes set to infinity
    $prev(v) \leftarrow Undefined$ ▷ Previous node in optimal path from source
    add $v$ to $Q$ ▷ All nodes initially in $Q$ (non-visited)
  **end for**
  $dist(source) \leftarrow 0$ ▷ Distance from source to source is zero
  $u \leftarrow 0$
  **while** $u \neq target$ **do**
    $u \leftarrow$ vertex in $Q$ with min $dist(u)$ ▷ Node with least distance selected
    remove $u$ from $Q$
    **for** neighbor $v$ of $u$ **do** ▷ Check non-visited adjacent nodes of $u$
      $alt \leftarrow dist(u) + length(u, v)$
      **if** $alt < dist(v)$ **then**
        $dist(v) \leftarrow alt$ ▷ A shorter path to $v$ has been found
        $prev(v) \leftarrow u$
      **end if**
    **end for**
  **end while**
  **return** $dist(target), prev(target)$
**end procedure**

Note that when updating the costs for a line with a frequency of two trains per hour and direction, the cost of the arc should also include the updated weights of the nodes of train paths deferred half an hour (i.e. $\theta(m) + \frac{T}{2}$)

In the case of a rush hour line, the new graph is only concerned about one direction and therefore, the cost of the arc can be simplified to the following equation:

$$c_a = \Delta(w_1^m, w_1^n) - \eta_{w_1^m} - \kappa_{l,w_1^m} - \psi_{l,w_1^m} - \varepsilon_{l,w_1^m} - \zeta_{u_1^m} - \vartheta_{u_1^m})$$

Let $A_q$ be the set of arcs forming the shortest path in the graph calculated using Dijkstra´s algorithm. The reduced cost $\hat{c}_q$ of the line train paths $q \in Q^l$ is the following:

$$\hat{c}_q = \sum_{a \in A_q} c_a - \pi_l$$

Every time the PP finds a line train paths group $q \in Q^l$ with a negative reduced cost, it is added as a new variable (i.e. column) to the RMP and it is included in all the constrains where it has a non-zero coefficient.

## 6.3 Separation procedure

The number of constraints of the whole problem does not increase in the same proportion as the variables do but can be also large. There are $O(|D| \cdot |E| \cdot |A| \cdot |T|)$ arrival and departure Headway constraints and $O(|D| \cdot |\Xi| \cdot |\hat{A}| \cdot |T|)$ Frequency constraints where $\hat{A}$ is the maximum set of shared track segments by any pair of trains considered. Additionally, the amount of Overtaking and Crossing constraints are much larger, approx $O(|D| \cdot |\Upsilon^2| \cdot |\hat{A}| \cdot |T| \cdot b^{max})$ and $O(|\Upsilon^2| \cdot |\hat{E}| \cdot |T| \cdot b^{max})$ respectively, where $b^{max}$ is the maximum time interval between two incompatible departures over all stations and train pairs and $\hat{E}$ is the maximum set of single-track segments shared by any pair of trains considered. It can be seen that the amount of Overtaking, Crossing and Frequency constraints is much larger than the Headway constraints. Considering all the constraints from the beginning results in a huge RMP that may be time consuming to solve. Most of those constraints may never be binding as the headway and instance restrictions do not allow certain overtaking or crossings in the network by default. Therefore, it is decided to add the Overtaking, Crossing and Frequency constraints by separation.

Once the column generation procedure stops providing line train paths groups with negative reduced cost the separation procedure is applied. The separation of constraints (6.6)-(6.8) is done by enumeration and all of them are checked in the same iteration. In this case, checking for constraint violations is a bit trickier as there might be fractional solution values. First, for each line $l \in L$ and node $v \in \hat{V}^l$ a value $z_v^l$ is initialized to value 0. Next, for each positive variable $\lambda_q$ in the RMP solution for $q \in Q$, the value of $z_v^l$ is increased by $\lambda_q$ for all nodes visited by $q$.

The violation of the Overtaking constraints (6.6) is given by the following condition:

$$\sum_{w \in W_i^a \cap V^j : v_1 \preceq w \prec v_3} z_w^{l^j} + \sum_{w \in W_i^a \cap V^k : v_2 \preceq w \prec v_4} z_w^{l^k} > 1,$$
$$\forall j, k \in \Upsilon, v_1, v_2 \in W_i^a$$
$$(\text{where } l^j \neq l^k, d^j = d^k,\ i, h \in S^j \cap S^k, a = (i,h) \in (A^j \cap A^k))$$

If the condition is satisfied, the corresponding Overtaking constraint is violated and it should be added to the RMP.

Similarly, the violation of the Crossing constraints (6.7) is given by the following condition:

$$\sum_{w \in W_i^e \cap V^j : v_1 \preceq w \prec v_3} z_w^{l^j} + \sum_{w \in W_h^e \cap V^k : v_2 \preceq w \prec v_4} z_w^{l^k} > 1,$$
$$\forall j, k \in \Upsilon, v_1 \in W_i^e, v_2 \in W_h^e$$
$$(\text{where } l^j \neq l^k, d^j \neq d^k,\ i, h \in S^j \cap S^k, e = (i,h) \in E^j, (h,i) \in E^k)$$

If the condition is satisfied, the corresponding Overtaking constraint is violated and it should be added to the RMP.

Finally, the violation of the Frequency constraints (6.8 is given by the following condition:

$$\sum_{v \in W_i^a : v \preceq w, \Delta(v,w) < K} z_v^{l^j} + z_v^{l^k} > 1$$

$$\forall (l^j, l^k) \in \Xi, d \in D, w \in W_i^a$$

$$(\text{where } j, k \in \Upsilon, i \in S^j \cap S^k, a \in \delta_N^+(i) \cap (A^j \cap A^k))$$

If the condition is satisfied, the corresponding Frequency constraint is violated and it should be added to the RMP.

Once the violated constraints are added to the model, the column generation procedure should be restarted. Adding more constraints to the model modifies the solution space and new columns with negative reduced cost can be found. The overall procedure of column generation and separation can be summarized in the following steps:

1. Initialize a reduced LP including only constraints (5.2)-(5.6) and ”dummy” variables that form a feasible solution to the problem.

2. Solve the reduced LP and obtain primal $x^*$ and dual $y^*$ solutions.

3. Apply column generation and if variables with negative reduced cost are found with respect to $y^*$, add them to reduced LP and go back to Step 2.

4. Apply separation for constraints (5.7)- (5.9) and, if constraints violated by $x^*$ are found, add them to the reduced LP and go back to Step 2.

5. The process is terminated since $x^*$,$y^*$ is an optimal primal and dual solution to the whole LP.

The initial ”dummy” variables correspond to line path groups that belong each to a different line in order to satisfy constraint (5.2) but do not stop anywhere and have a very high cost, in order to avoid being chosen in the optimal solution.

## 6.4 Dive heuristic

Once the separation procedure does not find any more violated constraints, the optimal solution for the Master Problem (MP) has been found. Nevertheless, this solution can still be fractional. In order to find an integer solution, a heuristic method is applied. Each time the MP optimal solution is fractional, the values $z_v^l$ are calculated for each graph node (see Section 6.3) and one with a fractional value $z_v^l$ is enforced to be chosen in the following iterations. This means that the following train paths generated for line $l$ need to contain node $v$ in their graph path. Apart from fixing the node, all the previously generated columns from line $l$ that do not include the node $v$

need to be removed from the RMP. Once the heuristic step is concluded the column generation should be started again as new promising columns may be generated.

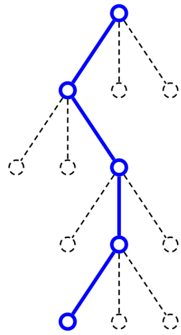

Figure 6.2: Heuristic dive in an enumeration tree (Sadykov et al., 2018)

Figure 6.2 shows a small illustration of how the dive heuristic affects the path chosen in the *branch-and-bound* tree. It can be appreciated that branches of the tree are skipped and only one direction is followed. One advantage of this method is that it can lead faster to an integer feasible solution, although it would probably be sub-optimal. A disadvantage of this method is that, as seen in Figure 6.2, some branches of the tree are left unexplored and forcing the integrality of specific nodes that were fractional can lead to an infeasible final solution. If that is the case, it is important to realize as soon as possible if the procedure is heading towards infeasibility and cut the whole algorithm. One way of estimating this is by measuring the percentage of "dummy" columns used in an MP optimal solution. Once the column generation and separation procedures finish achieving a MP optimal solution, if the lambda values of the initial dummy columns are used more than 2 % in the solution, then it is assumed that the algorithm cannot longer find any integer feasible solution to the problem and should be restarted. This percentage has been chosen as a rule of thumb in relation to the high cost assigned to the dummy columns in the RMP objective function and the optimal LP solution at the root node.

Next, the two main strategies are described and further tested in order to choose the best node at each iteration.

### 6.4.1 Sequential strategy

The nodes are checked in an specific order. This order is given by the sequence of the lines. For each Symmetric Line graph, the nodes are checked by simple enumeration. The sequence of the lines is randomized at each algorithm iteration. Next, the $z_v^l$ values are checked for each node. After checking all the node values, only the ones with the most fractional value are stored in a pool. For example, if a node used 0.5 times is found, the final pool of most fractional nodes will only contain nodes with a fraction of 0.5. However, if one of the most fractional node is used 0.4 times, the final pool of nodes can also include nodes with a 0.6 fraction, as they are both considered equally fractional. As the name of this strategy indicates, the nodes are chosen in a sequential way, meaning that always the first node found that is in the final pool is chosen. This prioritizes nodes from lines that are at the beginning of the line sequence order and inside the graph.

### 6.4.2 Random strategy

In this strategy, the pool of most fractional nodes is generated in the same way as before, but this time, the first one is not chosen and a random one is chosen instead. This allows more diversity in the lines and parts of the network affected by the nodes but it can also be more unstable and risky.

In the next chapters, these two strategies are referenced by Sequential strategy and Random strategy respectively.

Last but not least, in both strategies, the nodes with a fractional value greater or equal to 0.5 are prioritized. Following the previous example, if the pool of most fractional nodes includes both nodes with a fraction of 0.4 and 0.6, then, the nodes with a fraction of 0.6 are prioritized. This decision increases the chances of finding a feasible solution as the branch is not that disruptive. Nevertheless, it may require more iterations as it has a lower impact in the solution space.

## 6.5 Passenger travel time

One of the main overall objectives of the model is to improve passenger travel time. So far, the method minimizes the length of the train paths. This avoids extra additional dwelling of the trains at the stations and allows passengers travelling in the train to reach their destination fast. However, many passengers travelling along the network need to take more than one train in order to reach their final destination meaning that they need to transfer at a specific station. The whole trip time of these type of commuters is not just dependent on the train path length but also on the waiting time at the station (from now on referred as *transfer time*). Therefore, minimizing these transfer times becomes part of the overall objective of optimizing the passenger travel time.

### 6.5.1 Passenger routing

The first step for calculating the passenger travel time in the network is defining the routes (i.e. train combinations) that each passenger will choose to travel from its origin station to its final station.

Once an integer feasible solution has been found, a graph is created representing all the train paths that form it. The same graph representation is used as the initial Train graph formulation described at the beginning of Section 5.4. Let $w_t^s$ be the departure node related to station $s \in S$ and time $t \in T$. Analogously, let $u_t^s$ be the arrival node related to station $s \in S$ and time $t \in T$. Moreover let $\sigma_s$ and $\tau_s$ be the respective artificial *source* and *sink* nodes for station $s \in S$. The train path arriving and departing times are connected with arcs only linking those stations where the train stops (i.e. where passengers can board or leave the train). The cost of those arcs is the time interval between the two nodes (i.e. $\Delta(w, u)$). Figure 6.3 shows an example of two train paths in the same direction represented in the graph. It can be seen, for example, that the purple train goes from Odense (OD) to København H (KH) and does not stop at Roskilde (RO).

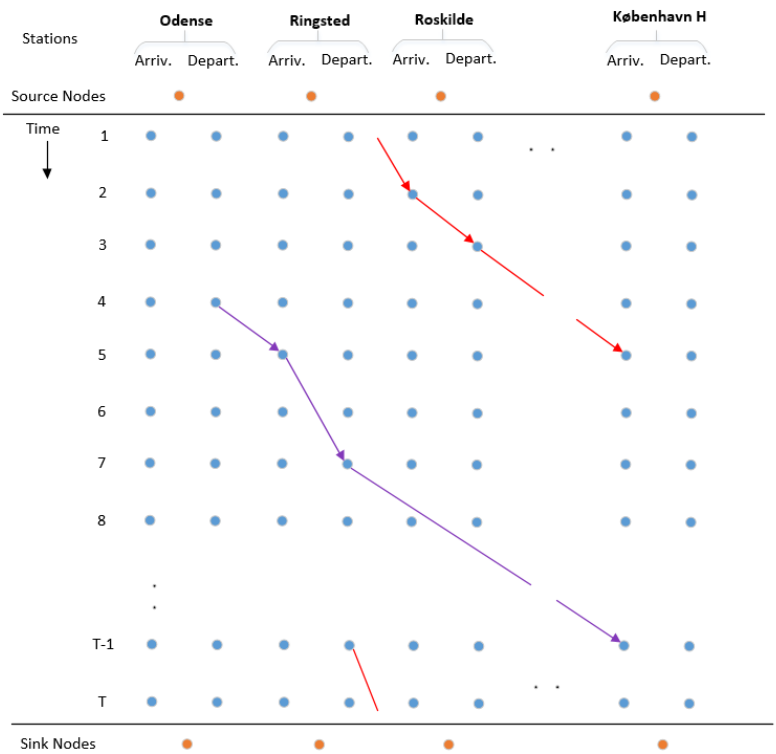


Figure 6.3: Solution train paths representation example. The purple train goes from OD to KH and the red train goes from RG to KH

In order to represent the transfer times, an arc is created between any arrival time and departure time nodes for different trains in the same station with the related time interval as cost. It is assumed that from a train it is possible to transfer to any other train in the station. However, a transfer between a train arriving and another departing at the same exact time is not realistic and a minimum transfer time is needed. The minimum transfer time is the amount of time needed by the passengers to get off the arriving train, walk from the arriving platform to the departing platform and board the departing train. This time can be very different depending on the complexity of the station, the number of platforms, etc. The analysis of the minimum transfer time for each station can lead to a whole case study itself, and in this case, a rule of thumb value of 5 minutes has been chosen for all stations in the network. This value is reasonable taking into account that in a real-life scenario the arriving train can be slightly delayed and the passengers should preferable still be able to transfer trains. Furthermore, from a psychological perspective, knowing that the transfer time is tight is not comfortable for the passenger and the fear for the possibility of missing the transfer can lead to dissatisfaction.

Apart from allowing transfer between trains at each station, a passenger starting his/her trip should be allowed to board any train departing from his original station. In order to do so, arcs are added to the graph from the artificial source node of each stations to the departure time node of each train in that station. Analogously, an arc is needed from any arrival of a train at a station to the artificial sink node of that station. These two type of arcs are defined as origin/destination arcs and have a null weight. Therefore, it is assumed that all passengers arrive at their origin station at the exact time their train departs. Figure 6.4 shows the solution graph from the previous example but with the transfer and origin/destination arcs included.

Once all the solution train paths are added to the graph and the respective transfer and origin/destination arcs, the route of the passenger can be computed as the shortest path from the artificial source node of the original station to the artificial sink node of the destiny station and the total travel time is directly given by the sum of costs of the arcs. Figure 6.5 depicts an example route of a passenger travelling from OD to RO for the given solution train paths from the previous figures.

### 6.5.2 Origin-Destination matrix

Once the solution graph is created to calculate the combinations of trains that each passenger takes, the Origin-Destination matrix is used to calculate the total passenger travel time. For each pair of stations the route is calculated and multiplied by the expected amount of passengers travelling between the two stations. As mentioned in Section 4.3, the OD matrix is given with the annual forecast of passengers for the 2022. As this case study covers the morning rush hour, an estimation for the passenger demand during the cycle time has been accorded with DSB based on the working days and daily demand distribution. These passenger demand values are relative to the network considered and they cannot be seen as absolute value estimations.

In fact, the updated OD matrix does not include information about the passengers travelling from or to any station outside the network studied. Therefore, people travelling from or to Jutland, Sweden, Germany or North-east of Zealand are not

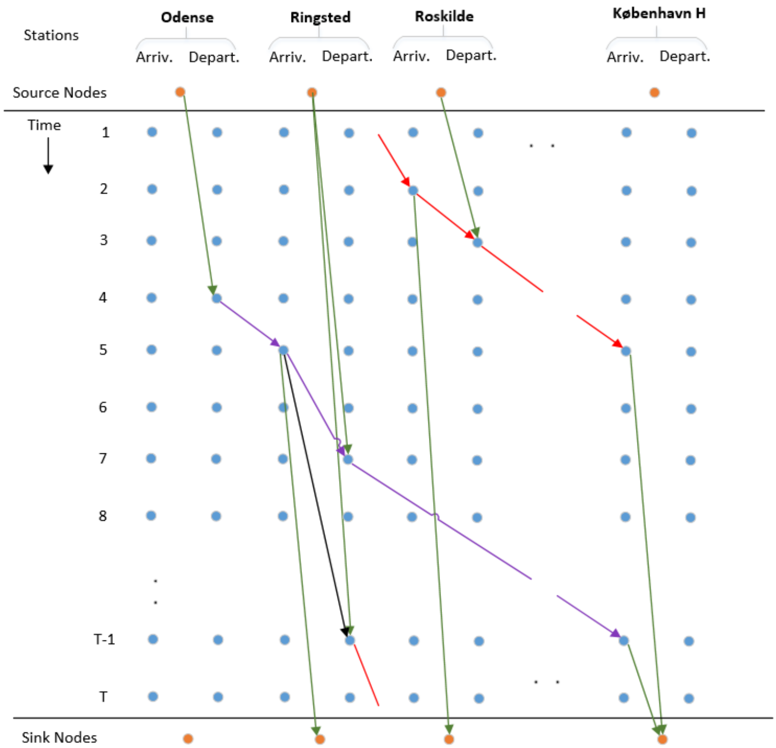


Figure 6.4: Solution graph example with transfer and origin/destination arcs. Transfer graphs are black and origin/destination arcs are green.

considered. This decision has been taken due to the number of stations in the whole country and resources required to make the calculation. Moreover, most of the stations correspond to small stations that have a negligible passenger flow. Also, the lines outside the network that are not considered in this study may affect the passenger route. In a nutshell, the updated OD matrix gives a good overview of the internal flow of passengers but could be improved by, for example, taking into account additional flows such the ones coming from Aarhus or Aalborg.

## 6.6 Large Neighborhood Search

The process so far allows to find an integer feasible solution to the network instance. The solution can vary from run to run due to the randomness in the diving heuristic. However, as mentioned in Section 6.5, the column generation and separation procedure does not take into account the passenger routing and transfer times and only

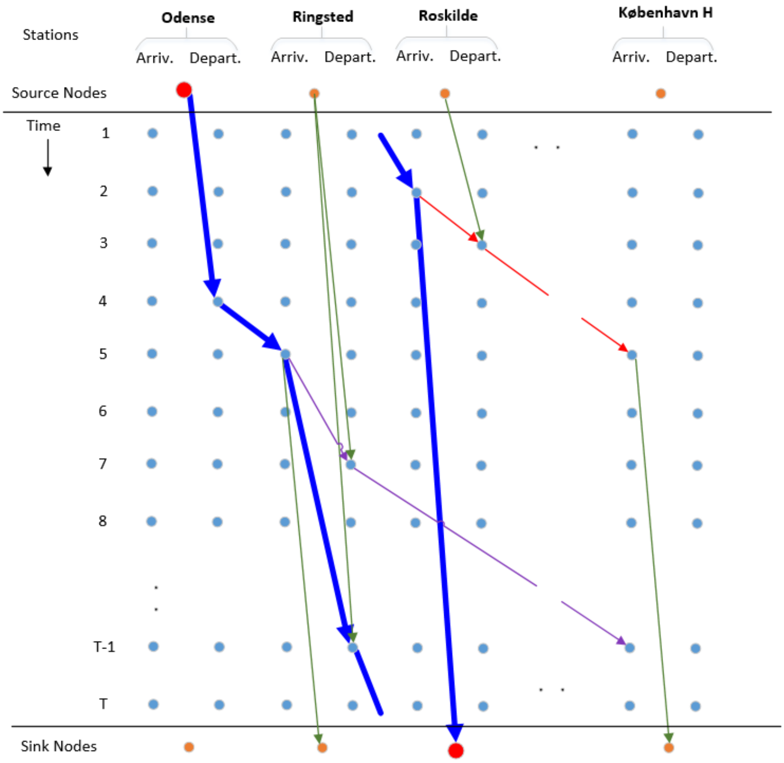


Figure 6.5: Route example of a passenger travelling from Odense to Roskilde

the passenger travel time is computed once the solution is found. Therefore, it is interesting to study, based on the solution found, how to improve the current solution taking into account the passenger travel time in the network. The idea applied here resembles the metaheurictic method Large Neighborhood Search (LNS). This method, was first introduced by Shaw (1998) and consists basically in destroying part of the solution and repairing it in order to get a new solution. The new solution hopefully is better than the previous one and the procedure repeats iteratively. The procedures are called *destroy* and *repair* method respectively and Algorithm 2 depicts the generic version of the LNS for a minimization problem.

