Downloaded from orbit.dtu.dk on: May 13, 2024

DTU Library

Efficiency and Robustness in Railway Operations

Bull, Simon Henry

Publication date:
2016

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Bull, S. H. (2016). *Efficiency and Robustness in Railway Operations*. DTU Management Engineering.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Efficiency and Robustness in Railway Operations

Simon Henry Bull

Kongens Lyngby 2016

Technical University of Denmark
DTU Management Engineering
Management Science
Produktionstorvet, Building 426,
2800 Kongens Lyngby, Denmark
Phone +45 4525 4800
www.man.dtu.dk

Summary

Passenger railway transport is an effective means of providing high capacity transport that is energy efficient and has low emissions. As the population of Denmark grows and there is an increased request for mobility, there is a need for railway services offering greater capacity and more reliability. Offering these services presents a challenging sequence of planning problems for operators. These range from problems considered on a daily basis to planning for years in the future, with different problems interacting and influencing each other.

Operations research methods can be used to effectively model, investigate, and solve railway planning problems. Despite advances in computational power these large problems are still challenging to solve, especially as more modelling detail is sought. Within a Danish context this thesis seeks to apply operations research methods to different planning problems beyond past approaches, and where applicable, investigate solution methods that place more focus on the passenger and passenger experience. To cater to the growing demand for rail transport, and compete with different modes of transport, Danish railway operators must offer a consistent, reliable service, that is well planned from both a passenger and operator perspective. This thesis therefore considers different planning problems within passenger railway considering robustness of the system, and efficiency and optimality from the point of view of the passenger or operator.

The contributions of the thesis are in the investigation of robustness in railway, the application of optimization to a number of railway planning problems, and a detailed consideration of the specific concerns of Danish railway services. These contributions are summarised in the introductory chapter, and in the latter part of the thesis are given in each chapter.

Resumé (Danish summary)

Passager-transport via jernbane er en effektiv transport form der sikrer høj kapacitet, er energi effektiv og har lav udledning af drivhusgasser. Efterhånden som Danmarks befolkning vokser og der er et ønske om øget mobilitet er der brug for at jernbanen som en del af den offentlige transport tilbyder større kapacitet og pålidelighed. At udbyde disse services stiller operatørerne overfor en udfordrende række af planlægningsproblemer. Disse går fra operationelle problemer på daglig bases til mere strategiske planlægningsproblemer, hvor tidshorisonten er et år eller mere.

Operationsanalyse kan bruges til at modellere, analysere og løse mange af disse jernbane-relatede planlægningsproblemer. På trods as stigende computerkraft er en række af disse optimeringsproblemer stadig en udfordring at løse bla. fordi man søger at få stadig flere detaljer med i modelleringen. Med baggrund i problemstillilngerne som de ser ud for en dansk operatør belyser denne ph.d.-afhandling hvordan operationsanalytiske metoder kan anvendes til forskellige planlægningsproblemer i den samlede planlægningsproces. Afhandlingen afspejler også et øget fokus på passagerne og deres oplevelse af den transport-service de tilbydes. Der er specifik behandlet problemstiilinger omkring robusthed af togsystemet samt effektivitet og optimalitet ud fra et passager fokus og/eller operatør-fokus.

Denne afhandlings primære bidrag ligger i undersøgelser af robusthed i jernbanedrift, anvendelse af optimering på en række centrale planlægningsrproblemer og en detaljeret beskrivelse af centrale udfordringer for danske operatører. Bidragende er overordnet beskrevet i det introducerende kapitel og efterfølgende behandlet mere grundigt.

Preface

This thesis has been submitted to the Department of Management Engineering, Technical University of Denmark, in partial fulfilment of the requirements for acquiring the PhD degree. The work has been supervised by Professor Jesper Larsen and Professor David Pisinger.

The project has been co-funded by the Danish Council for Strategic Research as part of the interdisciplinary RobustRailS project.

The thesis consists of two parts. The first part introduces the thesis background and motivation, and introduces the problems studied. The second part is a compilation of scientific articles written during the project.

Kongens Lyngby, Denmark

May 2016

Simon Henry Bull

Acknowledgements

Firstly, I would like to thank my primary supervisor Professor Jesper Larsen for his support and guidance throughout the project. With his wide knowledge of optimization and industry problems, and contacts in the rail industry in Denmark, he has been a great source of assistance and advice. I thank my co-supervisor Professor David Pisinger for providing advice and support when needed. Associate Professor Richard Lusby has provided invaluable assistance and guidance, taking regular meetings and discussions, proof-reading work, and offering suggestions for new directions of research. Also, I thank my co-authors for their contributions to the individual research works.

I also want to thank Associate Professor Andrew Mason and Dr Andrea Raith for hosting me at the University of Auckland, and everybody in the Department of Engineering Science for making the trip a worthwhile, valuable experience. In Denmark, I am thankful to my fellow PhD students and colleagues in Management Science for always providing a fun, supportive environment. It has been a pleasure spending three years in the department.

I am very thankful to my friends and family in Denmark, New Zealand, and around the world for their support, and I especially thank those who have proof-read parts of this thesis.

Much of the work has been applied to problems encountered by the Danish rail operator DSB, and they have provided time, direction, and real world data for the project. Finally, I thank the funding of RobustRailS, provided by the Danish Council for Strategic Research, without which the project would not have been possible.

Contents

Part I

Introduction

Chapter 1

Thesis Overview

1.1 Motivation

Globally, passenger numbers are increasing, placing more demand on infrastructure, operations, and consequently on planning. In a Danish context passenger railway systems are relied upon daily by commuters and long distance travellers. Railway services offer a viable alternative to private vehicles, especially as cities become more densely populated. However with increasing passenger numbers, challenges arise in meeting demand while still providing a reliable, punctual service.

This thesis is largely focused on railway operations in Denmark, with rail operator Danske Statsbaner (DSB) and especially on the Copenhagen S-tog commuter network. In a Danish context, there is recognition that passengers are not committed to railway and will make use of alternative modes of transport if railway services are not sufficiently reliable. Quoting the 2011 annual report for DSB S-tog [DSB S-tog a/s, 2011]:

> ... a continued increase in passenger numbers is expected. Therefore, an important task will be to ensure capacity for the growth while at the same time ensuring that we can retain the customers who have already abandoned other means of transport ...

The challenge of avoiding a degradation in service while increasing capacity is highlighted, and the fact that passengers will not maintain a commitment to rail if service degrades.

Passenger numbers have indeed increased; the 2015 annual report for DSB [DSB, 2015a] tabulates annual passenger numbers (in thousands) for the years 2011–2015, reproduced in part in Table 1.1. Passenger numbers for both the S-tog network in Copenhagen and for other Danish rail use grew every year, and in total S-tog passenger numbers grew by over 10% during the five year period.

Table 1.1: Rail usage in Denmark, in thousands of annual passenger trips.

	2011	2012	2013	2014	2015
S-tog	103,393	106,133	109,242	111,967	114,121
Other	72,710	74,844	75,973	76,360	77,760
Total	176,103	180,977	185,215	188,327	191,881

To meet this challenge, planning and decision support tools are needed. Despite increases in computational capability, railway planning problems are still large and complex, and it is difficult to find good quality solutions for all problems. In many cases planning is or has been performed manually, by teams of planners with detailed knowledge and experience. However as greater capacity is sought, and the complex interactions and inter-dependencies between parts of plans and between different systems become more significant and more constraining, such manual planning becomes infeasible.

Operations research methods provide tools that can effectively assist in describing and solving these complex planning problems. However more research is needed in exploring, testing and validating new models and methods for solving such problems. Despite similarities, operations in different countries often have different details and requirements that facilitate (and require) different methods being investigated. Changing attitudes, policies and expectations for rail transport require new approaches and greater assurances.

RobustRailS (Robustness in Railway OperationS) is an interdisciplinary research project funded by the Danish Council for Strategic Research bringing together four departments of DTU to investigate improvements for railway reliability, punctuality and sustainability in Denmark. This PhD thesis is one of four RobustRailS PhD projects and aims to apply operations research and optimization methods to railway planning problems, specifically considering passenger perspectives of service quality and reliability.