Starting from an integer feasible solution, part of the solution is destroyed. The solution needs to be destroyed enough so that a new solution can be found. The repair method repairs the destroyed solution creating a new one. The combination of the two methods form a neighborhood, that, depending on the level of destruction and the constructive procedure, can be very large. Once a new solution is found, two conditions are checked. First, an acceptance criteria should be matched. If the new solution is accepted, it becomes the next candidate solution to be destroyed and

**Algorithm 2** Large Neighborhood Search

**procedure** LNS
  Generate integer feasible solution $x$
  $x^b = x$ ▷ Found solution is stored as the best so far
  **repeat**
    $x^n = repair(destroy(x))$ ▷ Generates a new solution
    **if** $accept(x^n)$ **then** ▷ Acceptance criterion
      $x = x^n$
    **end if**
    **if** $cost(x^n) < cost(x^b)$ **then** ▷ Improvement criterion
      $x^b = x^n$
    **end if**
  **until** time limit
  **return** $x^b$
**end procedure**

repaired, otherwise, the previous solution is destroyed and repaired again. Secondly, an improvement criterion should be checked. If the new solution found is better than the current best solution, it is stored as the new current best one. The process is repeated until a termination criteria is met, usually is a time limit but it can also be limited by other parameters such as the number of iterations. Finally, when the process is terminated, the current best solution stored is returned.

### 6.6.1 Destroy method

The destroy method can be designed in many different ways. In this case, the main goal is to destroy part of a solution taking into account the overall optimization goal of improving the passenger travel time. The algorithm itself already creates the shortest possible train paths. Therefore, the focus of the destroy method should be in the transfer times between trains.

The goal of the proposed destroy method is to find the potential *best* transfers and enforce them to be part of the next solution by fixing information about the transfer and removing the rest of the train paths. Once the passenger routing for the found solution is calculated, it is also possible to see the occupancy of the trains and which stations are the most transited. In this case, there are two criteria to check when considering the transfers. The flow of passengers that transfer between any two trains at the station and the transfer time between the arrival of the first train and the departure of the second. If the transfer time is already small (i.e. the minimum time of 5 minutes) , the transfer selected will not be improved and the solution would more likely be the same or very similar. As a result, a minimum transfer time is needed above which the transfer found can be considered improvable. Therefore, it would be considered the *best* transfer the one that accounts for the most passengers in the current solution and satisfies a minimum transfer time defined.

As a result, this method requires two main parameters to select the *best* transfers. First, the amount of transfers to be considered and, second, the minimum transfer time required.

Once the transfers are selected, almost all the solution is destroyed except the arriving time of the arriving train in the transfer. In a similar way as performed in the dive heuristic described in Section 6.4, for each transfer selected, the node in the Symmetric Line graph related to the arrival time at the station of the arriving train is fixed. This means that when the algorithm starts a new iteration, all the columns generated that are related to that train, must include a train path that arrives at the selected station at the selected time. In the case of the departing train of the transfer, a linear cost function is applied to the nodes related to the departing time instants from the selected station. Figure 6.6 illustrates the linear cost function applied to the departing trains of each transfer. It is visible that the ”cheapest” departure time corresponds to the ideal transfer time of 5 minutes. From then, the cost increases linearly as the transfer time increases. In addition, the cost increment per time unit is proportional to the transfer demand in order to prioritize more demanded transfers to be closer in time.

When it comes to fixing the nodes of the arriving transfer lines of the chosen transfers, two main strategies have been considered.

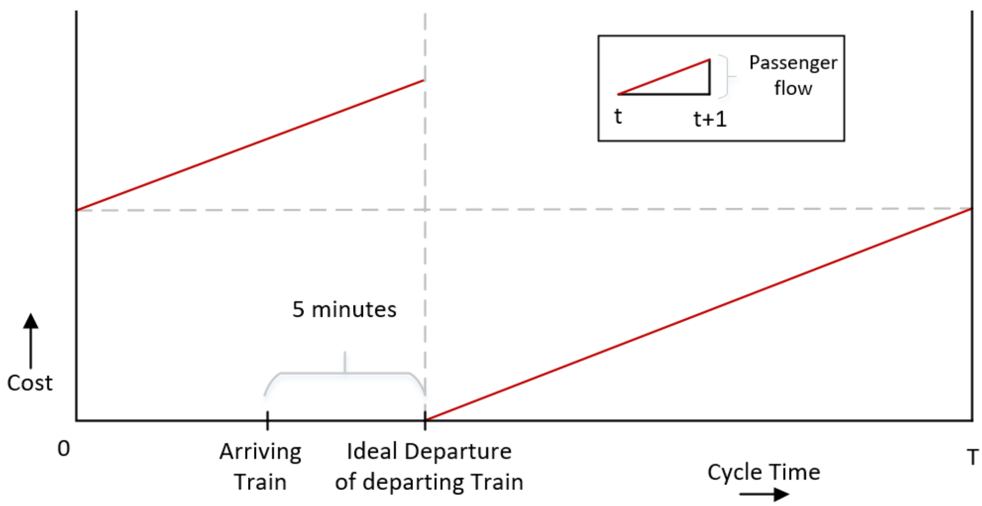


Figure 6.6: Linear cost function for the departing times of the departing train of the transfer where the departing time cost is represented by the red line

#### 6.6.1.1 All lines strategy

The first strategy considers fixing the arrival time at the stations from the arriving trains of all transfers considered. This means that, if there are multiple transfers affecting the same train, the train paths will include multiple fixed nodes. In the same manner, departing trains may accumulate multiple linear cost functions in their graphs. This strategy would never create any conflict in terms of feasibility because just parts of an actual integer feasible solution are being enforced and nothing additional is added. On the other hand, if many transfers are selected, the solution space can be drastically reduced and the margin of improvement decreased.

#### 6.6.1.2 Independent lines strategy

The second strategy is a more relaxed one and considers only fixing nodes of arriving trains that were not part of any transfers fixed before. In this way, trains will only have either maximum one node fixed or one linear cost function in the departure times added in the graph.

In the following chapters, both strategies will be referenced as All Lines strategy and Independent Lines strategy respectively.

In this algorithm the repair method consists in the applying the same procedure of column generation and separation together with the dive heuristic. The RMP is restarted without including any column and just the graphs are kept updated with the fixed nodes and the linear cost functions.

In this case, all solutions are accepted and used as candidates to be destroyed and repaired again, meaning that there is no acceptance criterion defined. This is based on the behaviour of the destroy method. As the PTT is calculated based on the quickest routes, a variation of half a minute in the train paths of a highly transited segment can change completely the flow of passengers. Therefore, finding a bad solution does not mean that, by fixing the right transfers, a total different solution can be achieved with the best PTT so far. Moreover, if only improving solutions are accepted the chance to ending up finding a repeated solution increases.

## 6.7 Solution method variants

Apart from the main solution method explained, different variants of it can be suggested and have been implemented.

First, following the train graph formulation described in Section 5.4, the PP can be modelled as a MIP model. There is a PP for each line, and it considers the Train graphs related to both direction of the line. Then, the MIP model tries to combine the shortest paths of both graphs taking into account symmetry and eventual crossings in the single-track. In this model, the symmetry of the paths is more permissive as symmetry between paths is only enforced in the end-of-line stations allowing a larger set of possible solutions. In order to find a feasible combination of train paths, the MIP model defines one binary variable for each arc in a Train graph and the necessary constraints forbidding incompatible combinations of arcs are added. The number of nodes and arcs considered is constant regardless of the symmetry gap allowed. However, the model is no longer a single shortest path problem and the computational time may increase.

Another variant of the solution method proposed consists in creating an initial set of line train paths (columns) that serve as an starting point for the algorithm. It is known that if a line does not use any single-track segment, the trains in both directions can travel at the minimum path length without conflicting with each other. Moreover, some of the planned timetables by DSB from the previous years actually included line train paths that run at the minimum path length. When the Symmetric Line graphs are initialized, the shortest path in the graph is seen as the compatible combination of the fastest trains. Additionally, the number of shortest paths in each graph would be at least $T$ as the same timetable for a line can be computed by shifting the shortest path one time instant at a time. Therefore, an iterative function that creates a set of line train paths that consists in the set of shortest paths in the Symmetric Line graph for each line is suggested.

A different approach for improving the transfers in the network is also suggested. After an integer feasible solution is found, the passenger flow through the network is calculated and variables referring to the transfer times between any pair of trains are created and included in the objective function with a coefficient related to the respective flows. The extension of the problem can be formulated as follows:

| | |
|---|---|
| $t_s^{b^j,b^k}$ | transfer time between the $b^j \in F^{l^j}$ frequency train $j \in \Upsilon$ and the $b^k \in F^{l^k}$ frequency train $k \in \Upsilon$ at station $s \in S$ |
| $y_s^{b^j,b^k} \in \{0,1\}$ | 1 if $t_s^{b^j,b^k}$ includes a cycle time change, 0 otherwise |
| $H_s^{b^j,b^k}$ | passenger flow between the $b^j \in F^{l^j}$ frequency train $j \in \Upsilon$ and the $b^k \in F^{l^k}$ frequency train $k \in \Upsilon$ at station $s \in S$ |
| $\Lambda_s^{b_j} \subseteq Q$ | the set of line train paths that include a $b^j \in F^{l^j}$ frequency train $j \in \Upsilon$ arriving at station $s \in S$ |
| $\Lambda_s^{b_k} \subseteq Q$ | the set of line train paths that include a $b^k \in F^{l^k}$ frequency train $k \in \Upsilon$ departing from station $s \in S$ |
| $u_s^q$ | arrival time of $b^j \in F^{l^j}$ frequency train $j \in \Upsilon$ at station $s \in S$ for column $q \in \Lambda_s^{b_j}$ |
| $w_s^q$ | arrival time of $b^k \in F^{l^k}$ frequency train $k \in \Upsilon$ from station $s \in S$ for column $q \in \Lambda_s^{b_k}$ |

The new objective function of the problem can be defined as the following:

$$min \quad \sum_{q \in Q} c_q \cdot \lambda_q + \sum_{s \in S} \sum_{j \in \Upsilon} \sum_{k \in \Upsilon} \sum_{b^j \in F^{l^j}} \sum_{b^k \in F^{l^k}} H_s^{b^j,b^k} \cdot t_s^{b^j,b^k}$$

and and additional set of constraints is required to enforce the actual transfer times

$$(\sum_{q \in \Lambda_s^{b_k}} w_s^q \cdot \lambda_q) - (\sum_{q \in \Lambda_s^{b_j}} u_s^q \cdot \lambda_q) - t_s^{b^j,b^k} + T \cdot y_s^{b^j,b^k} = 0$$

$$\forall s \in S, j,k \in \Upsilon, b^j \in F^{l^j}, b^k \in F^{l^k}$$

$$t_s^{b^j,b^k} \geq 0$$

$$y_s^{b^j,b^k} \in \{0,1\}$$

This extension of the formulation includes the transfer time in the main objective function. If relaxation is applied to this formulation, the binary variables can become fractional and would prioritize to combine in order to find zero values for the transfer time variables. In addition, new dual variables need to be considered in the model. Therefore, the approach suggested solves the original formulation and, once a solution is reached and the passenger routing is computed, a MIP model is formulated including the extension presented. The new MIP model is solved taking into account the generated columns so far and hoping to find a different solution (or at least the same one) that has a lower passenger travel time.

Finally, it is noticed that sometimes the Column Generation procedure iterates more than usual over the same RMP objective value, presenting slight symptoms of degeneracy. A quick strategy is suggested that can be applied in order to avoid these additional iterations. A small percentage gap can be defined, and, if during a defined number of consecutive iterations, the RMP objective value improves less than the gap considered, the CG procedure can be forced to terminate.

CHAPTER 7

# Computational results

This chapter starts by introducing both the visual and computational results one can extract from the model. Then, the parameters defining the algorithm are presented and the best setting for it is found performing a parameter tuning using diverse instances. Afterwards, the algorithm's performance is tested in different scenarios. Finally, the robustness level of the timetables is analyzed and as well as the capabilities and performance of the algorithm.

## 7.1 Introduction to an example graphical solution

Before analyzing the performance of the algorithm in terms of PTT and train path lengths it is recommended to familiarize with the visual tools that the model provides. The output of the algorithm is basically the best solution encountered during the iterative process. When analyzing the performance of the algorithm the PTT and sum of train paths length is taken into account. However, from the planning perspective it is useful to actually look at the timetable and evaluate it from a visual perspective. Therefore, from the output solution a set of graphs are generated. On one side, the timetables for the lines running in the different corridors using the already described time-space diagram are generated. Using the same graphical technique, the amount of passengers using each train along its path is shown. Finally, in order to check the passengers flow at the main transfer stations, a transfer graph is created for each station showing how many passenger transfer between any pair of trains at the station.

Becoming familiar with this graphical tools is useful to understand better the performance analysis of the algorithm, as there may be references to some of the graphs presented in this section.

### 7.1.1 Timetable per corridors

As mentioned in Section 4.2, there are six main corridors in the network considered. In addition, there are six possible combinations of corridors to match the end stations of the network with KH. These combinatios of corridors are used in Figures 7.1-7.6 to display the timetables of the lines running in the network. The timetable is shown for two consecutive rush hours to have a better perspective of the whole path of the trains. The symmetry axis is not explicitly displayed but train paths of the same lines are symmetrical with respect to 30 and 60 minutes.

The timetable for the corridor linking København H (KH) and Kalundborg (KB) is depicted in Figure 7.1. There is a single-track segment between KB and HK that can be seen as a low congested segment with maximum three trains running in a rush hour. The segment between HK and RO is double-track with four lines running towards KH per hour. The segment between RO and KH is seen as a highly congested segment where ten trains arrive at KH per hour during a morning rush hour. The congestion is specially stressed in the segment between HTA and KH where there is only one track per direction to allocate the trains whereas there are two tracks per direction between HTA and RO. In this case, the arrivals and departures at HK are concentrated around short time periods resembling the behavior of a hub. This can be seen as highly beneficial for passengers transferring at that station. However, hubs are not directly enforced in the model in order to allow a wider range of solutions.

Figures 7.2 and 7.3 depict the timetable for the lines between KH and and OD that run through the Main-Old and Main-new corrdiors respectively. Both timetables show all the lines between RG and OD. Then the lines using the Main-Old and Main-New corridor are split in the respective timetables. RG is seen as an important transfer station where four different corridors join (Great Belt, Main-Old, Main-New and Large-South corridors).

Figures 7.4 and 7.5 show the timetables of the lines running between KH and NF

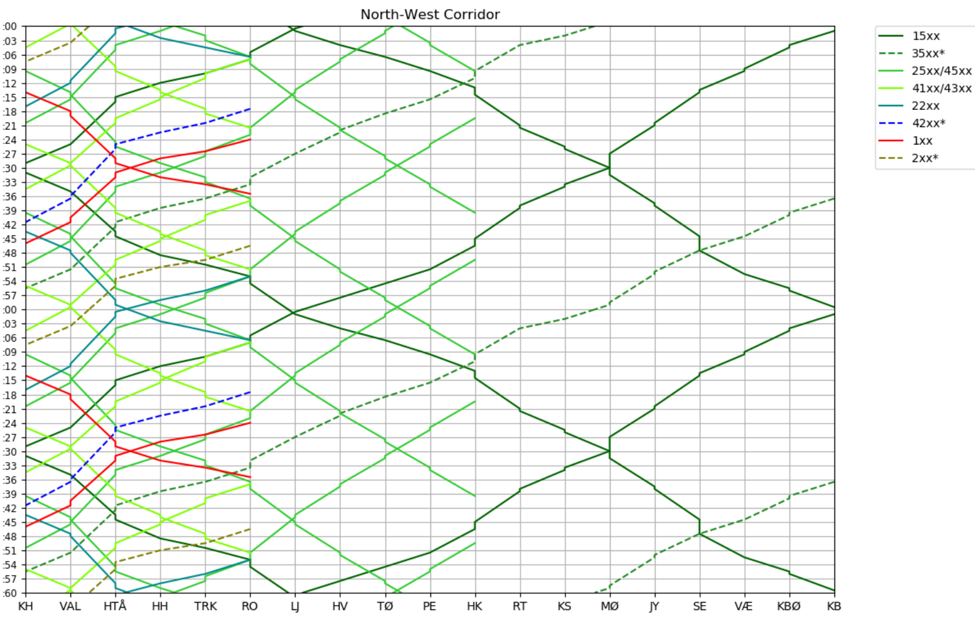


Figure 7.1: Timetable of the lines running through the North-West corridor

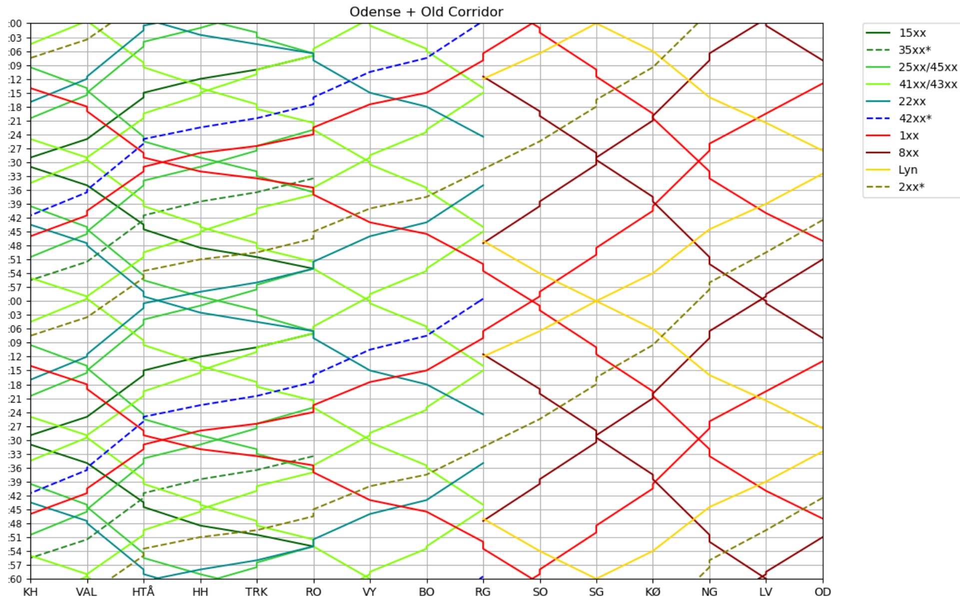


Figure 7.2: Timetable of the lines running through the Main-Old and Great Belt corridor

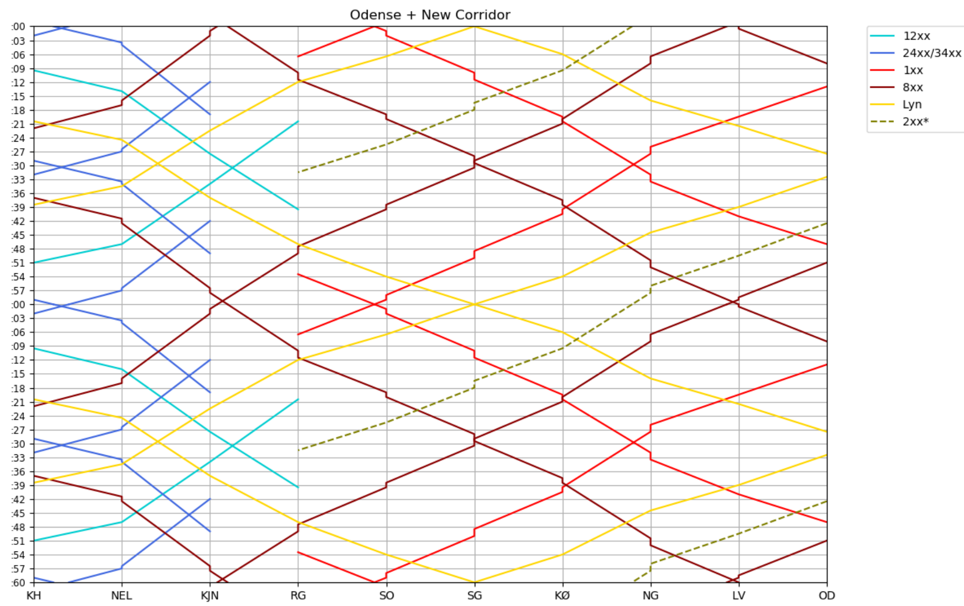


Figure 7.3: Timetable of the lines running through the Main-New and Great Belt corridor