1.2 Thesis structure

This thesis consists of three parts. Part I contains an introduction to the topic, background, the methods used and problems studied. Part II contains the main research topics studied. Part III contains additional research work not included in the main body.

Part I has a single bibliography chapter, while in Parts II and III, each chapter has its own bibliography section.

1.3 Thesis contributions

The main contributions of this thesis are contained in the four chapters of Part II. These are summarised here:

Chapter 6: A Survey on Robustness in Railway Operations.
A survey on the concept of robustness as used for railway planning or for assessing railway plans. The scope spans all levels of railway planning, from long term strategic planning to short term tactical level planning.

Chapter 7: An applied optimization based method for line planning to minimize travel time.
A line planning problem is defined here, with particular features for the Copenhagen S-tog problem included, and particular features for modelling passenger costs in a detailed way. The model is tested on the Copenhagen S-tog problem, showing a range of solutions, and several metrics are proposed for comparing the different solutions. Different methods are also proposed for solving the problem, and though we can not generally find optimal solutions we can for some special cases, and can find good quality solutions for the general case.

Chapter 8: Integrating robust timetabling in line plan optimization for railway systems.

The problem of line planning while considering well spread, “robust” timetabling, is described here. Taking a heuristic approach, a method is presented for modifying a line plan based on a timetabling model. The iterative approach is applied to the Copenhagen S-tog network, showing how different line plans, including two real line plans, can be modified in this method to ensure good timetables can be created for them.

Chapter 9: Exact Methods for Solving the Train Departure Matching Problem.
The Departure Matching Problem is defined, as a subproblem of the ROADEF/EURO Challenge 2014 problem [Ramond and Nicolas, 2014]. Two different methods for solving problem instances are derived and compared with the given instances problems of Ramond and Nicolas [2014]. The departure matching (sub-)problem is proven to be NP-complete, suggesting that the entire challenge problem is difficult. The sub-problem is not solvable to optimality for all challenge instances but solutions are obtainable and bounds discovered which are valid for the full challenge problem. The method for the challenge problem itself, presented in Appendix A, was awarded the second place PhD prize for the ROADEF/EURO Challenge 2014.

CHAPTER 2

Railway planning and operations

Passenger railway is a complex service that operates daily on tight time schedules. There are different stakeholders concerned with the operation, each holding different viewpoints on the objectives and outcomes observed in operations, such as the passengers, the operator, and the transport authority. The plans executed on the day of operation are a composition of different sub-problems that have been assembled in a planning process over a long period of time. Typically, a sequence of sub-problems is solved in the lead up to the day of operation, starting from problems that have a very long term outlook on passenger railway (course plans made potentially years in advance), and terminating with very detailed and exact planning for a single day. Solutions to long-term planning problems may be valid for long periods of time and only reconsidered occasionally, while shorter term planning problems may be regularly re-solved.

In this chapter we give an overview of a typical decomposition of railway planning for passenger railway services. These will be expanded upon in the sections below, but in short these consist of:

- Network design: determining the station locations and network topology.
- Line planning: determining the “lines” (routes, frequencies) for trains in the network.

- Timetabling: determining exact times for specific trains travelling through the network.
- Rolling stock planning: assigning sequences of trips to rolling stock units; planning precise movements in stations and depots.
- Crew planning: assigning crew persons to duties required by the timetable and rolling stock plans.