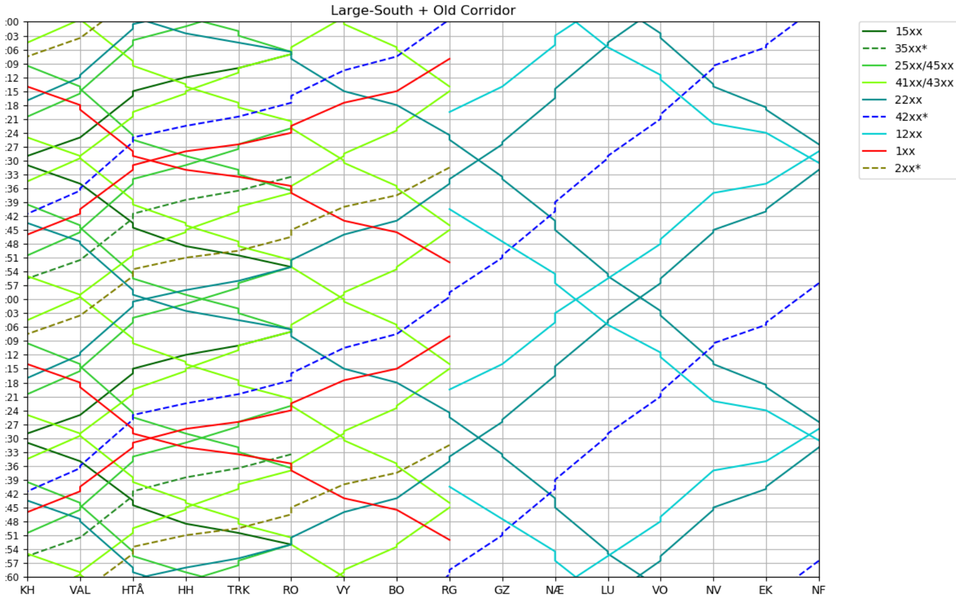


Figure 7.4: Timetable of the lines running through the Main-Old and Large-South corridor

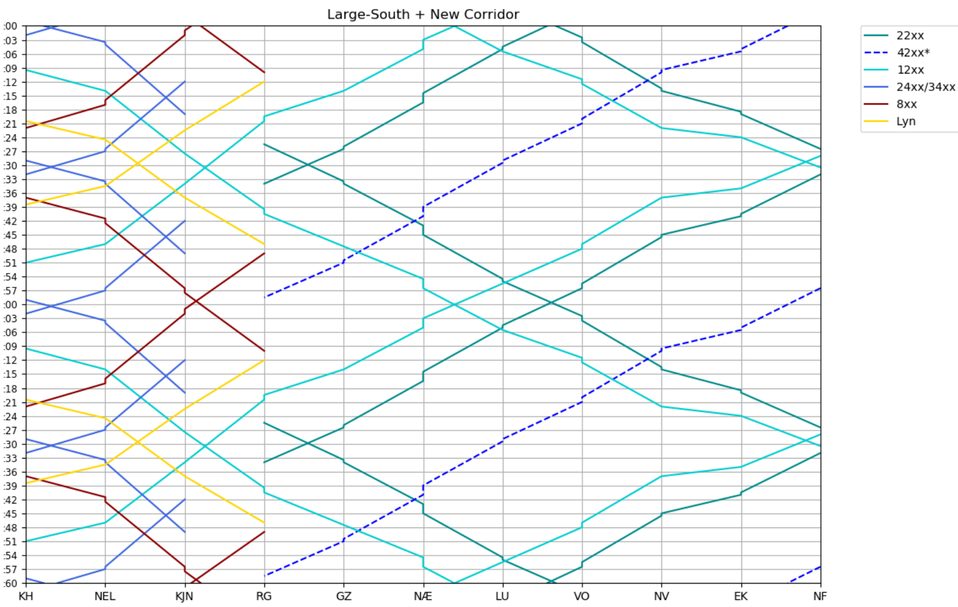


Figure 7.5: Timetable of the lines running through the Main-New and Large-South corridor

through the Main-Old and Main-New corridor respectively. These both timetables show all the lines running between NF and RG and also share RG as a possible transfer station.

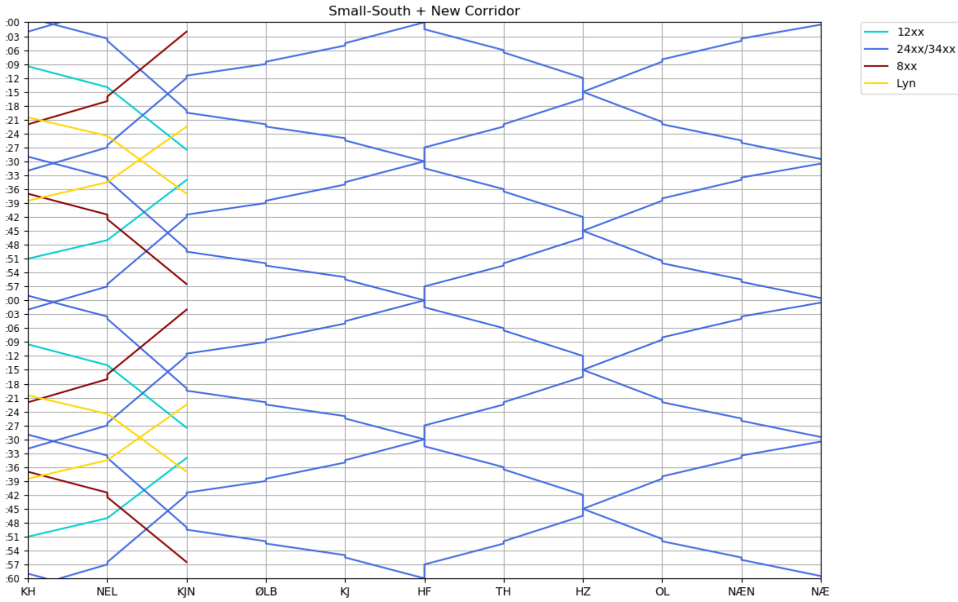


Figure 7.6: Timetable of the lines running through the Main-New and Small-South corridor

The last timetable is shown in Figure 7.6 where the lines running in the Small-South corridor are shown. This corridor is formed by a single-track. The segment linking KJN and KH is part of the Main-New corridor formed by double-track segments.

### 7.1.2 Train occupancy graphs

Another aspect of the solution that can be visualized with the model is the occupancy level of the trains at each track segment. The level is determined by the amount of passengers using the train as part of their trip. The occupancy levels are shown using the same time-space diagrams used for the timetables but the train path segments are coloured in a Green-Yellow-Red-Black scale where each color covers an interval of passengers amount. It can be seen as a "heat-map" where the green color is used for the less occupied train path segments and black color used for the highest occupancy level. In this study, the trains do not have a specific passenger capacity limit as from DSB was assumed that the capacity was not an issue for the demand considered.

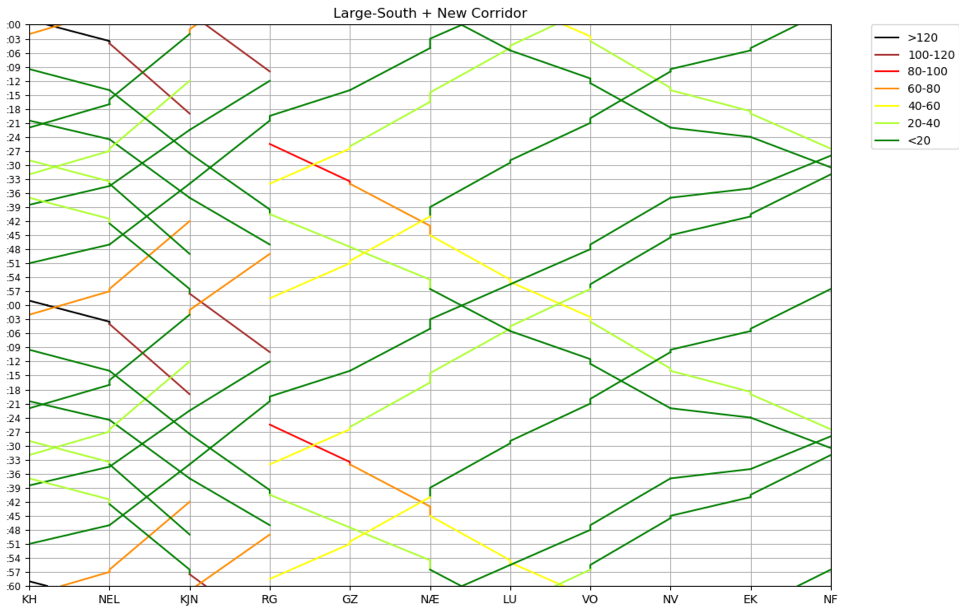


Figure 7.7: Amount of passengers travelling within the network on each train of the lines running through the Main-New and Large-South corridor

Figure 7.7 shows an example of train occupancy graph. This graph shows the same timetable as shown in Figure 7.5 but coloring the train paths in relation to the amount passengers inside the train. It can be seen that in more congested areas (i.e. KH-RG), the occupancy of the trains is irregular, with just one or a few trains carrying most of the passengers travelling in each track segment. On the contrary, in the less congested parts of the network (i.e. RG-NF) the occupancy of the trains seems to be more homogeneous and stable along the cycle time. This unbalanced distribution of passengers is mainly a result of the passenger routing approach and the assumption made that the passengers arrive at the exact time of the train departure. It is known that a high frequency of trains per hour is a result of the need to satisfy a high demand of passengers. Most of the passengers travel within relatively short trips that are done without transferring between trains. As a result, in network parts with a high frequency of trains per hour, all the passengers travelling directly between two stations would choose the fastest train, regardless of the the departure time. For example, 100 passengers travel per hour between two consecutive stations. There are 5 trains covering that trip per hour where 4 of them require 5 minutes and 1 reaches in 4.5 minutes to the station. The passenger routing approach described would consider that the 100 passengers will take the fastest train to reach their destination as it is the one that minimizes their travel time without considering any demand distribution along the rush hour. On the other hand, in less congested network parts where fewer trains run per hour, the transfer times become more important for the passengers as there are fewer transfer possibilities for the passengers to consider. This results in passengers choosing different trains depending on their route.

### 7.1.3 Passenger flow at main stations

Analyzing how occupied the trains travel is not the only tool that can be used to improve passenger travel time. The travel time for a passenger that uses a direct train is directly given by the train path. However, many passengers need to combine two or more trains to reach their destination. As a result, the transfer time is also part of their travel time.

A good way to visualize which train combinations are more recurrent for passengers and how long should they wait at the station, is by the use of *Station Transfer Graphs*. These graphs are illustrated in a circular layout resembling a *clock* where each node in the outer cycle represents a time instant (i.e. half a minute ) of the cycle time considered. Outside the circle, the arriving and departing trains are depicted referenced by the train number and direction (i.e. *trainNumber/direction*) and an arrow that tells if the train is arriving to the station (i.e. arrow towards circle) or departing from the station. (i.e. arrow outwards the circle). Each transfer between an arriving train and another departing train is illustrated with an arc. The thickness of the line is proportional to the flow of passengers. This graph can be created for any station in the network, nevertheless, just a small subset of them are actually used by passengers to transfer trains. These stations are RG, RO, KH and NÆ. Next, the transfer graphs for KH and RG related to the previously shown timetables are displayed in Figures 7.8 and 7.9 respectively.

At KH the passengers can only transfer to trains in the opposite direction as this station is the end-point station for all the lines. Trains of the same line arrive and depart at symmetrical times in accordance to the symmetry axis set at 30 and 60 minutes. One advantage mentioned in Section 3.1 about symmetric timetables, is that it allows the same transfers in both directions. In some cases, both symmetric transfers are used in a similar proportion indicating that the shortest path between two stations may share same transfers in both directions. It can be seen that the busiest transfers tend to have a low interval of time (i.e. less than 20 minutes) between trains and some of them even are at the minimum transfer time established of 5 minutes (see, for example, transfer from *8xx/1* to *43xx/2* in Figure 7.9). It is coherent with the passenger routing approach that seeks to find the quickest path for every passenger. Furthermore, the transfers that exceed the 20 minutes are uncommon and used by very few passengers.

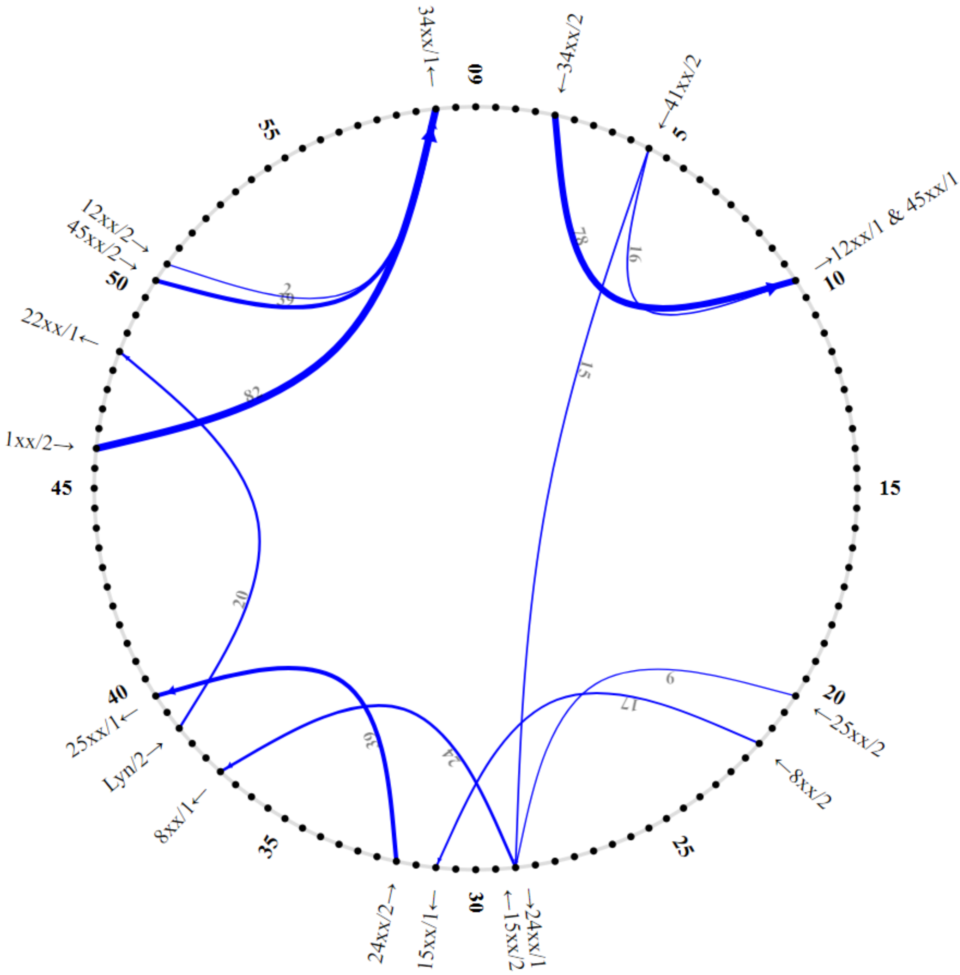


Figure 7.8: Amount of passengers transferring between trains at København H

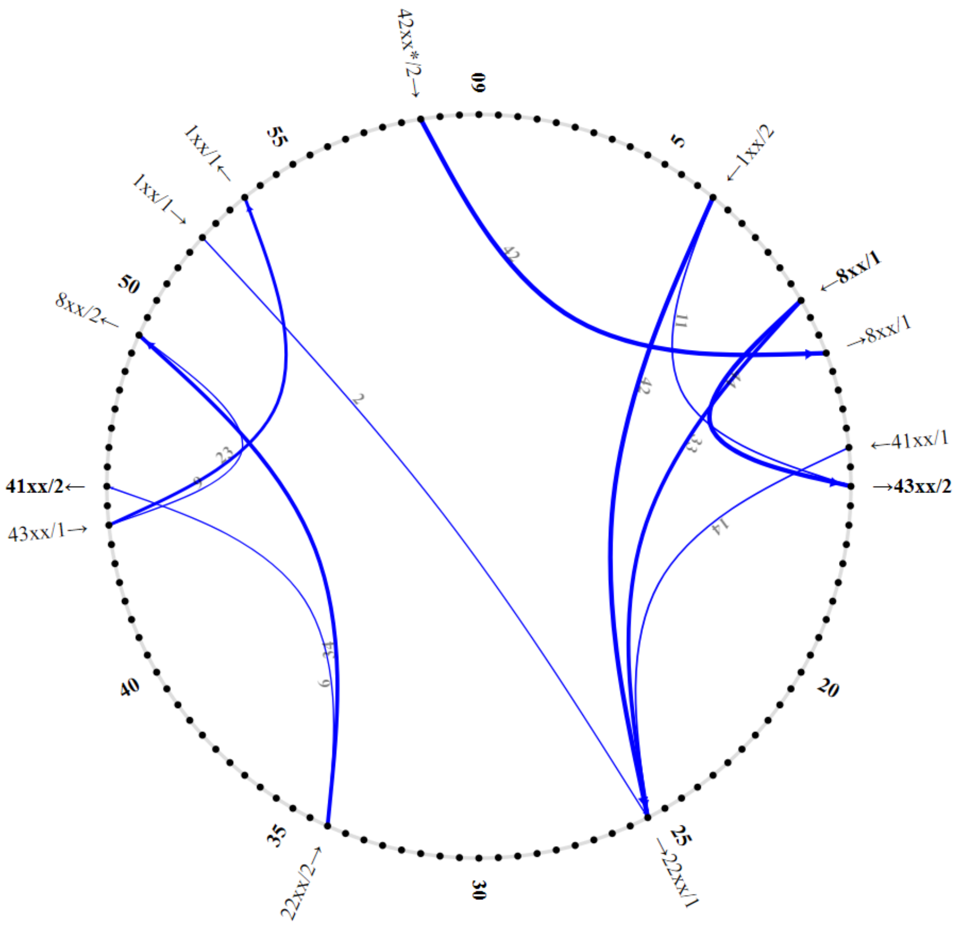


Figure 7.9: Amount of passengers transferring between trains at Ringsted

## 7.2 Parameters to study

When analyzing all the parameters that affect the algorithm and the study itself we can classify them in two main groups: *Instance parameters* and *algorithm parameters*.

### 7.2.1 Instance parameters

These parameters are the ones that affect the settings of the network, create different instance scenarios and modify the solution space. These parameters are:

- Minimum headway between consecutive arrivals and departures at København H. As further explained in Section 7.6.1, KH is seen as one of the most congested stations in the network where all lines stop at and, therefore, the headway at this station becomes interesting to analyze individually. This parameter measures in minutes the minimum interval between consecutive arrivals or departures at København H in the same track segment.

- Minimum headway between consecutive arrivals and departures at any station in the network. This parameter measures in minutes the minimum interval between consecutive train arrivals or departures at each track segment and station in the network.

- Maximum symmetry gap between train paths travelling in opposite directions. This parameter measures, in $\pm$ minutes, the offset allowed between the departure from a station of a train travelling in one direction and the symmetric arrival to the same station from the train in the opposite direction belonging to the same line.

- Minimum headway between consecutive departures of Frequency trains in the same direction from common stations. As it is mentioned in Section 4.3, these pair of trains may have slightly different stopping patterns or running and dwell times. This makes impossible to separate both train paths exactly half an hour during their entire trip. Therefore, a lower bound is needed that should be respected in any station both trains share in common. In this case a minimum headway is defined for the consecutive departures from each station.