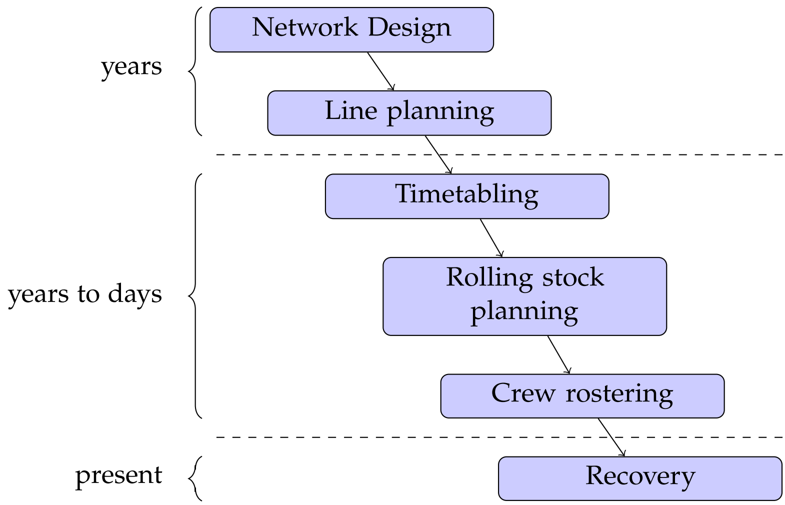


Figure 2.1: A typical sequence of rail planning problems, indicating their relative time horizons.

These problems are shown in Figure 2.1, showing the relative timings of the planning problems and some indication of the time horizon they consider. Also displayed on the figure is recovery, which also includes disruption management and other decision processes that occur during operations, but here we do not focus on these problems. The arrows are only indicated in a single direction here but in reality there are also feedback mechanisms where plans at an "earlier" stage are adapted due to considerations or difficulties encountered in a "later" stage.

We expand upon each of these in the following sections with a view toward the problems' susceptibilities to operations research and optimization methods.

2.1 Network design

At a strategic level of planning, planners consider the location and size of stations and the tracks connecting them. In the European model of public transport, network resources are owned and managed by a single entity, while the operation of trains on the network is carried out by one or more independent operators.

Network design, or network planning, is the consideration of these long term decisions about investment in railway infrastructure. It includes the building of new stations, tracks, and network systems, and also considers in greater detail aspects such as the number of platforms in a station and detailed track layouts.

With such a long term view, decisions made must remain valid for the long term. Station locations for example tend to remain fixed for decades, while the nature of the operation of trains running through them may change significantly in that time. Such decisions are therefore made considering a lot of uncertainty, and are typically made irregularly. The decisions made have economic and political implications outside the immediate scope of passenger railway transport itself, and as such, these problems have not been as studied in the literature in a similar manner to some of the following problems we describe, especially from an operations research point of view. Nevertheless, some work has been undertaken.

Magnanti and Wong [1984] give an overview of models and method used for network design in transportation. There, the authors comment that due to the time scale involved in transportation network decisions, and the role that uncertainty plays in the outcome rather than combinatorial effects, operations research models are not always applicable. As an exception, the author identifies transport *service* networks such as an airline network, where network alteration does not involve capital expenditure, such as building tracks, but rather simply deciding to offer a service between a pair of cities. An analogy may be drawn with the liner shipping network design: the problem of designing a service network for liner container ships that carry shipping containers on fixed schedules. Brouer et al. [2014] give a recent overview of the problem, reference model and example benchmark instances. Here, ports (analogous to railway stations) are taken as fixed, and implementing a connection between them simply requires sailing a ship directly between them (and a ship may generally sail between any pair of ports). Therefore, the liner shipping network design problem has much freedom for changes to the underlying network structure in terms of connections between ports, while in contrast in railway such changes would require significant expenditure. Due to that freedom, liner ship network design considers not only the existence of connections between ports but also the ship circulations (analogous to train lines) that operate in the network as part of a single problem. In railway, line planning is generally considered as a distinct problem that is elaborated upon in the following section. Quak [2003] considers the construction of a bus line plan, which would again be a combination of the distinct problems of network design and line planning. The author takes as a problem parameter the driving time between *every* pair of stations, which is possible without additional investment in infrastructure due to the existence of a road network. In the work, bus station locations are fixed, but as bus stations are relatively easy to build or reposition, their locations can also be considered as part of the network design planning problem. Silman et al. [1974] for example consider the problem of planning bus routes by dividing a city up into *zones*, and the bus network consists of connections and routes travelling between adjacent zones, without explicitly considering the presence or locations of bus stops. Again, this problem combines elements of network design in the bus stops and connections between them, with the line planning problem elaborated upon in the subsequent section.