- Maximum dwell time at any stopping station in the line. Each line connects two major stations passing by a number of intermediate stations on the way. The train may stop or not at any of those intermediate stations. In the ones that it stops, there are both a minimum and maximum dwell time for each train. The minimum one cannot be modified but the maximum one can be. In this case, a common maximum dwell time in minutes is defined for all stopping stations of each line.

### 7.2.2 Algorithm parameters

These parameters define the behaviour of the algorithm in some of the heuristic steps. They should be tuned in order to find the most efficient version of the solution method. These parameters are:

- Number of transfers analyzed. As explained in Section 6.6.1, once an integer feasible solution is found, the most preferred transfers are checked and the arriving train paths fixed according to the strategy given. Depending on the amount of transfer fixed, the amount of nodes fixed in the train paths increases.
- Minimum transfer time to consider the transfer improvable. If at the end of the algorithm iteration, the transfers analyzed are already very close in time, the new solution will probably not be improved. Therefore, a minimum transfer time interval should be defined for the transfers checked in order to improve them in a new iteration.
- Strategy for choosing which lines to fix among the chosen transfers. When checking the chosen transfers, two strategies have been presented for fixing the nodes of the arriving trains (see Section 6.6.1):
  - All Lines strategy (see Section 6.6.1.1).
  - Independent Lines strategy (see Section 6.6.1.2).
- Diving heuristic strategy. As mentioned in Section 6.4, once a LP fractional solution is found and no more separation constraints are violated, a node in the graph that is used fractionally, is forced to be entirely used in the next solutions. Two different strategies have been presented to choose which node to fix:
  - Sequential strategy (see Section 6.4.1).
  - Random strategy (see Section 6.4.2).

## 7.3 Core instances

Due to the large amount of parameter setting combinations for the algorithm parameters, three main instances are defined that serve as a reference point to tune each of the algorithm parameters. These parameter settings are chosen to resembles different possible real-life requirements and to allow the parameter tuning to be done covering different scenarios. Table 7.1 shows the instance parameters chosen for the three instances defined

Instance 1 can be seen as the instance that satisfies the minimum requirements. There is a minimum headway between consecutive departures and arrivals of 3 minutes in the whole network. Moreover, each pair of the Frequency lines should have departures at least 15 minutes apart at each station they have in common. Desirably, Frequency lines should be 30 minutes apart, however, this is not possible to have at

Table 7.1: Instances used for the algorithm parameter tuning

| **Instance Parameter \ Instance** | Headway København H (min) | Headway Network (min) | Headway Siblings (min) | Maximum Dwell time (min) | Maximum Symmetry Gap (± min) |
|---|---|---|---|---|---|
| **1** | 3 | 3 | 15 | 3 | 1.5 |
| **2** | 4 | 3 | 15 | 3 | 2 |
| **3** | 3.5 | 3.5 | 15 | 4 | 2.5 |

each station due to the different running times of the lines. Therefore, this parameter can be seen as a lower bound to the separation between train paths and, the different running times between the train can only increase or maintain this separation at the stations. The last two parameters affect directly to the complexity of the graph and are only increased in order to increase the solution space for more constraint instances. When two trains cross in a station in a single-track segment, the first train needs to arrive 3 minutes before to the station. Therefore, a minimum dwell time of 3 minutes is required in the single-track stations in order to allow crossings between trains. As the maximum dwell time is considered identical for all the stopping station in this study case, Instance 1 has a maximum dwell time of 3 minutes at all stopping stations. Moreover, a maximum symmetry gap of $\pm 1.5$ minutes is allowed in order to have feasible solutions.

Instance 2 can be seen as the instance for more robust timetables in the most congested area of the network: København H. This instance, increases the minimum headway between departures and arrivals at KH to 4 minutes. In the morning rush hour studied in this case, there are 10 trains arriving to KH per hour through the main-old corridor and 5 trains per hour arriving through the main-new corridor. This sets an upper bound in the maximum headway of 6 minutes between consecutive arrivals. The increment in the headway in KH reduces the solution space as previous solutions that contained a separation of 3 minutes between any two trains in KH are no longer allowed. In order to smooth that reduction of the solution space, the maximum symmetry gap is slightly increased to $\pm 2$ minutes to add more flexibility to the model.

Instance 3 can be seen as the most constrained instance of all three presented. It enforces a minimum headway of 3.5 minutes in the whole network. This increment reduces considerably the solution space and, therefore, it has been decided to increase the maximum dwell time to 4 minutes and the maximum symmetry gap to $\pm 2.5$ minutes.

## 7.4 Parameter tuning

In order to find the best configuration of the algorithm to solve the problem, the algorithm parameters need to be tuned. Therefore, different candidate values that try to cover a broad scope of settings are presented for each parameter. Table 7.2 summarizes the values tested for each parameter.

There is one parameter in the dive heuristic step. As mentioned in Section 6.4, the most fractional nodes can either be fixed in a sequential or a random way.

The other three parameters are part of the destroy method. First, the amount of transfers checked from the solution found. regarding the size of the network, four values have been tested: 1,2,5 and 10 transfers. Second, the minimum transfer time required to consider the transfer as improvable. Four different values have been tested: 5.5 (the minimum interval that can be improved), 7, 10 and 15 minutes. Finally, when the solution is destroyed, the nodes related to the arrival times of the transfers considered are fixed. The nodes of all arrival trains can be fixed to each transfer station or just the ones from lines that have not been considered before.

In total, there are 64 parameter setting combinations. All combinations of parameter settings have been tested on the three instances mentioned in Section 7.3. The algorithm has been tested twice in each parameter setting and instance adding in total 384 algorithm runs. The time limit for each algorithm run has been set to 5 hours taking into account the average time needed for the algorithm to iterate on all three instances. All the test cases have been run in the *DTU HCP* cluster using a *Intel Xeon Processor X5550 (quad-core, 2.66 GHz)* to be able to compare results.

### 7.4.1 Analysis of results

All the average results values for each parameter setting can be found in Appendix A. In order to measure the performance of each parameter setting, different aspects of the algorithm have been checked. The average Passenger Travel Time (PTT) and the sum of path lengths of the best solution of each run have been stored, the number of algorithm iterations, the amount iterations that lead to a feasible solution (i.e.

Table 7.2: Algorithm parameter values used in the parameter tuning

| **Algorithm step** | **Parameter** | **Values** |
|---|---|---|
| Dive heuristic | Strategy for fixing a fractional node | Sequential (1) or Random (2) |
| Destroy Method | Strategy for fixing arriving trains at transfers checked | Independent lines (1) or All lines (2) |
| | Number of transfers checked | {1,2,5,10} |
| | Minimum transfer time to check (minutes) | {5.5,7,10,15} |

feasibility rate), and the amount of times the destroy method actually improved the previous solution and lead to the best solution.

The main optimization goal of this study is to minimize the total PTT in the network. Therefore, this aspect is considered crucial for determining the best setting. The chosen algorithm setting should find solutions, ideally, with the best PTT among all settings. However, there may not be a dominant setting as such. A good approach starts by computing the average PTT for each setting across all three settings. This values are shown in Figure 7.10 where the best average is marked in bold. This setting corresponds to fixing the nodes randomly in the dive heuristic, and considering two transfers with a minimum transfer time of 7 minutes whose arriving train paths are fixed independently.

The average value of PTT for all instances is not a solid estimation and the best setting cannot be chosen based only in that. It can be the case, that the values in the different instances are very irregular. Fortunately, that is not the case and the setting seems to provide good PTT for all three instances (see Figure A.1 on page 118).

The PTT is the main optimization objective but not the only one. Actually, the objective function of the mathematical problem is to minimize the sum of train path lengths. If this value is high means that the trains are dwelling more than usual and are considered "slow", whereas if the values are close to the root node solution value, the trains only dwell the required amount of time and they can be considered "fast". Definitely, another optimization goal is to find solutions whose train paths are the fastest possible. In overall, the problem can be seen as a bi-objective optimization problem where the PTT is more relevant. Therefore, the path lengths for the solution found in the chosen setting should be analyzed. The same procedure is followed where the average across the three instances is checked (see Table 7.11). The chosen setting seems to provide solutions that are both good in PTT and sum of path lengths. When checking the sum of paths length individually per instance (see Figure A.2), the good

| Average Instances | Sequential Dive | | | | | Random Dive | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **minTT\NT** | **1** | **2** | **5** | **10** | **minTT\NT** | **1** | **2** | **5** | **10** | Independet transfer lines |
| | **5,5** | 32,82 | 32,86 | 32,89 | 32,99 | **5,5** | 32,84 | 32,92 | 32,88 | 33,00 | |
| | **7** | 33,05 | 32,87 | 32,91 | 32,93 | **7** | 33,01 | **32,77** | 32,99 | 32,91 | |
| | **10** | 32,90 | 32,84 | 32,88 | 32,97 | **10** | 32,86 | 32,87 | 32,96 | 32,87 | |
| | **15** | 32,98 | 32,85 | 32,91 | 32,92 | **15** | 32,92 | 32,87 | 32,88 | 33,01 | |
| | Sequential Dive | | | | | Random Dive | | | | | |
| | **minTT\NT** | **1** | **2** | **5** | **10** | **minTT\NT** | **1** | **2** | **5** | **10** | All transfer lines |
| | **5,5** | 32,94 | 32,89 | 32,97 | 32,87 | **5,5** | 32,83 | 33,03 | 32,87 | 32,86 | |
| | **7** | 32,89 | 32,90 | 32,81 | 32,88 | **7** | 32,95 | 32,88 | 32,93 | 32,93 | |
| | **10** | 32,99 | 32,98 | 32,89 | 32,90 | **10** | 32,85 | 32,90 | 32,96 | 32,95 | |
| | **15** | 32,96 | 32,89 | 32,98 | 33,15 | **15** | 32,91 | 32,93 | 32,89 | 33,03 | |

Figure 7.10: Average PTT across all three instances for each parameter setting measured in minutes per passenger in a morning rush hour

**Average Instances**

| | Sequential Dive | | | | Random Dive | | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 2012,7 | 2006,2 | 2004,2 | 2020,7 | 5,5 | 2004,7 | 2007,2 | 2011,0 | 2014,5 |
| 7 | 2015,3 | 2004,2 | 2021,7 | 2013,5 | 7 | 2015,8 | 2003,3 | 2008,5 | 2002,7 |
| 10 | 2005,0 | 2004,0 | 2007,3 | 2017,7 | 10 | 2002,8 | 2005,0 | 2009,3 | 2019,8 |
| 15 | 2005,8 | 2004,3 | 2013,8 | 2010,5 | 15 | 2005,5 | 2012,2 | 2012,0 | 2014,3 |

Independet transfer lines

| | Sequential Dive | | | | Random Dive | | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 2008,3 | 2004,5 | 2007,8 | 2004,3 | 5,5 | 2004,5 | 2012,5 | 2005,2 | 2006,8 |
| 7 | 2005,5 | 2015,5 | 2002,8 | 2004,8 | 7 | 2014,7 | 2016,3 | 2007,2 | 2003,8 |
| 10 | 2025,7 | 2005,8 | 2005,5 | 2003,3 | 10 | 2008,5 | 2007,2 | 2020,3 | 2004,7 |
| 15 | 2014,7 | 2004,2 | 2012,3 | 2019,2 | 15 | 2017,7 | 2009,0 | 2007,7 | 2017,7 |

All transfer lines

Figure 7.11: Average sum of paths length across all three instances for each parameter setting

performance of the setting is confirmed.

When looking at the number of iterations (see Figure A.3), it seems to be related to the level of restriction of the instance as the amount of iteration decreases across instances. However, it does not seem to be any parameter value affecting directly to the number of iterations and the fact that each setting was run twice could have emphasized some irregular performances.

Although the sequential dive heuristic seems to have a better reliability of the algorithm in general compared to the random strategy (see Figure A.4), a more detailed level of study should be taken. It can be seen that random diving heuristic combined with fixing the nodes of all arriving trains at transfers provides a very irregular feasibility rate. However, the random dive heuristic combined with fixing only nodes of independent arriving lines at transfers seems to perform better. More specifically, the chosen setting provides a feasible solution in more than half of the algorithm iterations for the first two instances and a 40 % feasibility rate for the third instances. Note that the third instance is the most constrained one and performs fewer iterations. All the setting values of this instance are subject to a possible higher variation and can be considered less reliable than the values of the other instances when evaluating the performance of the settings.

Finally, the effectiveness of the destroy method in the algorithm is debatable. The use of the destroy method is directly related to the feasibility rate of the algorithm run. If a solution is considered infeasible the destroy method is not applied and the algorithm process is restarted. The amount of iterations decreases when the instance provides a more restricted solution space and the feasibility rate becomes also more irregular. As a result, the runs of the algorithm for Instance 2 and 3 usually alternate infeasible and feasible solutions that do not allow the destroy method to perform well. That behaviour is the equivalent to restarting the algorithm directly at each iteration and not applying the destroy method. Moreover, as any solution resulting from the destroy method is accepted, the PTT of the solutions can fluctuate, alternating improvements and worsenings in the solution. As a result, the destroy method can improve the solution several times during an algorithm run but not be able to find the best one in some cases (see Figures A.5 and A.6). In general, there is no clear relation between the usage of the destroy method and good solution values.

### 7.4.2 Best algorithm setting

As a result of the parameter tuning, the following values of parameters and strategies has been decided as the best setting for the algorithm:

- **Random strategy** for fixing fractional nodes in the dive heuristic.
- Consider **two transfers** when an integer feasible solution is found and the passenger flow calculated.
- Consider only transfers that include a **minimum separation of 7 minutes** before the arriving time of the first train and the departure time of the second at the station.
- Fix only arriving nodes for the trains of those lines that were not part of a transfer considered before (**Independent Lines strategy**).

Figure 7.12 shows the average solution values for the three instances where the ones referring to the chosen parameter setting are indicated in orange. They provide fast train path solutions close to the optimal and the PTT is among the bests for all three instances. Whereas for Instance 1 and 2 it does not seem to be an specific relation between train path lengths and PTT, Instance 3 shows a positive correlation between these two measures and it can be seen that higher PTT solutions are a result of longer train paths.

## 7.5 Model performance

Once the best parameters values have been found for the algorithm, the performance of the algorithm can be tested in the three core instances. The algorithm is run 10 more times for each instance and the average values are calculated.

Table 7.3: Average Performance on the algorithm for the core instances

| **Instance** | **It.** | **PTT (min/pass.)** | **Paths length (min)** | **Root Node Solution** | **Feasibility** |
|---|---|---|---|---|---|
| 1 | 15,8 | 32,89 | 2010,5 | 1998,5 | 55% |
| 2 | 7,8 | 32,91 | 2015,9 | 1997,5 | 57% |
| 3 | 6,2 | 32,88 | 2005,4 | 1997,5 | 36% |

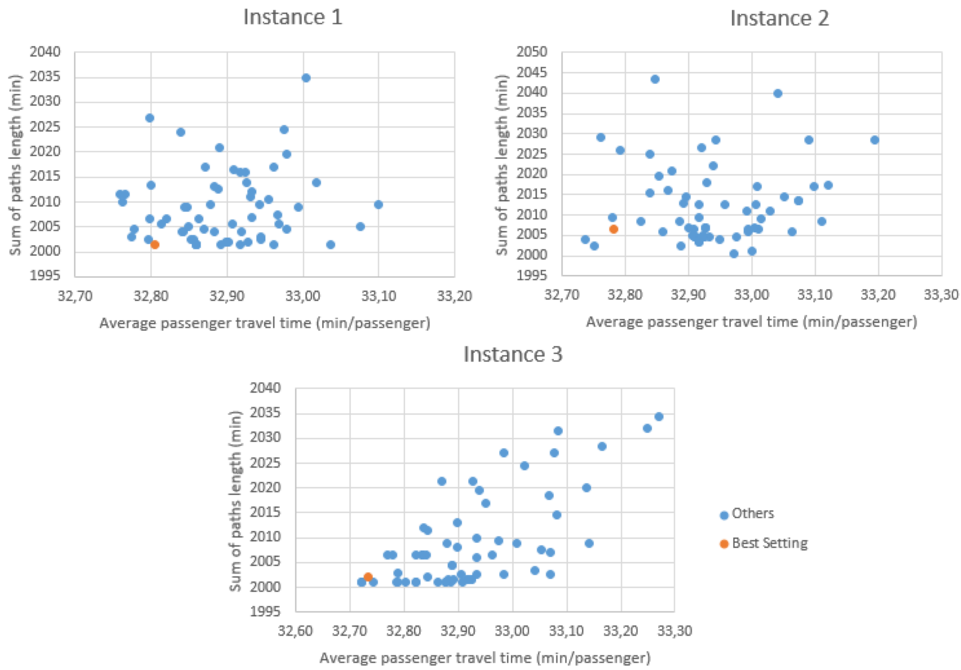


Figure 7.12: Average performance of the runs related to the chosen parameter setting for the three instances

Table 7.3 shows the average performance of the algorithm in the three instances. The number of iterations decreases as the restrictions of the instances increase and coincide with the ones calculated in the parameter tuning. This, combined with the feasibility rate and the algorithm running time established, allows to find an integer feasible solution roughly every 34 minutes for Instance 1, every 67 minutes for Instance 2 and every 134 minutes for Instance 3.

The average PTT is similar for the three instances and they appear to be greater than the values obtained in the parameter tuning. This suggests that the two runs of the parameter tuning somehow were a bit "lucky". More runs per setting would have given more reliable results for the parameter tuning. However, due to time constraints this was not possible and the average PTT values seem to still be way above the average of all setting solutions.

The paths length is slightly greater for the first two instances compared to the third one but all of them are not far from the root node solution. The higher variation of the paths length could be mainly due to the fact that the best solution is uniquely chosen in relation to the PTT regardless of the paths length associated to it. The solution may have slow trains that are not used by any passenger but that are counted towards the total sum of paths length. However, it has been shown in the parameter tuning that this setting is able to find solutions with near to optimal path lengths. Finally, the feasibility rate is in concur with the estimation from the parameter tuning.

Paradoxically, Instances 2 and 3 provide a lower solution at the root node (1997.5 min) compared to the one of the Easy instance (1998.5 min). The solution at the root node can be seen as the optimal LP solution after finishing the first CG procedure. Although the headway is greater, this does not necessarily increase the length of the paths. The fact that the symmetry gap is increased, enlarges the arc set of the graphs allowing more line train paths combinations (resulting in a harder to solve graph as well). This, in the single-track segment allows the train paths of the same line to cross with a bit more flexibility. When two train cross on a single-track, one train always needs to wait at least three minutes until the other train arrives. This requirement, if the symmetry gap is tight, can make the other train dwell longer in order to be symmetrical. Then, if the symmetry gap is increased, the dwell time of the train can be more flexible in a crossing station. That is the case of Instances 2 and 3 where the symmetry gap is higher than the one of Instance 1.

For analyzing the inner behaviour of the algorithm, the total amount of iterations of the different inner loops required to find a solution (or to discard it as infeasible) are computed. These values are displayed in Table 7.4. It can be seen that the level of the restriction of the instances do not affect considerably the number of inner iterations. In general, Instance 1 requires fewer iterations of each loop per algorithm iteration. This corresponds with the level of the restriction of the instances. It seems that Instance 2 is the one requiring more columns and iterations in general, although it is able to run the iterations faster than Instance 3 as seen from the total amount of algorithm iterations. In overall, it does not appear to be a performance change in the algorithm performance across the instances with the exception of the time required to perform an algorithm iteration.

The number of heuristic iterations measure the level of depth of the branch-and-bound tree explored by the Dive heuristic. The values are not high suggesting that the problem could be solved to optimality applying a *branch-and-price* algorithm. However, that the tree is not too deep does not mean necessarily that the problem can be solved to optimality. The tree can have a lot of branches or be highly unbalanced. For instance, in this case, forcing a node to be used (i.e. $z_v^l = 1$) implies reducing considerably the solution space and has a much greater impact than forcing it not to

Table 7.4: Number of inner loops iteration needed and columns generated per algorithm iteration

| **Instance** | **Heuristic Iterations** | **Separation Iterations** | **Column Generation Iterations** | **Number of Columns** |
|---|---|---|---|---|
| 1 | 6,8 | 21,9 | 201,3 | 545,3 |
| 2 | 8,2 | 27,5 | 259,9 | 770,9 |
| 3 | 7,9 | 25,8 | 221,0 | 678,1 |

be used (i.e. $z_v^l = 0$).