2.2 Line planning

As alluded to in the previous section, the line planning problem is that of planning *routes* in the network. Typically, in a passenger railway context, trains operate on fixed published routes within the network at some regular schedule, with some stopping pattern and published frequency. For example a local train may operate every hour between two major cities stopping at every minor town in between, terminating at the end city, while an inter-city train may also pass between the two cities once every hour but as part of a longer route. Given the fixed network in infrastructure then, the line planning problem is that of constructing lines, that are each a route, stopping pattern, frequency, and service type.

A route is a path in the infrastructure network, which in railway networks is likely either a simple path between two end stations, or a simple cycle. A stopping pattern is the sequence of stations along the network that the line stops at. A frequency is the number of trains that operate on the line every hour, generally in both directions. A service type dictates the speed or type of train unit that operates on the line. In a typical line planning problem, some of these decisions would be fixed or implied by others. For example, if planning the lines for an inter-city train system in some country, stopping patterns would already be known (being the major cities, with other stops being ignored) as would the type of service e.g. high speed trains.

Like the network design problem, the line planning problem takes a long-term view, producing plans that may remain unchanged for several years. However, unlike network design, line plans may be relatively easily changed, and many considerations for valuing a line plan are combinatorial in nature (i.e. how the different lines "fit together"). Therefore, mathematical optimization has been successfully applied to many line planning problems in railway transport. See Schöbel [2012] for a recent review of the literature on line planning, primarily in railway transport.

From a passenger point of view, in a city, the line plan is often expressed as a map showing the physical network and the individual routes indicated. See Figure 2.2 for an example showing (some elements of) the line plan for commuter trains in Copenhagen, Denmark. In this particular network, routes and stopping patterns are part of the line planning problem, but all lines offer the same type of service by running at the same speed and with (almost the) same types of units.

For an example of another dense map of lines, see Figure 2.3 showing lines from the Hong Kong MTR; here there are no stopping pattern considerations in the current planning as lines stop at every station they pass.

In both examples, decisions at the network design level have strong implications for what is possible in line planning. For example only certain stations in networks have the necessary infrastructure to be end stations where lines terminate. Furthermore, stations may be visited by several lines travelling in different directions, but due to track layout it may currently be impossible for trains arriving in certain directions to continue in other directions, therefore limiting line possibilities.

For a region, the full line plan problem may be decomposed by service type; for example, inter-city train lines may be planned independently of local trains,

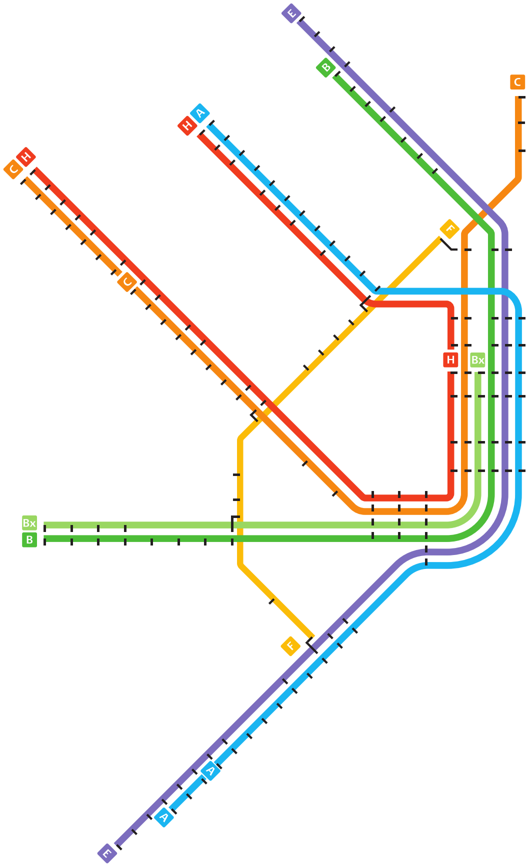


Figure 2.2: DSB S-tog network of Copenhagen; adapted from DSB [2015b]

even though they may share infrastructure and could be planned conjointly. However changes can be made; a station once bypassed by inter-city trains may be stopped at in a new plan, and in many cases such decisions are made on an individual basis (making a small change to an existing plan rather than creating a new plan).