When looking at the number of columns needed per iteration, it is also interesting to look from which PP are these columns mainly coming from. Figure 7.13 shows the average distribution of columns per line at each algorithm iteration for Instance 1. More than half of the total columns belong to lines that run along the North-West corridor. This is related to the fact that at the single-track segment of this corridor, is the only place where a crossing between trains of different lines can occur. In order to cross, one of the two trains needs to dwell for three minutes in one of the stations resulting in a ”worse” path length. Because the Crossing constraints are added by separation once they are violated, the column generation process prioritizes to generate fast paths that may cross each other until all the possible crossing violations are added. This results in a large amount of columns. It is also remarkable to mention that 94% of the columns belong to lines using the Main-Old corridor. The reason for this relies, apart from the one mentioned before, in the quadruple track segment. Allowing two routes for the trains means that identical timetables can be found where the split of the lines in the quadruple track is different. This, enlarges the possible combinations and, therefore, the number of columns. It is known from DSB that there are some restrictions on the track to choose depending if the train is stopping along the quadruple segment or not. Nevertheless, those restrictions are not considered in this study but may be considered for further studies (see Section 8.3).

In order to analyze better the performance of the algorithm, the most time consuming processes are identified and quantified. Figure 7.14 shows the distribution

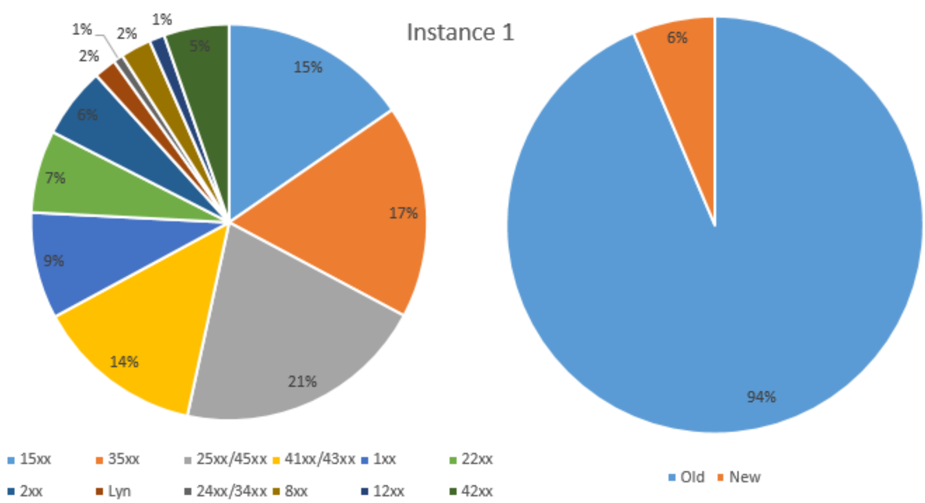


Figure 7.13: Average columns generated for each line (left) and average columns of lines using the Main-Old and Main-new corridor (right)

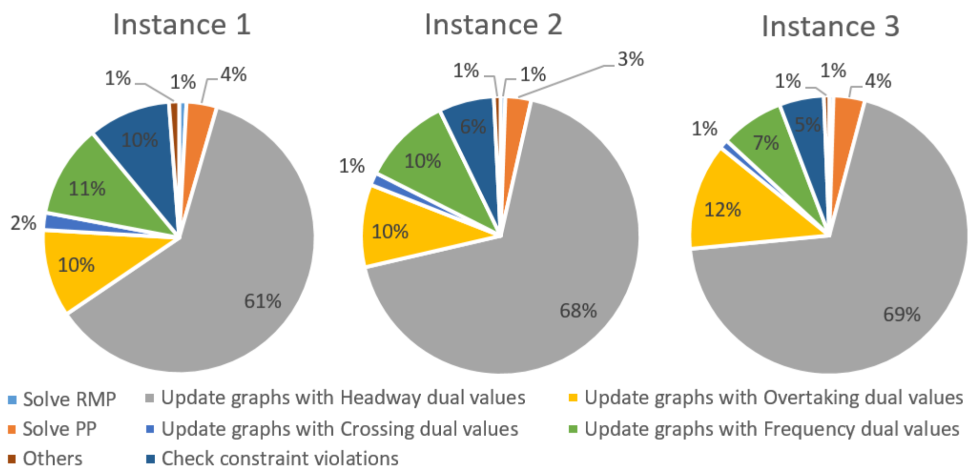


Figure 7.14: Main computational processes of the algorithm for each instance

of the time at each algorithm iteration. The pie chart differentiates the time spent solving the RMP and PPs, the time spent updating the PP graphs with the dual values of each of the type of constraints and the time spent looking for constraint violations in the separation procedure. The distribution is similar throughout the three instances and the most time-consuming part of the algorithm is definitely the process of updating the non-zero dual values of the Headway constraints in the RMP in the affected PP graphs. The dual value for a specific track segment needs to be updated in all the graphs whose lines can use that track segment. Depending on the density of the graphs this process can be hard to do efficiently. In general, when applying column generation techniques, most of the time is spent solving the RMP and PP. Although updating the weights of the graphs can be seen as part of solving the PP, this process can definitely be fasten and should be further studied (see Section 8.1.1).

### 7.5.1 Comparison with manual timetable

Another method to measure the performance of the algorithm is by comapring it with the timetable that DSB has planned manually. Table 7.5 shows the solution values of DSB's solution using our approach of passenger travel time calculation. The solutions achieved are better than the ones calculated by DSB in both aspects.

Table 7.5: Solution values of the manually planned solution by DSB

| TTP (min/pass.) | Paths Length ( min) |
|---|---|
| 33,21 | 2049,5 |

However, this comparison may be misleading as there are some differences in the scenarios considered.

First, the planned track upgrade from single-track to double-track for the segment between VO and NF is not considered by DSB. This means that DSB considers that segment as a single-track. This may result in a increment of the path length due to the additional dwell time required by the trains to cross at the station.

Second, due to the on-going implementation of the new signalling system in the Main-New corridor, DSB considers a temporary higher headway in this corridor that may restrict more the solution.

The timetable planned by DSB can be considered as a totally feasible one, meaning that it also takes into account possible conflicts that are out of the scope of this study. These additional conflicts may affect to the solution considerably.

On the other hand, the timetable considers a minimum headway of 2.5 minutes for crossing at station that is part of a single-track instead of the 3 minutes considered in this study. This decrements allows slightly faster trains paths for those lines.

The track segment between KJ and ØLB is a single-track segment. However, when leaving KJ the two platforms tracks are prolonged a few hundred meters until they are merged into the single-track. DSB takes advantage of this double-track "segment" to allow a crossing between the trains. In this case study, that crossing is not allowed.

## 7.6 Robustness analysis

As already mentioned in section 4.4, one of the main aspects that DSB takes into account, when evaluating the quality of a timetable, is the robustness of it. According to the Oxford dictionary (Stevenson, 2010), robustness can be defined as "the ability to withstand or overcome adverse conditions or rigorous testing". In railway transportation the main disturbance affecting the robustness of the timetable are train delays. There are two levels of delays. *Primary delays* are the first ones happening and are mainly caused by external factors (i.e. bad weather) and machinery or signal failure, whereas *secondary delays* are the ones caused by the primary ones. The latter commonly happens in highly congested networks where the primary delay of a train can easily propagate to other trains. As a result, a robust timetable can be defined as the one that remains unaltered after small delays and does not allow them to propagate easily along the network.

The three instance parameters measuring headway between arrivals or departures are analyzed to create different scenarios. The parameter values are variated keeping the other ones fixed based on the setting of Instance 1. In some cases, for specially restricted cases (i.e. when increasing the headway on the whole network), the maximum dwell time at stations and maximum symmetry gap may be increased proportionally in order to facilitate finding feasible solutions, although the computation time increases accordingly.

### 7.6.1 Headway at København H

Taking a closer look at the network of this case study, all the lines in the network have an endpoint at KH. This station can be seen as the most congested point in the network and, therefore, it becomes an important focus point of the robustness analysis.

In order to measure the robustness at this station, different instances are created considering different minimum headways at KH. Each setting is run 10 times using the tuned algorithm and maintaining the same running time of five hours.

Table 7.6 shows the results for the different values of headway at KH where it can be seen that the model is able to find solutions for all the headway values including the maximum headway allowed of 6 minutes. The solution values worsen when the headway increases. For headways below 5 minutes the solution value detriment does not seem to be significant and the algorithm is able to find relatively good solutions with a number of columns similar to the ones of Instance 1. For high headway values, the solutions are worse in terms of PTT and paths length. Also the feasibility rate decreases considerably. If compared with the manual solution from DSB it can be seen that even the solutions achieved with the highest headway provide better solution values. Nevertheless, both cases cannot be strictly compared and should be taken as an orientation value.

Figure 7.15 shows that for small headways, the distribution of the arriving or departing trains in KH is unbalanced sometimes. During morning rush hour, there are three fewer trains departing per hour along the Main-Old corridor compared to the ones arriving. The lack of a symmetrical train for the rush hour lines creates larger time interval without any departure reaching sometimes more than 15 minutes between consecutive departures. As the headway is increased in KH, both the departures and arrivals are more homogeneously distributed. The amount of trains arriving during one rush hour is 10 meaning that a maximum headway of 6 minutes can be imposed. The model is able to find feasible solutions for 6 minutes of headway as it shown in Figure 7.15, where the homogeneity and robustness of the arrival and departures at KH can be confirmed.

Table 7.6: Average performance of the algorithm for different headway values at KH

| **Headway at KH (min)** | **It.** | **PTT (min/pass.)** | **Paths Length (min)** | **Avg. columns per solution** | **Avg. CG It. per solution** | **Feasibility** |
|---|---|---|---|---|---|---|
| 3,5 | 14,8 | 32,84 | 2007,4 | 546,5 | 186,1 | 62% |
| 4 | 11 | 32,96 | 2008,0 | 683,8 | 233,8 | 47% |
| 4,5 | 9,8 | 32,95 | 2013,8 | 741,4 | 232,6 | 51% |
| 5 | 8,2 | 33,01 | 2017,2 | 885,1 | 251,3 | 44% |
| 5,5 | 5,7 | 33,01 | 2023,1 | 1140,3 | 305,7 | 32% |
| 6 | 4,4 | 33,10 | 2019,9 | 1315,5 | 324,9 | 31% |

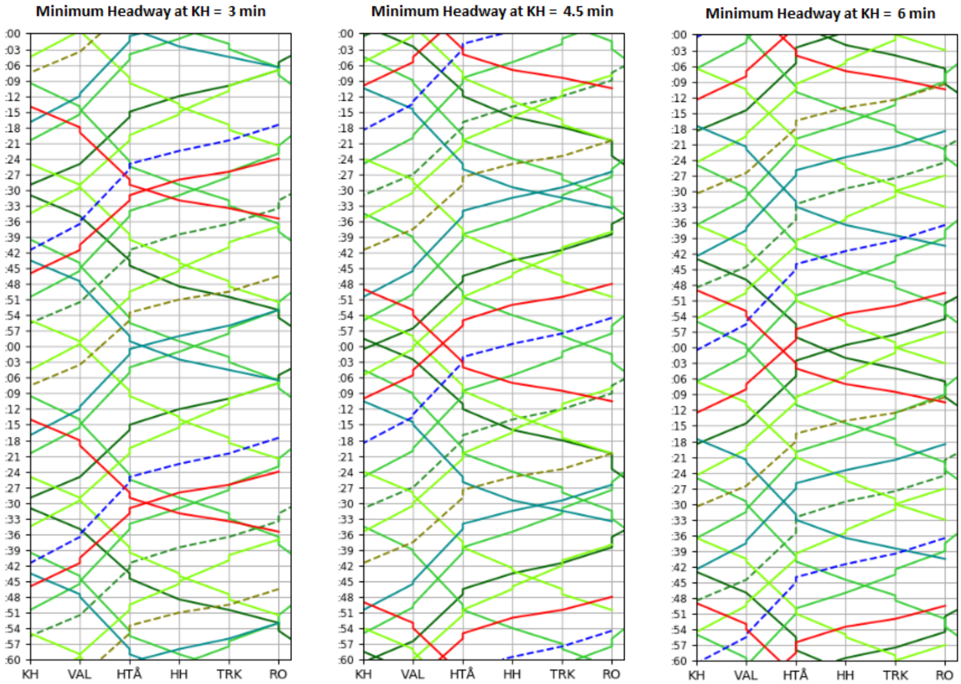


Figure 7.15: Three solution examples for the corridor between KH and RO considering different minimum headways at KH

## 7.6.2 Headway in the whole network

Another way of measuring the robustness of the timetable can consider increasing the headway at all stations in the network. The level of robustness analyzed here is much stronger than the one analyzed before because by increasing the headway in the whole network allows to absorb any primary delay in any part of the network, whereas just increasing the headway in KH enhances the robustness of only that station. However, the problem becomes much more constrained.

Table 7.7 shows the average results of headway values considered. It is quickly

Table 7.7: Average performance of the algorithm for different headway values in the network

| Headway at network (min) | It. | PTT (min/pass.) | Paths length (min) | Avg. columns per solution | Avg. CG It. | Feasibility | Max. dwell time (min) |
|---|---|---|---|---|---|---|---|
| 3,5 | 9,2 | 32,96 | 2021,8 | 691,4 | 221,7 | 20% | 3,5 |
| 4 | 6,2 | 33,14 | 2040,8 | 789,7 | 227,8 | 13% | 4 |
| 4,5 | 3,1 | 33,33 | 2055,5 | 1327,4 | 360,1 | 16% | 4,5 |

noticed that the solution quality worsens considerably when the headway is increased. Although the algorithm is able to find solutions for a headway of 4.5 minutes in the whole network, both the PTT and paths length are poor. On the other hand, a solution with an overall headway of 3.5 minutes seems achievable without significant worsening of the solution. The study of this parameter suggests that considering a common overall headway in the whole network is not a good idea and a more individualized approach should be consider where the more congested stations are prioritized over the less busy ones.

### 7.6.3 Separation of Frequency lines

The lines with similar stopping patterns whose train paths are wished to be as separated as possible along the cycle time, are constrained by a general minimum headway at each station. The lower bound considered for the three core instances has been 15 minutes. However, the existence of solutions with a higher minimum separation can be tested. Different scenarios have been tested increasing this parameter:

Table 7.8 shows the results for the different values of the parameter studied. The algorithm seems to perform similarly until headway values of 26 minutes. The PTT values are stable and good as well as the paths length Above 26 minutes the results worsen and no more feasible solutions are found for headways of 28 minutes or more. In overall the model allows large variations of this parameter without worsening the solutions. There are only three pair of lines that require this separation and apparently their separation can be handled efficiently without affecting the paths of the other lines.

Table 7.8: Average performance of the algorithm for different headway between Frequency line departures

| Headway for Freq. lines (min) | It. | PTT (min/pass.) | Paths length (min) | Avg. columns per solution | Avg. CG It. | Feasibility |
|---|---|---|---|---|---|---|
| 16 | 14,6 | 32,90 | 2009,5 | 504,2 | 185,1 | 54% |
| 17 | 13,8 | 32,86 | 2007,9 | 539,5 | 192,4 | 51% |
| 18 | 18,9 | 32,88 | 2006,3 | 459,3 | 170,9 | 37% |
| 19 | 17,3 | 32,90 | 2006,3 | 483,8 | 181,3 | 39% |
| 20 | 20,1 | 32,82 | 2008,2 | 465,2 | 163,1 | 43% |
| 21 | 17,5 | 32,91 | 2006,1 | 502,3 | 172,1 | 41% |
| 22 | 23,8 | 32,92 | 2005,9 | 402,7 | 140,7 | 37% |
| 23 | 21,9 | 32,84 | 2004,4 | 448,1 | 158,3 | 44% |
| 24 | 19,9 | 32,87 | 2007,8 | 452,3 | 148,5 | 47% |
| 25 | 18,3 | 32,91 | 2008,6 | 460,5 | 153,6 | 44% |
| 26 | 17,8 | 32,95 | 2016 | 474,6 | 146,1 | 47% |
| 27 | 21,4 | 32,89 | 2014 | 413,5 | 127,1 | 22% |
| 28 | 29,3 | - | - | 299,2 | 97,1 | 0% |

### 7.6.4 Additional scenarios

Apart from the analyzed tests, further scenarios have been tested and the results are summarized in Table 7.9. The scenario for Instance 1 has been tested increasing the algorithm running time to 2 days (48 hours) The solution values are slightly better in terms of PTT and suggest that by increasing the algorithm running time can lead to better results.

Moreover, a test has been conducted considering the combination of some of the robust parameter values that the algorithm managed to find a feasible solution for. In this case, it has been tried to find a solution with a headway of 6 minutes at KH and a minimum separation between Frequency trains of 25 minutes while keeping the rest of the parameters fixed to the ones defined for Instance 1. The results show that feasible solutions are achieved but with a relatively poor solution quality. As a result, it can be confirmed that requiring high robustness in the solution decreases the quality of the solution in terms of passenger travel time and paths length. Moreover, the feasibility rate suggests that feasible solutions for more combinations of robust parameter values can be achieved.

One last test has been conducted without applying the destroy method and just restarting the algorithm after each iteration. The results provide good solution values that are similar to the ones calculated applying the destroy method. Therefore, these results suggest that the algorithm performance is similar with and without including the destroy method. However, the destroy method could be used for other purposes as it is discussed in Chapter 8.

Finally, in order to have an overall perspective of the quality of the scenarios tested, the solution values of all scenarios are shown in Figure 7.16.

Table 7.9: Average performance of the algorithm on the additinal scenarios considered

| **Scenario** | **It.** | **PTT (min/pass.)** | **Paths length (min)** | **Avg. columns per solution** | **Avg. CG It.** | **Feasibility** |
|---|---|---|---|---|---|---|
| 48h running time (Instance 1) | 157,5 | 32,77 | 2009,5 | 520,6 | 189,5 | 51% |
| Headway at KH = 6 min<br>Headway Freq lines = 25 min | 3,9 | 33,15 | 2037,8 | 1381,4 | 301,2 | 39% |
| No destroy method | 17,7 | 32,90 | 2004,3 | 494,0 | 187,9 | 49% |

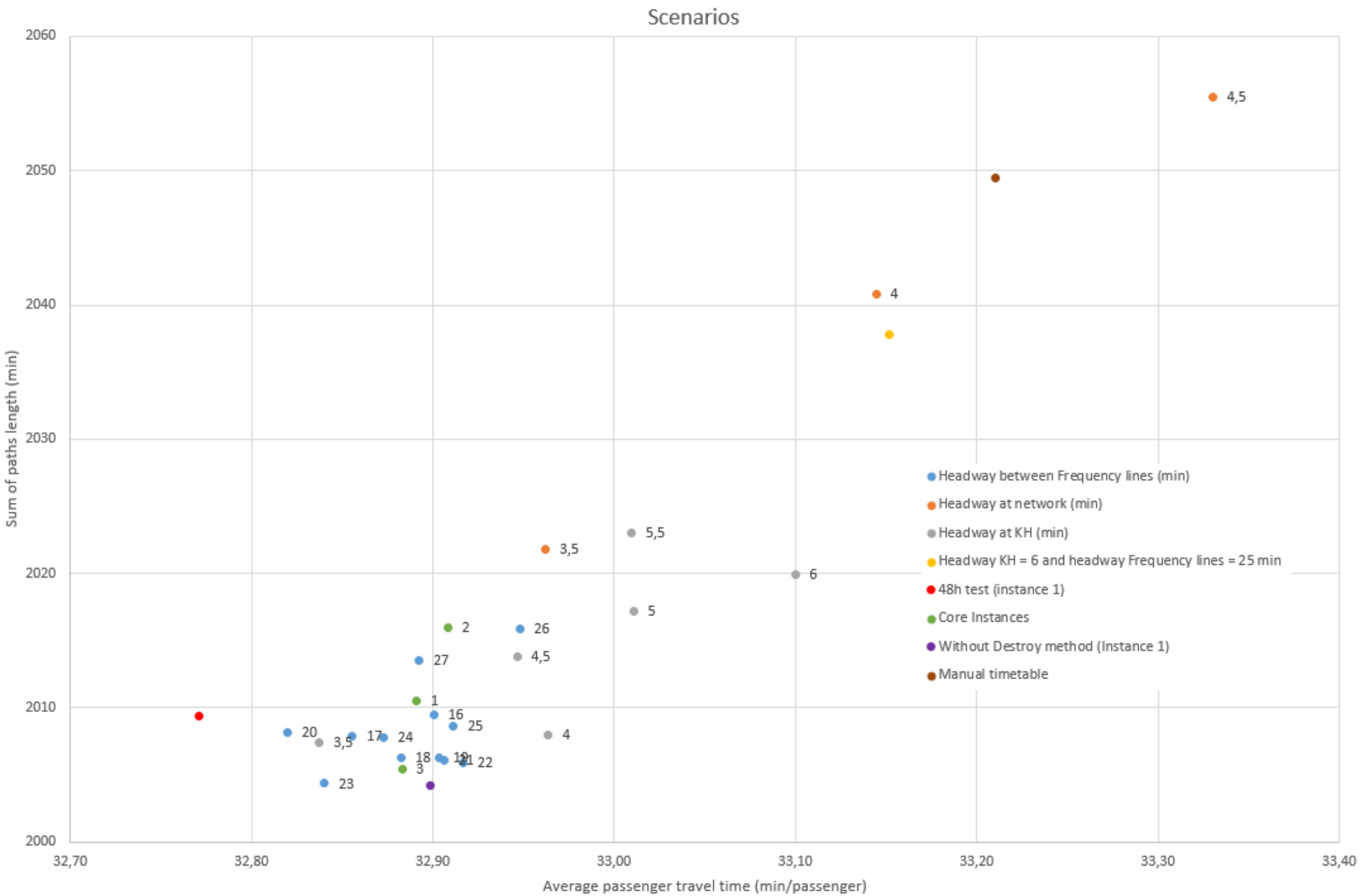


Figure 7.16: Comparison of all scenarios considered

## 7.7 Performance of solution method variants

This section briefly comments the performance of the solution method variants described in Section 6.7.

The method where the PP is solved as an MIP model that combines the train graph formulations for both directions has been implemented and tested. The solution method has shown to be very time consuming. The integer nature of the PP made it hard to solve especially for lines that share a single-track segment, spending more than a minute in some cases. Moreover, the less conservative approach of just enforcing symmetry in a few stations of the line increased the amount of feasible solutions and, logically, the arc density of the graph. It has been initially tested just for the lines running in the North-West corridor and the estimated average time to find an integer feasible solution for the three lines exceeded the hour. Therefore, the solution method has been discarded and not tested in the whole network.

The idea of starting with an initial pool of ”fast” line train paths has not yield any apparent benefit neither. Due to the symmetry gap allowed and time period, the amount of initial ”shortest paths” for each line is large (roughly $O(2|T|K)$ and the double if the line is using the main old corridor) and the needed time to create them increased considerably. If a subset of them was considered (i.e. the line train paths exactly symmetrical between them), they were almost never part of the final solution, as the symmetry gap is usually used in the solutions to add flexibility to the solution and accommodate the combinations of lines easier. Therefore, it has been decided not to include any initial pool of line train paths in the solution method.

The suggestion of creating an MIP model that extends the current formulation by including the transfer times has shown more promising results. The model was tested by running the algorithm for one iteration until finding an integer feasible solution. Then, the model was extended including the variables to measure the transfer time and converted into an MIP model. In few best cases, the new solutions found, had an improvement of up to 1 % in the passenger travel time which can be considered significant. On the other hand, some cases shown that there was only one optimal solution, and the method did not provide any gain. Furthermore, the integer nature of the problem made it really hard to solve sometimes, especially when the number of columns and the constraints added by separation was high. As a result, this method does not seen attractive to be combined it with the proposed solution method for this study. Nevertheless, it showed potential for providing promising results and should be definitely considered for further studies.

Finally, the strategy proposed to avoid degeneracy has not shown any benefit. After a defined number of CG iterations, if the RMP objective value has not shown any improvement, the CG process is forced to terminate. Depending on the limit of iterations defined, two different outcomes were seen. If the number of iteration set as a limit was low (i.e. less tan 15) then the model tended to infeasiblity more often. If it was too high (i.e. more than 15), then the degeneracy was still present and no apparent gain is achieved. Moreover, from the results analyzed it is noticed that the amount of CG iterations performed is not significantly high and that the degeneracy cases are rather sporadic. Therefore, this strategy has not been further tested.

# CHAPTER 8
# Discussion

This chapter conducts an overall discussion about the model and the case studied. The discussion is classified in different topics and for each of them, the limitations of the model are described as well as possible improvements and further studies are suggested.

## 8.1 Modelling

It has been shown that the model is able to find good solutions to the network relatively fast. However, several aspect of the model can be further improved.

### 8.1.1 Algorithm running time

As it was shown in Section 7.5, more than 60% of the time of the algorithm running time is spent updating the weights of the PP graphs with the dual values related to the Headway constraints. Although this process has been accelerated several times along the implementation of the algorithm, there is still room for improvement. Currently, due to programming requirements, the dual values are included in all the out-going arcs of the related node. A way of boosting the process could be to include the value in the node itself. In this way, iterating over all the out-going arcs is not needed anymore leading to a significant save of time.

### 8.1.2 Extension of the problem formulation

The formulation of the problem solved by the current model only considers the minimization of the path lengths and the algorithm only includes the transfers implicitly based on the previous feasible solution. Since optimizing the PTT is the main focus of this study, including the transfer times in the problem formulation can imply a great advantage. As described in Section 6.7, an extension of the formulation is already presented. Nevertheless, it has been only tested as a post-process improvement rather than including it in the main formulation. The few tests performed showed a great potential for this type of formulation.

### 8.1.3 Destroy method

The route of the passengers can change completely from one algorithm iteration to the next one. Forcing to have a transfer, that in the previous solution had a large flow, closer in the new solution, can require enlarging the train path in another segments and making the transfer not attractive anymore. Therefore, it cannot be said that there is a clear connection between the destroy method and the improvement of the PTT. However, the destroy method can be used with a different purpose when used in the planning system. Instead of using it as a destroy method that keeps the already found transfers in the next solution, it can be used as a tool to introduce specific transfer priorities. Similarly to the different headway scenarios tested, the destroy method can be used to test different transfer preferences in the network. In fact, the destroy method has already found good solutions were some transfer stations resemble a *hub* where a lot of possible transfers can be done within a small time interval.

The level of destruction of the solution is quite high. Practically the complete solution is destroyed at each iteration confusing the procedure with the Greedy Randomized Adaptive Procedures (GRASP) (Feo and Resende, 1995). However, the solution is not constructed using a greedy procedure and the destroy method does not destroy completely the solution. Still, implementing the model as a GRASP procedure could be explored given the performance of the destroy method.

Another option can be to include a second destroy method that destroys the solution in a different way. Examples of alternative destroy methods can consider, for example, only destroying specific lines or parts of them. The options are innumerable. Including multiple destroy or repair methods transforms the model from a LNS to an Adaptive Large Neighborhood Search (ALNS) (Røpke and Pisinger, 2006). This method keeps track of the performance of the destroy and repair methods and prioritizes the ones that suit better to the problem. An idea of combining multiple destroy methods sounds interesting, however, these type of methods are more suitable for processes with a large amount of iterations which is not strictly the case.

## 8.2 Train running time

The model is based on the assumption that the running times between stations are fixed. This approach, although can resemble real scenarios reasonably well, does not contemplate timetables where the trains can slow down at some track segments. Both the Overtaking and Crossing constraints from the problem are formulated based on fixed running times. In the case of considering flexible running times, the set of incompatible departures of two trains do not longer depend on a parameter but on a variable. As a result, the currently formulated constraints become non-linear and need to be adapted. This adaptation would probably incur in a much larger set of constraints. Nevertheless, including flexible running times, allow to explore a much larger solution space and the margin of operation could be larger.

## 8.3 Track allocation

When two corridors join in one station, depending on the direction and route of the passengers, the train may inevitably need to cross a track segment while departing or arriving at the station. These type of conflicts are mainly happening at Ringsted station. This station can be seen as a junction of four corridors (Main-Old, Main-New, Great Belt and Large-South corridors). Figure 8.1 gives an overview of possible tracks that a train may need to cross when passing by Ringsted. Definitely, the figure does not represent exactly the infrastructure at the station but gives an overview of the possible track-crossing conflicts that may occur. In order to have a better view, the connections of the tracks with the platforms should be considered. Other stations where there could be similar conflicts are RO and KH where multiple corridors meet. These track crossing conflicts are not considered in the model developed in this thesis. As a result, part of the obtained solutions in this case study may probably not be feasible in a real scenario that takes into account these conflicts. Two approaches are suggested to try to solve this issue.

First, the graph formulation can be slightly enlarged by adding nodes that represent the station where track-crossing may happen in a more detailed level. The different track-crossing points can be denoted and only routes that allow a feasible routing of the trains are allowed.

Second, an adaptation of the Headway constraint in the single-track for trains in opposite directions can be formulated. This constraint would only apply to pairs of trains that can end in a track-crossing conflict at the specific station and would enforce them to arrive to the station with a defined headway time. This second approach does not need to enlarge the set of nodes of the graph.

Another possible conflict regarding the track allocation, lays in the quadruple-track segment between Høje Taastrup and Roskilde. The trains that run through this segment can be classified in two groups: the ones stopping at intermediate stations (i.e. stopping trains) and the ones that do not stop during the quadruple segment (i.e. non-stopping trains). Due to restrictions in the structure of intermediate stations, the stopping trains are preferred to use the outer track whereas the non-stopping ones should use the inner one. This requirement is not strictly enforced in the model although the solutions obtained tend to naturally fulfill this requirement. The differentiation of tracks is only needed when two trains travel at the same time along the track segment in the same direction. The path length along the quadruple-track segment for the stopping trains is larger due to the dwell time at the intermediate stations. For instance, consider two trains that start travelling along the quadruple-track in the same direction, one in each track. If the two trains are from the same type (stopping or non-stopping) it is most likely that they also arrive at the same or similar time to the other end of the quadruple-track segment. Then, as the quadruple-track is ended, and, if they continue along the same corridor, one of the two trains would be required to dwell longer in order to respect the departure headway. If two trains of different type start at the same time, their difference in path length along the quadruple track segment would make that the trains arrive at

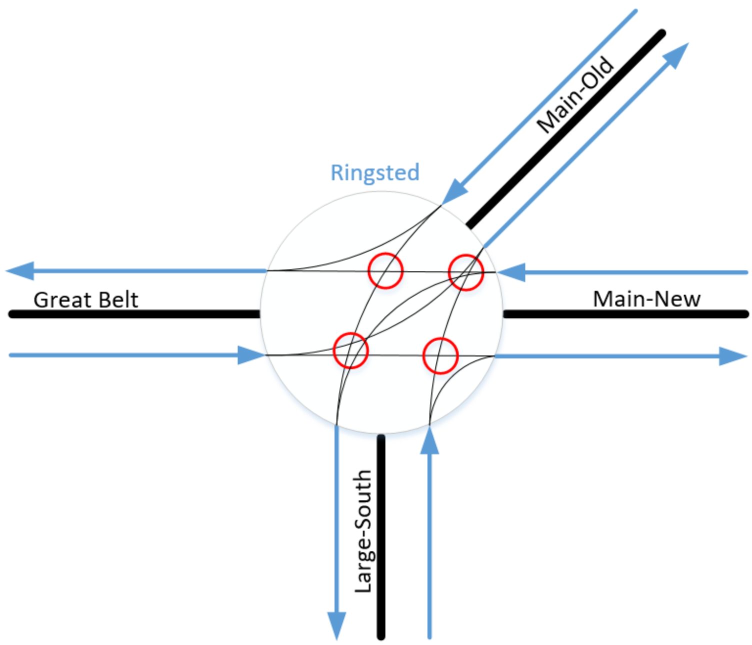


Figure 8.1: Overview of possible track-crossing conflicts at Ringsted station

different times at the other end of the segment and probably do not require to dwell longer. An example of this is illustrated in the timetable example displayed in Figure 7.1 for the trains *15xx* and *41xx/43xx* travelling towards KH that depart at the same time from RO.

## 8.4 Passenger routing

Ingvardson et al. (2018) analyzed the arrival and waiting time of passengers at stations in the Great Area of Copenhagen and showed that the share of passengers arriving randomly decreases as the headway increases (see Figure 8.2. Therefore, in the high congested parts of the network (KH-HTA-RO or KH-KJ), passengers will tend to arrive at random times more often as they are aware that a train may depart within a few minutes. However, for the less congested parts of the network where the frequency of trains is not that high and specially stations with a low train service level, people will tend to arrive closer to the actual departure time of the train. As a result, the proposed model to route the passengers seems more accurate for the less congested areas than for the more congested ones. That said, the occupancy of the trains in the high congested parts of the network may be highly unbalanced and

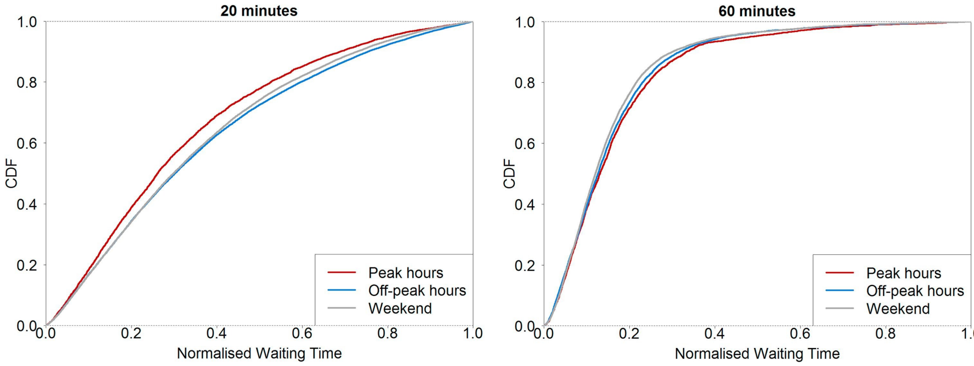


Figure 8.2: Difference between passenger waiting time distributions for headways of 20 and 60 minutes (Ingvardson et al., 2018)

require a smoother distribution of passengers along the cycle time. Currently, it was assumed that passengers where arriving at the station at the exact time their train was departing, nevertheless, the model can be easily extended by distributing the demand at specific times along the time cycle. The origin arcs mentioned in Section 6.5.1, instead of linking the artificial station source node to the departure of the trains at that station, the source node can be linked to specific time instants along the hour and assign a proportional demand to each of them. Nevertheless, in order to create an accurate demand distribution, how the demand fluctuates during the rush hour lines should be further studied.

Furthermore, the passenger routing problem is currently modeled as a Multi Commodity Network Flow without capacity constraints that can be solved by just computing the shortest path between each pair of stations. Although the capacity of the trains may not be a major issue taking into account the usual demand and the capacity of the lines operating in the network, it could be interesting to model the problem considering the capacity of the trains. This could probably result in a more balanced and coherent routing of the passengers but the computational time is expected to increase due to the additional capacity constraints.

## 8.5 Passenger demand

An accurate passenger demand estimation is crucial for defining a good timetable. in this study, the passenger demand between stations for a morning rush hour was based on an annual forecast for 2022 that can be improved.

First, the passengers travelling within the network were the only ones considered. Meaning that passengers travelling from or to Jutland, Germany, Kastrup/Sweden or the Coast line were not taking into account in the computation.

Second, it was assumed that the passenger travelling between two stations were only using the transport systems considered (i.e. Regional or InterCity lines). However, there are some segments of the network that can be reached by S-train as well such as the segment between KH and HTA. Although the Regional and InterCity trains are faster, the S-train provides a higher frequency of trains, becoming more attractive for some people.

Moreover, the study has not considered the effect of any other transport system in the network. Usually, many passengers arriving to KH usually transfer to the S-trains to reach their final destination. It is assumed that the high frequency of the S-train already ensures a good transfer between transport system. However, there are other transport systems that could be considered such as bus lines at certain stations or other trains (i.e. *Arriva* in Jutland).

Currently, the hourly demand has been considered as a symmetric demand of passengers. However, the study time covered concerns the morning rush hour. It is expected that in this time slot more passenger are travelling towards KH than out of it. Therefore, a proper adaption of the OD matrix should be considered for a more accurate study.

## 8.6 Network

The network considered covers most of the daily regional and InterCity railway traffic in Zealand. However, it could be interesting to include more lines and consider a larger network. The model shows potential to solve larger instances and can be tested, for example, including the missing Coast line and Kastrup line. A more ambitious target could be to consider the whole Danish network for Regional and InterCity trains.

### 8.6.1 Individual station settings

Regarding the minimum headway between arrivals and departures, an overall value has been decided that is applied to all track segments, independently of their length or speed limit. Headway values can be defined individually for the stations. In this way, the most-congested stations can consider a higher value than the less congested ones. This has already been done but just for KH and it could be extended to other stations. Liu and Han (2017) already showed that considering different headways allows to accommodate more trains in the network. Moreover, the future implementation of the new signalling system, that is based on the Communications-Based Train Control (CBTC) (IEEE, 1999), will allow to decrease the headways between trains allowing to increase frequency in some segments and, as a result, more timetable possibilities.

In the same manner as the headway, an overall maximum dwell time has been considered for all stopping stations. In order to do a more realistic approach this value should be defined taking different aspects of the station such as the number of platforms, frequency of trains or passenger demand. The maximum dwell time is directly related to the number of arcs in the graphs of the model. Therefore, defining a more accurate maximum dwell time per station can reduce considerably the number of arcs in the graph and, as a result, improve the performance of the model.

## 8.7 Rolling stock

The Rolling Stock Scheduling Problem (RSSP) is directly related to the timetable (see Figure 2.1) and an important phase in the railway planning process. The correct utilization of the available rolling stock resources is a main priority for TOCs. Therefore, the timetable planning process should take into consideration the rolling stock to some extent.

For example, DSB runs three main types of rolling stock in the network considered. The best type can be used to run any of the lines, however, the worst one cannot run the fastest lines (e.g. InterCity lines). In this study, the calculated routing of the passengers can help to make a better estimation of the occupancy of the trains and, as a result, assign more or fewer coaches to the trains. A train composed of fewer coaches is able to run faster than one of the same type of rolling stock with more coaches. Therefore, a better estimate of running times between stations can be done and may lead to better timetables.

### 8.7.1 Turnaround time

One of the main timetable aspects that affect the rolling stock utilization is the turnaround times at the end-of-line stations. Restrictions in turnaround times have not been considered in the lines. From DSB planning perspective, it is preferable that trains ending a path line can turnaround and be used for running the same line in the other direction within the minimum time required. For instance, a minimum turnaround time of approx. 30 minutes is considered in KH as the trains need to reach Østerport station, turnaround and travel back. At other stations, this minimum turnaround time may be lower (e.g. 10 minutes). However, at less congested end-of-line stations the turnaround times become more relevant in the sense that more rolling stock may be needed to cover the same paths. For example, a timetable that considers a lower turnaround time than the minimum one required, would require an extra train circulating to cover both paths. As a result, not considering turnaround times in the network can incur in timetables with a poor utilization of rolling stock. Nevertheless, the model should be easily extended to accommodate this requirement by just adding a restriction in the arriving and departure times of the trains at the end-of line stations.

CHAPTER 9

# Conclusion

In this thesis the optimization of railway timetables has been studied from a passenger perspective. A model has been implemented to solve the network for Regional and InterCity trains in Zealand.

The model is based on a graph formulation that takes advantage of the symmetric timetabling strategy and the assumed fixed train running times between stations. As a result, all the required train paths for a line in a cycle time of one hour can be computed by a single *shortest path*. Furthermore, the algorithm relies mainly in both Column Generation and Constraint Separation techniques. This, combined with heuristic methods to achieve an integer solution, can be transformed in an iterative process that seeks to find multiple solutions by improving the transfers from one solution to the other.

The model has been shown to find good solutions to the network in a relatively fast time. Moreover, the model is able to find the potential best transfers used by the passengers and prioritize them to find new solutions. However there are different aspects of the model that can improved.

First, a more efficient way of updating the graphs with the dual values of the RMP constraints can be found and reduce considerably the time needed by the algorithm to find a solution.

Second, alternative methods including the transfer times as part of the main mathematical model have shown promising results that can lead to more complete model that finds better solutions.

Third, the destroy method has not shown any significant relation with the improvement of the solutions and should be reconsidered. An option could be to use it as an input tool to indicate specific transfer preferences that the solutions should contain.