Schöbel [2012] categorize line planning approaches into those that are cost oriented or passenger oriented. Generally, in any line planning problem, there is an estimate of passenger demands for different trips, and a line plan must cater to every such demand.

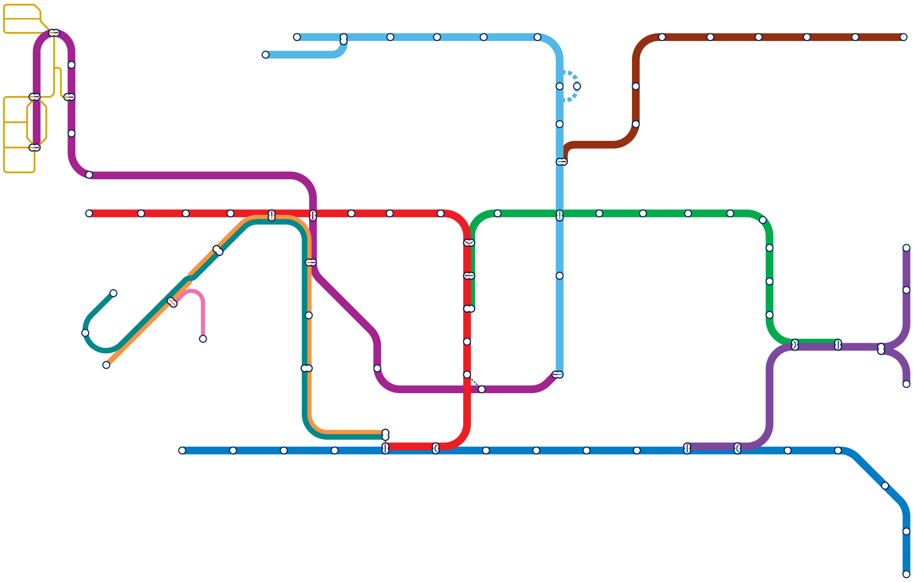


Figure 2.3: MTR system map in Hong Kong. Adapted from MTR [2016]

A cost oriented approach aims to find a line plan with minimal cost to the operator while guaranteeing all passengers are transported. A limitation here is that some passengers are likely to be particularly poorly served. Another limitation is that operator costs can not be precisely known when only planning the line plan; costs almost certainly depend upon more detailed plans that are not known at the line planning stage. Therefore, costs must be estimated. Goossens et al. [2006] for example model the problem from an operator-cost perspective, in the (uncommon case) of a problem where stopping pattern is a decision variable.

In contrast a passenger oriented approach aims to minimise some measure related to the passenger, such as the total travel time for all passengers. A weakness here may be that solutions are impractically costly for the operator, and an operator cost limit is often imposed. However another weakness is that many passenger related measures can not be accurately assessed with only a line plan, as things such as travel time can depend on the coordination between trains of different lines which is not specified in the line plan, and running time on trains between stations can depend on timetabling decisions. Some authors [Bussieck et al., 1997, for example] simplify the passenger-oriented objective by maximizing the number of direct travellers the line plan caters to, while in work such as Borndörfer et al. [2007] passenger travel time in trains is modelled but time spent waiting when transferring is not.

We see then that the line planning problem has conditions imposed on it by decisions made earlier in planning, and estimates must be made for decisions to be made in later planning. A similar pattern can be observed in later planning problems.

2.3 Timetabling

The timetabling problem is perhaps the most studied in the literature from an operations research point of view, and is perhaps the most "passenger-visible" component of a railway plan. While the line plan specifies only routes and operating frequencies, the timetable must assign specific times to trains throughout an operational period. This time allocation must interact with other train systems, such as freight rail, and in an area with several operators of passenger railways, with each other.