Fourth, although the fixed running times between stations simulate realistic cases to a large extent, allowing the trains to run at different speeds increase the flexibility of the possible solutions.

Fifth, routing the trains at a more detailed level at some stations can allow to have completely conflict-free solutions in the network. Currently, feasibility issues may arise from the model due to track-crossing conflicts at some stations where corridors join.

Sixth, the model seems to predict the PTT accurately, nevertheless, a more realistic routing of the passengers in the most congested areas can help to have a complete perspective of the trips of the passengers and the occupancy of the trains. This can be furthe rimproved if a more accurate estimation of the passenger demand is done as well.

Seventh, an individualized study of the track segments and stations can help to define more accurate parameter values for headways and dwell and transfer times and allow to have a more efficient timetable.

Eighth, it has been shown that not taking into account the turnaround times can lead to a poor usage of the rolling stock.

Last but not least, from the timetable planning process perspective of a TOC like DSB, this model can be seen as a helpful tool to include in the planning process. The model can be used to generate solutions for a wide range of scenarios where most of the parameters can be modified: from the headways and dwell times in the network to the frequency, rush hour direction and stopping patterns of the lines. It allows to test and verify any gut feeling during the planning process relatively fast. Furthermore, the graph formulation of the model allows to further implement additional real-life constraints. In addition, a good estimation of the passenger travel time can also be extracted. On the other hand, the distribution of passenger demand in congested areas seems to be estimated in an irregular way whereas in the less congested parts of the network, the train occupancy calculations seem more accurate.

# Bibliography

Alexandersson, G. & Hulten, S. (2008). The Swedish railway deregulation path. *Review of Network Economics*, *7*(1), 18–36.

Bradshaw, G. (1850). George Bradshaw's guide.

Brannlund, U., Lindberg, P., Nou, A., & Nilsson, J.-E. (1998). Railway timetabling using Lagrangian relaxation. *Transportation Science*, *32*(4), 358–369. doi:10.1287/trsc.32.4.358

Bussieck, M. R., Winter, T., & Zimmermann, U. T. (1997). Discrete optimization in public rail transport. *Mathematical Programming*, *79*(1), 415–444. doi:10.1007/BF02614327

Cacchiani, V., Caprara, A., & Toth, P. (2008). A column generation approach to train timetabling on a corridor. *4OR*, *6*(2), 125–142. doi:10.1007/s10288-007-0037-5

Cacchiani, V., Caprara, A., & Toth, P. (2010a). Non-cyclic train timetabling and comparability graphs. *Operations Research Letters*, *38*(3), 179–184. doi:https://doi.org/10.1016/j.orl.2010.01.007

Cacchiani, V., Caprara, A., & Toth, P. (2010b). Scheduling extra freight trains on railway networks. *Transportation Research Part B: Methodological*, *44*(2), 215–231. doi:https://doi.org/10.1016/j.trb.2009.07.007

Cacchiani, V., Caprara, A., & Toth, P. (2013). Finding cliques of maximum weight on a generalization of permutation graphs. *Optimization Letters*, *7*(2), 289–296. doi:10.1007/s11590-011-0416-x

Cacchiani, V. & Toth, P. (2012). Nominal and robust train timetabling problems. *European Journal of Operational Research*, *219*(3), 727–737. doi:10.1016/j.ejor.2011.11.003

Cai, X. & Goh, C. (1994). A fast heuristic for the train scheduling problem. *Computers & Operations Research*, *21*(5), 499–510. doi:https://doi.org/10.1016/0305-0548(94)90099-X

Caprara, A., Fischetti, M., & Toth, P. (2002). Modeling and solving the train timetabling problem. *Operations Research*, *50*(5), 851–861. doi:10.1287/opre.50.5.851.362

Caprara, A., Kroon, L., Monaci, M., Peeters, M., & Toth, P. (2007). Chapter 3 passenger railway optimization. In C. Barnhart & G. Laporte (Editors), *Transportation* (Volume 14, pages 129–187). Handbooks in Operations Research and Management Science. Elsevier. doi:https://doi.org/10.1016/S0927-0507(06)14003-7

Caprara, A., Monaci, M., Paolo Toth, P., & Guida, P. L. (2006). A Lagrangian heuristic algorithm for a real-world train timetabling problem. *Discrete Applied Mathematics*, *154*(5), 738–753. IV ALIO/EURO Workshop on Applied Combinatorial Optimization. doi:https://doi.org/10.1016/j.dam.2005.05.026

Carey, M. (1994). Extending a train pathing model from one-way to two-way track. *Transportation Research, Part B (methodological)*, *28B*(5), 395–400, 395–400. doi:10.1016/0191-2615(94)90038-8

Carey, M. & Lockwood, D. (1995). A model, algorithms and strategy for train pathing. *Journal of the Operational Research Society*, *46*(8), 988–1005. doi:10.1057/jors.1995.136

Cordeau, J., Toth, P., & Vigo, D. (1998). A survey of optimization models for train routing and scheduling. *Transportation Science*, *32*(4), 380–404. doi:10.1287/trsc.32.4.380

Cormen, T. H. e. a. (2009). *Introduction to Algorithms*. MIT Press,

Dantzig, G. B. & Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research*, *8*(1), 101–111. doi:10.1287/opre.8.1.101. eprint: https://doi.org/10.1287/opre.8.1.101

Desrosiers, J., Dumas, Y., Solomon, M., & Soumis, F. (1995). Chapter 2 time constrained routing and scheduling. *Handbooks in Operations Research and Management Science*, *8*, 35–139. doi:10.1016/S0927-0507(05)80106-9

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, *1*(1), 269–271. doi:10.1007/BF01386390

Dobbin, J. (1888). The opening of the Stockton and Darlington Railway, 1825. Watercolour painting. Darlington Railway Museum.

DSB. (2018a). DSB's history. https://www.dsb.dk/om-dsb/skole--og-studiemateriale/milepale-i-dsbs-historie/milepale-i-dsbs-historie/. (Accessed on 11/06/2018).

DSB. (2018b). Langsigtet Planlægning.

DSB. (2018c). Tog i Danmark.

European Commission. (1991). Directive no 91/440/EEC. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A31991L0440.

European Commission. (2013). Proposal for a regulation amending regulation (EC) no 1370/2007 as regards the opening of the market for domestic passenger transport services by rail bibtex. http://ec.europa.eu/transparency/regdoc/rep/1/2013/EN/1-2013-28-EN-F1-1.Pdf. (Accessed on 09/03/2018).

Feo, T. A. & Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, *6*(2), 109–133. doi:10.1007/BF01096763

Hansen, I. A. (2009). Railway network timetabling and dynamic traffic management. In *2nd international conference on recent advances in railway engineering, icrare-2009, tehran, ir iran, sept. 27-28, 2009*.

Harrod, S. (2012). A tutorial on fundamental model structures for railway timetable optimization. *Surveys in Operations Research and Management Science*, *17*(2), 85–96. doi:10.1016/j.sorms.2012.08.002

Higgins, A., Kozan, E., & Ferreira, L. (1997). Heuristic techniques for single line train scheduling. *Journal of Heuristics*, *3*(1), 43–62. doi:10.1023/A:1009672832658

IBM. (2011). Rail liberalisation index 2011 - market opening: Comparison of the rail markets of the member states of the European Union, Switzerland and Norway.

IEEE. (1999). Ieee standard for communications-based train control (cbtc) performance and functional requirements. *IEEE Std 1474.1-1999*, i–. doi:10.1109/IEEESTD.1999.90611

Ingvardson, J. B., Nielsen, O. A., Raveau, S., & Nielsen, B. F. (2018). Passenger arrival and waiting time distributions dependent on train service frequency and station characteristics: A smart card data analysis. *Transportation Research. Part C: Emerging Technologies*, *90*, 292–306. doi:10.1016/j.trc.2018.03.006

Jovanovic, D. & Harker, P. T. (1991). Tactical scheduling of rail operations - the SCAN-I system. *Transportation Science*, *25*(1), 46–64. doi:10.1287/trsc.25.1.46

Liebchen, C. (2004). Symmetry for periodic railway timetables. *Electronic Notes in Theoretical Computer Science*, *92*, 34–51. doi:10.1016/j.entcs.2003.12.021

Liebchen, C. (2007). Periodic timetable optimization in public transport. In K.-H. Waldmann & U. M. Stocker (Editors), *Operations research proceedings 2006* (pages 29–36). Berlin, Heidelberg: Springer Berlin Heidelberg.

Liebchen, C. (2008). The first optimized railway timetable in practice. *Transportation Science*, *42*(4), 420–435. doi:10.1287/trsc.1080.0240

Liu, P. & Han, B. (2017). Optimizing the train timetable with consideration of different kinds of headway time. *Journal of Algorithms & Computational Technology*, *11*(2), 148–162. doi:10.1177/1748301816689685. eprint: https://doi.org/10.1177/1748301816689685

Lusby, R. M., Larsen, J., Ehrgott, M., & Ryan, D. (2011). Railway track allocation: Models and methods. *Or Spectrum - Quantitative Approaches in Management*, *33*(4), 843–883. doi:10.1007/s00291-009-0189-0

Min, Y. H., Park, M. J., Hong, S. P., & Hong, S. H. (2011). An appraisal of a column-generation-based algorithm for centralized train-conflict resolution on a metropolitan railway network. *Transportation Research Part B: Methodological*, *45*(2), 409–429. doi:https://doi.org/10.1016/j.trb.2010.08.001

Odijk, M. A. (1996). A constraint generation algorithm for the construction of periodic railway timetables. *Transportation Research Part B-methodological*, *30*(6), 455–464. doi:10.1016/0191-2615(96)00005-7

Peeters, L. (2003). *Cyclic railway timetable optimization = optimalisatie van cyclische spoorwegdienstregelingen* (PhD thesis, ETH Zurich). Diss. Univ. Rotterdam, 2003.

RailNetEurope. (2018). RNE process handbook for international path allocation for infrastructure managers. http://www.rne.eu/rneinhalt/uploads/1.0_HANDB_IntlPathAllocation_20180518.pdf.

Robenek, T., Chen, T., & Bierlaire, M. (2014). The ideal train timetabling problem.

Røpke, S. & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, *40*(4), 455–472. doi:10.1287/trsc.1050.0135. eprint: https://doi.org/10.1287/trsc.1050.0135

Sadykov, R., Vanderbeck, F., Pessoa, A., Tahiri, I., & Uchoa, E. (2018). Primal Heuristics for Branch-and-Price: the assets of diving methods. *INFORMS Journal on Computing*. **retrieved from** https://hal.inria.fr/hal-01237204

Schittenhelm, B. H. (2013). *Quantitative methods for assessment of railway timetables.*

Serafini, P. & Ukovich, W. (1989). A mathematical model for periodic scheduling problems. *SIAM J. Discret. Math.* *2*(4), 550–581. doi:10.1137/0402049

Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. *Lecture Notes in Computer Science*, *1520*, 417–431.

StatBank. (2018). Population 1. january by islands - StatBank Denmark - data and statistics. http://statistikbanken.dk/statbank5a. (Accessed on 07/01/2018).

Stevenson, A. (2010). *Oxford dictionary of English.* **retrieved from** http://www.oxfordreference.com/view/10.1093/acref/9780199571123.001.0001/acref-9780199571123

Szpigel, B. (1973). Optimal train scheduling on a single track railway. *Proceedings of the Ifors Conference Operational Research '72*, 343–52, 343–352.

Transportministeriet. (2013). Togfonden DK - en hurtigere og mere miljøvenlig jernbane i Danmark. https://www.trm.dk/da/publikationer/2013/togfonden-dk---fremtidens-jernbane. (Accessed on 05/28/2018).

Transportministeriet & DSB. (2015). Kontrakt mellem transportministeriet og dsb om trafik udført som offentlig service i perioden 2015-2024. https://www.trm.dk/da/publikationer/2015/dsb-trafikkontrakt.

APPENDIX A

# Parameter tuning results

All the results of the parameter tuning are summarized in Tables A.1-A.6.

**Instance 1** — Independet transfer lines

| Sequential Dive | | | | | Random Dive | | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 32,85 | 32,78 | 32,95 | 32,96 | 5,5 | 32,90 | 32,80 | 32,80 | 32,91 |
| 7 | 32,93 | 32,86 | 32,87 | 33,02 | 7 | 32,76 | 32,81 | 32,93 | 32,90 |
| 10 | 32,89 | 32,95 | 32,80 | 33,04 | 10 | 32,84 | 32,91 | 32,93 | 32,84 |
| 15 | 32,86 | 32,80 | 32,97 | 32,92 | 15 | 32,86 | 32,81 | 32,77 | 32,94 |

**Instance 1** — All transfer lines

| Sequential Dive | | | | | Random Dive | | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 32,99 | 32,86 | 32,93 | 32,97 | 5,5 | 32,88 | 32,93 | 32,96 | 32,84 |
| 7 | 32,97 | 32,76 | 32,86 | 32,85 | 7 | 32,92 | 32,77 | 32,98 | 32,82 |
| 10 | 32,86 | 32,98 | 32,85 | 32,86 | 10 | 32,92 | 32,92 | 33,00 | 33,08 |
| 15 | 32,89 | 32,87 | 32,89 | 33,10 | 15 | 32,84 | 32,95 | 32,88 | 32,88 |

**Instance 2** — Independet transfer lines

| Sequential Dive | | | | | Random Dive | | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 32,76 | 32,92 | 33,01 | 32,93 | 5,5 | 32,74 | 33,06 | 32,93 | 33,01 |
| 7 | 33,19 | 32,91 | 32,87 | 32,84 | 7 | 33,12 | 32,78 | 32,99 | 32,92 |
| 10 | 32,96 | 32,75 | 32,95 | 33,04 | 10 | 32,92 | 32,88 | 33,03 | 32,79 |
| 15 | 32,93 | 32,91 | 32,87 | 32,94 | 15 | 32,91 | 32,92 | 32,94 | 32,92 |

**Instance 2** — All transfer lines

| Sequential Dive | | | | | Random Dive | | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 32,90 | 32,97 | 32,91 | 32,93 | 5,5 | 32,82 | 33,10 | 32,89 | 32,84 |
| 7 | 32,93 | 33,09 | 32,86 | 32,91 | 7 | 33,01 | 32,92 | 33,00 | 32,89 |
| 10 | 32,85 | 33,00 | 32,93 | 32,93 | 10 | 32,78 | 32,98 | 33,01 | 32,99 |
| 15 | 33,05 | 32,90 | 33,07 | 33,07 | 15 | 32,85 | 32,91 | 33,00 | 33,11 |

**Instance 3** — Independet transfer lines

| Sequential Dive | | | | | Random Dive | | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 32,83 | 32,88 | 32,72 | 33,08 | 5,5 | 32,90 | 32,90 | 32,92 | 33,08 |
| 7 | 33,04 | 32,83 | 32,98 | 32,92 | 7 | 33,14 | 32,73 | 33,05 | 32,91 |
| 10 | 32,86 | 32,83 | 32,89 | 32,84 | 10 | 32,82 | 32,82 | 32,93 | 32,97 |
| 15 | 33,14 | 32,84 | 32,88 | 32,89 | 15 | 33,01 | 32,87 | 32,93 | 33,17 |

**Instance 3** — All transfer lines

| Sequential Dive | | | | | Random Dive | | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 32,92 | 32,82 | 33,07 | 32,72 | 5,5 | 32,79 | 33,07 | 32,74 | 32,91 |
| 7 | 32,77 | 32,84 | 32,72 | 32,88 | 7 | 32,93 | 32,94 | 32,80 | 33,07 |
| 10 | 33,25 | 32,96 | 32,89 | 32,92 | 10 | 32,84 | 32,79 | 32,88 | 32,79 |
| 15 | 32,95 | 32,89 | 32,98 | 33,27 | 15 | 33,02 | 32,93 | 32,78 | 33,09 |

Figure A.1: Average passenger travel time of the best solution from each parameter setting algorithm runs measured in total travel minutes per hour

**Instance 1 — Independet transfer lines**

| | Sequential Dive | | | | | Random Dive | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 2002,5 | 2004,5 | 2002,5 | 2017 | 5,5 | 2002 | 2002,5 | 2027 | 2016,5 |
| 7 | 2014 | 2001,5 | 2017 | 2014 | 7 | 2010 | 2001,5 | 2007 | 2002 |
| 10 | 2001,5 | 2003 | 2013,5 | 2001,5 | 10 | 2004 | 2005,5 | 2011 | 2024 |
| 15 | 2001,5 | 2006,5 | 2024,5 | 2001,5 | 15 | 2002,5 | 2005,5 | 2011,5 | 2009,5 |

**Instance 1 — All transfer lines**

| | Sequential Dive | | | | | Random Dive | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 2009 | 2006,5 | 2012 | 2007,5 | 5,5 | 2004 | 2002 | 2001,5 | 2004 |
| 7 | 2005,5 | 2011,5 | 2001,5 | 2009 | 7 | 2016 | 2003 | 2019,5 | 2006,5 |
| 10 | 2001,5 | 2004,5 | 2005 | 2001,5 | 10 | 2004 | 2016 | 2035 | 2005 |
| 15 | 2012,5 | 2004,5 | 2021 | 2009,5 | 15 | 2009 | 2010,5 | 2009,5 | 2013 |

**Instance 2 — Independet transfer lines**

| | Sequential Dive | | | | | Random Dive | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 2029 | 2012,5 | 2009 | 2018 | 5,5 | 2004 | 2006 | 2004,5 | 2012,5 |
| 7 | 2028,5 | 2004,5 | 2021 | 2025 | 7 | 2017,5 | 2006,5 | 2011 | 2003,5 |
| 10 | 2012,5 | 2002,5 | 2004 | 2040 | 10 | 2003,5 | 2008,5 | 2011 | 2026 |
| 15 | 2007 | 2004,5 | 2016 | 2028,5 | 15 | 2005 | 2009,5 | 2022 | 2005 |

**Instance 2 — All transfer lines**

| | Sequential Dive | | | | | Random Dive | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 2014,5 | 2000,5 | 2004,5 | 2004,5 | 5,5 | 2008,5 | 2017 | 2013 | 2015,5 |
| 7 | 2004,5 | 2028,5 | 2006 | 2004,5 | 7 | 2006,5 | 2026,5 | 2001 | 2002,5 |
| 10 | 2043,5 | 2006,5 | 2007 | 2007 | 10 | 2009,5 | 2004,5 | 2017 | 2006 |
| 15 | 2014,5 | 2007 | 2013,5 | 2013,5 | 15 | 2019,5 | 2006,5 | 2007 | 2008,5 |

**Instance 3 — Independet transfer lines**

| | Sequential Dive | | | | | Random Dive | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 2006,5 | 2001,5 | 2001 | 2027 | 5,5 | 2008 | 2013 | 2001,5 | 2014,5 |
| 7 | 2003,5 | 2006,5 | 2027 | 2001,5 | 7 | 2020 | 2002 | 2007,5 | 2002,5 |
| 10 | 2001 | 2006,5 | 2004,5 | 2011,5 | 10 | 2001 | 2001 | 2006 | 2009,5 |
| 15 | 2009 | 2002 | 2001 | 2001,5 | 15 | 2009 | 2021,5 | 2002,5 | 2028,5 |

**Instance 3 — All transfer lines**

| | Sequential Dive | | | | | Random Dive | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 2001,5 | 2006,5 | 2007 | 2001 | 5,5 | 2001 | 2018,5 | 2001 | 2001 |
| 7 | 2006,5 | 2006,5 | 2001 | 2001 | 7 | 2021,5 | 2019,5 | 2001 | 2002,5 |
| 10 | 2032 | 2006,5 | 2004,5 | 2001,5 | 10 | 2012 | 2001 | 2009 | 2003 |
| 15 | 2017 | 2001 | 2002,5 | 2034,5 | 15 | 2024,5 | 2010 | 2006,5 | 2031,5 |

Figure A.2: Average sum of path lengths of the best solution from each parameter setting algorithm runs measured in minutes

**Instance 1** — Independet transfer lines

| Sequential Dive | | | | | Random Dive | | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 17 | 14 | 13 | 10 | 5,5 | 20 | 21,5 | 15 | 19 |
| 7 | 12,5 | 14 | 13 | 18 | 7 | 15,5 | 21 | 16 | 12,5 |
| 10 | 17 | 9,5 | 13 | 9,5 | 10 | 20 | 14,5 | 17 | 19 |
| 15 | 15 | 16,5 | 15 | 9 | 15 | 18 | 15 | 12 | 7 |

**Instance 1** — All transfer lines

| Sequential Dive | | | | | Random Dive | | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 8 | 21 | 12 | 16 | 5,5 | 18 | 11,5 | 12 | 20 |
| 7 | 7 | 15 | 21 | 17 | 7 | 19 | 14,5 | 6,5 | 20,5 |
| 10 | 17 | 17 | 14 | 22 | 10 | 14,5 | 17 | 11 | 17,5 |
| 15 | 13 | 12,5 | 13 | 14 | 15 | 17 | 18 | 9,5 | 21 |

**Instance 2** — Independet transfer lines

| Sequential Dive | | | | | Random Dive | | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 6 | 6 | 6 | 7 | 5,5 | 8 | 7 | 9 | 7,5 |
| 7 | 4,5 | 8 | 6 | 6,5 | 7 | 5,5 | 11 | 8 | 8 |
| 10 | 6,5 | 6 | 6 | 8,5 | 10 | 8 | 6 | 6 | 5,5 |
| 15 | 5 | 7 | 7 | 4 | 15 | 13 | 10 | 6 | 4 |

**Instance 2** — All transfer lines

| Sequential Dive | | | | | Random Dive | | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 5 | 9 | 6 | 7 | 5,5 | 8 | 6 | 6 | 12 |
| 7 | 7 | 5 | 5 | 5 | 7 | 3,5 | 7 | 10 | 7 |
| 10 | 6 | 6 | 6 | 5 | 10 | 11 | 7 | 4 | 9 |
| 15 | 6 | 7,5 | 4 | 6 | 15 | 9 | 7,5 | 9,5 | 3 |

**Instance 3** — Independet transfer lines

| Sequential Dive | | | | | Random Dive | | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 5 | 2,5 | 5 | 3 | 5,5 | 4 | 2,5 | 1,5 | 5 |
| 7 | 5 | 4 | 4 | 5 | 7 | 5 | 5 | 3 | 5 |
| 10 | 4,5 | 4 | 2,5 | 3,5 | 10 | 8 | 8 | 2 | 3 |
| 15 | 2 | 5 | 3 | 3 | 15 | 4 | 8 | 5 | 3 |

**Instance 3** — All transfer lines

| Sequential Dive | | | | | Random Dive | | | | |
|---|---|---|---|---|---|---|---|---|---|
| minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 |
| 5,5 | 3 | 4 | 4 | 6,5 | 5,5 | 5 | 2,5 | 10 | 10,5 |
| 7 | 4,5 | 3,5 | 7 | 2 | 7 | 5 | 4,5 | 6 | 3 |
| 10 | 4,5 | 3 | 4,5 | 5 | 10 | 5,5 | 8 | 4 | 6,5 |
| 15 | 4 | 4 | 3 | 3,5 | 15 | 4 | 5 | 5 | 5 |

Figure A.3: Average algorithm iterations of each parameter setting algorithm runs

| | Sequential Dive | | | | Random Dive | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Instance 1** | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | Independet transfer lines |
| | 5,5 | 53% | 71% | 54% | 83% | 5,5 | 60% | 55% | 47% | 37% | |
| | 7 | 50% | 64% | 85% | 61% | 7 | 56% | 52% | 56% | 46% | |
| | 10 | 60% | 56% | 62% | 60% | 10 | 55% | 58% | 59% | 53% | |
| | 15 | 53% | 59% | 47% | 78% | 15 | 50% | 59% | 42% | 57% | |
| | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | All transfer lines |
| | 5,5 | 45% | 67% | 50% | 81% | 5,5 | 37% | 27% | 58% | 58% | |
| | 7 | 71% | 71% | 62% | 59% | 7 | 63% | 53% | 48% | 53% | |
| | 10 | 59% | 56% | 50% | 64% | 10 | 53% | 60% | 34% | 61% | |
| | 15 | 62% | 58% | 54% | 71% | 15 | 53% | 63% | 56% | 78% | |
| **Instance 2** | Case 1-1 | | | | | Case 2-1 | | | | | |
| | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | Independet transfer lines |
| | 5,5 | 83% | 58% | 50% | 71% | 5,5 | 63% | 71% | 67% | 76% | |
| | 7 | 40% | 63% | 67% | 57% | 7 | 33% | 55% | 63% | 75% | |
| | 10 | 67% | 83% | 50% | 88% | 10 | 75% | 33% | 17% | 100% | |
| | 15 | 60% | 71% | 71% | 75% | 15 | 54% | 70% | 83% | 50% | |
| | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | All transfer lines |
| | 5,5 | 80% | 56% | 50% | 57% | 5,5 | 63% | 17% | 83% | 58% | |
| | 7 | 86% | 40% | 60% | 60% | 7 | 33% | 71% | 80% | 57% | |
| | 10 | 83% | 50% | 67% | 80% | 10 | 45% | 43% | 75% | 56% | |
| | 15 | 50% | 63% | 50% | 33% | 15 | 44% | 49% | 60% | 50% | |
| **Instance 3** | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | Independet transfer lines |
| | 5,5 | 60% | 50% | 20% | 33% | 5,5 | 75% | 100% | 50% | 40% | |
| | 7 | 60% | 75% | 100% | 40% | 7 | 20% | 40% | 33% | 40% | |
| | 10 | 100% | 75% | 100% | 67% | 10 | 38% | 38% | 50% | 67% | |
| | 15 | 50% | 80% | 67% | 33% | 15 | 25% | 38% | 40% | 33% | |
| | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | All transfer lines |
| | 5,5 | 33% | 50% | 75% | 83% | 5,5 | 40% | 50% | 70% | 27% | |
| | 7 | 50% | 67% | 57% | 50% | 7 | 40% | 25% | 17% | 50% | |
| | 10 | 25% | 67% | 50% | 60% | 10 | 80% | 50% | 50% | 71% | |
| | 15 | 75% | 50% | 67% | 25% | 15 | 75% | 20% | 80% | 60% | |

Figure A.4: Average feasible solutions from the algorithm runs for each parameter setting measured as a percentage of the total amount of algorithm iterations

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance 1 | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | Independet transfer lines |
| | 5,5 | 100% | 100% | 0% | 50% | 5,5 | 50% | 100% | 100% | 50% | |
| | 7 | 100% | 50% | 100% | 0% | 7 | 0% | 100% | 0% | 0% | |
| | 10 | 0% | 50% | 0% | 0% | 10 | 50% | 0% | 0% | 100% | |
| | 15 | 0% | 100% | 0% | 50% | 15 | 0% | 100% | 0% | 0% | |
| | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | All transfer lines |
| | 5,5 | 0% | 100% | 50% | 100% | 5,5 | 50% | 0% | 0% | 50% | |
| | 7 | 0% | 100% | 0% | 100% | 7 | 100% | 100% | 0% | 0% | |
| | 10 | 50% | 50% | 100% | 0% | 10 | 0% | 0% | 50% | 100% | |
| | 15 | 0% | 100% | 0% | 0% | 15 | 100% | 0% | 0% | 0% | |
| Instance 2 | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | Independet transfer lines |
| | 5,5 | 100% | 50% | 0% | 0% | 5,5 | 0% | 100% | 0% | 50% | |
| | 7 | 0% | 0% | 100% | 100% | 7 | 50% | 0% | 0% | 50% | |
| | 10 | 0% | 100% | 100% | 100% | 10 | 100% | 0% | 0% | 0% | |
| | 15 | 0% | 50% | 50% | 0% | 15 | 100% | 50% | 100% | 0% | |
| | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | All transfer lines |
| | 5,5 | 50% | 50% | 0% | 0% | 5,5 | 0% | 0% | 100% | 100% | |
| | 7 | 0% | 0% | 100% | 0% | 7 | 0% | 100% | 0% | 0% | |
| | 10 | 100% | 100% | 0% | 0% | 10 | 0% | 0% | 0% | 100% | |
| | 15 | 0% | 50% | 0% | 0% | 15 | 50% | 50% | 100% | 0% | |
| Instance 3 | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | Independet transfer lines |
| | 5,5 | 0% | 0% | 0% | 0% | 5,5 | 0% | 100% | 0% | 0% | |
| | 7 | 50% | 0% | 100% | 0% | 7 | 0% | 0% | 0% | 0% | |
| | 10 | 0% | 0% | 0% | 100% | 10 | 0% | 0% | 0% | 0% | |
| | 15 | 50% | 100% | 0% | 0% | 15 | 0% | 100% | 0% | 0% | |
| | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | showq | minTT\NT | 1 | 2 | 5 | 10 | All transfer lines |
| | 5,5 | 0% | 0% | 0% | 0% | 5,5 | 0% | 0% | 0% | 100% | |
| | 7 | 0% | 0% | 0% | 0% | 7 | 0% | 0% | 0% | 0% | |
| | 10 | 0% | 100% | 0% | 0% | 10 | 100% | 0% | 0% | 0% | |
| | 15 | 100% | 50% | 100% | 0% | 15 | 100% | 50% | 0% | 100% | |

Figure A.5: Average number of times the best solution was found after applying the destroy method for each parameter setting algorithm runs measured as a percentage of the total amount of runs

| | Sequential Dive | | | | | Random Dive | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance 1 | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | Independet transfer lines |
| | 5,5 | 24% | 29% | 8% | 15% | 5,5 | 20% | 14% | 13% | 11% | |
| | 7 | 16% | 32% | 23% | 22% | 7 | 6% | 14% | 6% | 0% | |
| | 10 | 9% | 16% | 8% | 11% | 10 | 10% | 28% | 6% | 21% | |
| | 15 | 13% | 15% | 0% | 22% | 15 | 11% | 27% | 8% | 0% | |
| | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | All transfer lines |
| | 5,5 | 0% | 29% | 8% | 38% | 5,5 | 11% | 9% | 8% | 20% | |
| | 7 | 29% | 13% | 14% | 18% | 7 | 16% | 14% | 0% | 5% | |
| | 10 | 29% | 9% | 14% | 18% | 10 | 14% | 24% | 9% | 17% | |
| | 15 | 0% | 16% | 0% | 14% | 15 | 12% | 11% | 0% | 24% | |
| Instance 2 | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | Independet transfer lines |
| | 5,5 | 17% | 33% | 0% | 0% | 5,5 | 25% | 29% | 22% | 13% | |
| | 7 | 33% | 25% | 33% | 31% | 7 | 27% | 27% | 25% | 6% | |
| | 10 | 15% | 33% | 17% | 24% | 10 | 13% | 0% | 0% | 0% | |
| | 15 | 20% | 14% | 14% | 0% | 15 | 8% | 20% | 17% | 0% | |
| | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | All transfer lines |
| | 5,5 | 20% | 6% | 0% | 0% | 5,5 | 13% | 8% | 17% | 17% | |
| | 7 | 0% | 20% | 20% | 20% | 7 | 0% | 14% | 20% | 0% | |
| | 10 | 17% | 17% | 0% | 0% | 10 | 0% | 14% | 25% | 44% | |
| | 15 | 0% | 13% | 38% | 0% | 15 | 6% | 13% | 32% | 33% | |
| Instance 3 | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | 10 | minTT\NT | 1 | 2 | 5 | 10 | Independet transfer lines |
| | 5,5 | 10% | 0% | 0% | 0% | 5,5 | 13% | 40% | 0% | 0% | |
| | 7 | 20% | 0% | 25% | 0% | 7 | 0% | 0% | 0% | 0% | |
| | 10 | 0% | 13% | 20% | 29% | 10 | 0% | 0% | 0% | 0% | |
| | 15 | 25% | 20% | 0% | 0% | 15 | 0% | 13% | 0% | 0% | |
| | Sequential Dive | | | | | Random Dive | | | | | |
| | minTT\NT | 1 | 2 | 5 | showq | minTT\NT | 1 | 2 | 5 | 10 | All transfer lines |
| | 5,5 | 0% | 13% | 0% | 0% | 5,5 | 0% | 0% | 5% | 24% | |
| | 7 | 0% | 0% | 7% | 0% | 7 | 10% | 0% | 8% | 17% | |
| | 10 | 0% | 17% | 11% | 10% | 10 | 18% | 6% | 0% | 8% | |
| | 15 | 25% | 13% | 33% | 14% | 15 | 25% | 10% | 20% | 20% | |

Figure A.6: Average number of times the destroy method improved the previous solution for each parameter setting algorithm runs measured as a percentage of the total amount of algorithm iterations

# APPENDIX B
# Revised project plan

In this Appendix, the initially planned objectives are revised and the actual level of achievement is commented. Furthermore, a brief self-evaluation of the overall project plan is commented.

## B.1 Objectives

**Objective:** Describe the current Regional and InterCity railway network and timetabling system in Zealand.

This objective has been achieved. All the network has been carefully defined, distinguishing the parts of it that were covered by the study. Moreover, all the timetabling process both from the infrastructure manager and operator perspective have been described.

**Objective:** Conduct an extensive literature review of the state-of-the-art in Train Timetabling methods.

This objective has been achieved. Literature covering the main methods use to solve both non-cyclic and cyclic timetables have been described and classified according to different criteria.

**Objective:** Formulate and implement successfully a realistic model to find timetables to the defined system.

This objective has been achieved. A model that is able to find timetables to a given network has been successfully implemented. Indeed, the model is able to find solutions relatively fast and has been extended to improve the timetables from the passenger perspective.

**Objective:** Evaluate optimization results with respect to the current situation and identify the strengths and weaknesses of the implemented model.

This objective has been achieved. The performance of the algorithm has been analyzed and different scenarios have been compared to a manually planned timetable. Moreover, an extensive discussion has been conducted about the implemented describing the capabilities and limitations of the model.

**Objective:** Suggest possible further improvements to the system and discuss their feasibility.

This objective has been achieved. Improvements and further work to compensate the limitations of the problem have been suggested as well as difficulty level of implementing them.

## B.2 Self evaluation

This thesis has been my main priority during the last six months. I found the problem interesting and challenging from the beginning. Although my lack of expertise in some programming aspects at the beginning complicated an efficient implementation, I was able to document myself and find a reasonable workaround quickly. To be honest, I slightly underestimated the writing part of the thesis. Although, making a good implementation was my first goal at a personal level, I am aware that being able to document the project in a clear way is very important as well. As a result, the last few weeks of the project plan were a bit tougher (see Figure B.1) in order complete all the requirements and targets I set to myself.

I have definitely learned a lot in the academical aspect, specially regarding graph optimization and other aspects of Operations Research such as Column Generation and separation techniques or heuristic methods.

Being able to collaborate with a real company and having access to real data allowed me to learn considerably about the professional field and work environment. I realized the immense value that experience adds to any part of a planning process. At the beginning, my background within Railway and Public transportation was not strong, but it improved considerably during this last semester.

Last but not least, this thesis has improved my self-organization level and helped me being more structured and committed to research and work.

In overall, I am very happy and satisfied with this project and I would not hesitate to keep improving it if I had more time.

| ACTIVITY | PLAN START | PLAN DURATION | ACTUAL START | ACTUAL DURATION | PERCENT COMPLETE |
|---|---|---|---|---|---|
| **Context Setting** | 1 | 3 | 1 | 5 | 100% |
| **Case Study Definition** | 1 | 6 | 1 | 10 | 100% |
| **Data Management** | 2 | 7 | 2 | 15 | 100% |
| **Literature Review** | 1 | 20 | 1 | 19 | 100% |
| **Solution Method** | 2 | 13 | 1 | 15 | 100% |
| **Software Implementation** | 2 | 16 | 1 | 20 | 100% |
| **Computational Results** | 12 | 8 | 17 | 7 | 100% |
| **Analysis of Results** | 16 | 6 | 21 | 5 | 100% |
| **Discussion** | 20 | 4 | 22 | 5 | 100% |
| **Conclusion and Further Studies** | 23 | 2 | 23 | 4 | 100% |
| **Revision of Initial Project Plan** | 24 | 2 | 25 | 2 | 100% |
| **Report Writing** | 1 | 26 | 1 | 26 | 100% |

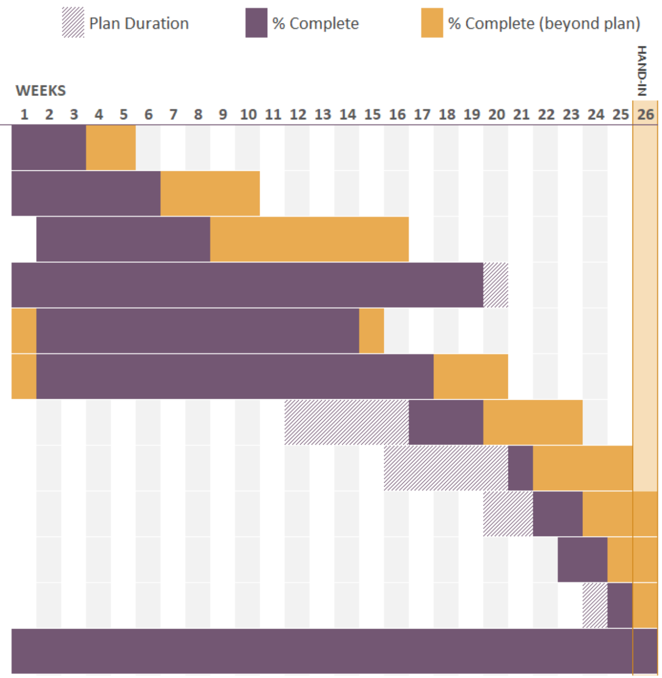


Figure B.1: Revised project time-plan

# Abbreviations

**ALNS** Adaptive Large Neighborhood Search. 102

**CBTC** Communications-Based Train Control. 106

**CG** Column Generation. i, viii, 52, 53, 55, 57, 59, 72, 90, 99, 126

**CSP** Crew Scheduling Problem. 5

**DSB** Danske Statsbaner. i, viii, xv, 2, 3, 28–31, 65, 71, 91–94, 107, 110

**DTA** Danish Transport Authority. 7, 15

**EC** European Comission. 6

**EU** European Union. 6, 15

**GRASP** Greedy Randomized Adaptive Procedures. 102

**HK** Holbæk. 26, 35, 36, 74

**HTA** Høje Taastrup. 35, 46, 74, 103, 104, 106

**IC** InterCity. 30, 31

**IFIT** Integrated Fixed-Interval Timetabling. 11, 15

**ILP** Integer Linear Problem. 18, 19, 33, 46, 53

**IM** Infrastructure Manager. 6, 7, 15, 16, 31

**KB** Kalundborg. 26, 30, 35, 51, 74

**KH** København H. ix, xi, xii, xv, 26, 30, 35, 36, 42, 43, 46, 51, 58, 64, 73, 74, 77–80, 82, 84, 94, 95, 97, 103, 104, 106, 107

**KJ** Køge. 93, 104

**KJN** Køge Nord. 77

**KPI** Key Performance Indicator. 6, 31

**LIB** Rail Liberalization index. 7

**LNS** Large Neighborhood Search. i, viii, 66, 67, 69, 102

**LP** Linear Problem. 53, 55, 61, 62, 90

**LPP** Line Planning Problem. 5

**MILP** Mixed Integer Linear Problem. 18, 19

**MIP** Mixed Integer Problem. 11, 70–72, 99

**MP** Master Problem. 55, 61, 62

**NF** Nykøbing Falster. 26, 30, 74, 77, 78, 93

**NPP** Network Planning Problem. 5

**NÆ** Næstved. 26, 79

**OD** Odense. xii, 26, 30, 64, 65, 67, 74

**OD** Origin-Destination. viii, 55, 65, 66, 106

**PESP** Periodic Event Scheduling Problem. 18, 21, 22

**PP** Pricing Problem. 55, 58, 59, 70, 91, 92, 99, 101

**PTT** Passenger Travel Time. xii, 55, 70, 73, 85–89, 94, 96, 97, 101, 102, 109

**RG** Ringsted. xii, 26, 30, 64, 74, 77–79, 81, 103

**RMP** Reduced Master Problem. viii, 53, 55, 58–62, 70, 72, 92, 99, 109

**RNE** RailNetEurope. xi, 15

**RO** Roskilde. xii, 35, 46, 64, 65, 67, 74, 79, 95, 103, 104

**RSSP** Rolling Stock Scheduling Problem. 5, 107

**SPP** Set Partitioning Problem. 46

**TOC** Train Operating Company. 7, 13–16, 18, 31, 107, 110

**TRP** Train Routing Problem. 5

**TTP** Train Timetabling Problem. i, 5, 18, 19, 21

**VO** Vordingborg. 93

**ØLB** Ølby. 93