Continuing with the example of the Copenhagen S-tog railway network, Figure 2.4 shows part of the timetable for a single line in the S-tog network, showing the minute of the hour that the train visits stations, with trains travelling from Hillerød to Solrød Strand to the left, and the opposite direction to the right. In this timetable, some stations are marked but bypassed (Virum, Sorgenfri etc.) by this particular line, although other lines will stop at these stations. This is a feature of the line plan that is then respected by the timetable.

Also, it is noteworthy that there are six trains' scheduled every hour (also defined in the line plan; other lines operate at different frequencies), and the exact same times repeat every hour. However at the lower end it should be noted that every second train stops at an earlier end station (Hundige rather than Solrød Strand). Nonetheless, the timetable for this line (and every line) repeats every hour. In fact the current S-tog timetable repeats every 20 minutes, and would therefore be referred to as a cyclic or periodic timetable. Other timetables may not have this periodic feature, but it is common in most examples of timetabling.

In contrast to the regularity of the S-tog network, Figure 2.5 shows the timetable for a single line travelling southward toward Wellington city in

A Hillerød

02	12	**22**	32	**42**	52	Hillerød	45	**55**	05	**15**	25	**35**
07	17	**27**	37	**47**	57	Allerød	38	**48**	58	**08**	18	**28**
12	22	**32**	42	**52**	02	Birkerød	33	**43**	53	**03**	13	**23**
17	27	**37**	47	**57**	**07**	Holte	28	**38**	48	**58**	08	**18**
						Virum						
						Sorgenfri						
22	32	**42**	52	**02**	12	Lyngby	23	**33**	43	**53**	03	**13**
25	35	45	55	05	15	Jægersborg	21	**31**	41	**51**	01	**11**
						Gentofte						
						Bernstorfsvej						
30	40	**50**	00	**10**	20	Hellerup	15	**25**	35	**45**	55	**05**
32	42	**52**	02	**12**	22	Svanemøllen	13	**23**	33	**43**	53	**03**
34	44	**54**	04	**14**	24	Nordhavn	11	**21**	31	**41**	51	**01**
37	47	**57**	07	**17**	27	Østerport	09	**19**	29	**39**	49	**59**
39	49	**59**	09	**19**	29	Nørreport	06	**16**	26	**36**	46	**56**
41	51	**01**	11	**21**	31	Vesterport	04	**14**	24	**34**	44	**54**
44	54	**04**	14	**24**	34	**København H**	03	**13**	23	**33**	43	**53**
46	56	**06**	16	**26**	36	Dybbølsbro	59	**09**	19	**29**	39	**49**
49	59	**09**	19	**29**	39	Sydhavn	57	**07**	17	**27**	37	**47**
50	00	**10**	20	**30**	40	Sjælør	55	**05**	15	**25**	35	**45**
52	02	**12**	22	**32**	42	Ny Ellebjerg	53	**03**	13	**23**	33	**43**
54	04	**14**	24	**34**	44	Åmarken	51	**01**	11	**21**	31	**41**
56	06	**16**	26	**36**	46	Friheden	49	**59**	09	**19**	29	**39**
58	08	**18**	28	**38**	48	Avedøre	47	**57**	07	**17**	27	**37**
01	11	**21**	31	**41**	51	Brøndby Strand	45	**55**	05	**15**	25	**35**
03	13	**23**	33	**43**	53	Vallensbæk	42	**52**	02	**12**	22	**32**
06	16	**26**	36	**46**	56	Ishøj	40	**50**	00	**10**	20	**30**
08	18	**28**	38	**48**	58	Hundige	37	**47**	57	**07**	17	**27**
	21		41		01	Greve		**44**		**04**		**24**
	23		43		03	Karlslunde		**42**		**02**		**22**
	27		47		07	Solrød Strand		**38**		**58**		**18**

A Hundige / Solrød Strand

Figure 2.4: Timetable for a single S-tog line. Adapted from DSB [2016]

New Zealand. Here the trains do not operate in a regular fashion but rather operate approximately every thirty minutes in the morning, though timings vary, and only two train services are offered between midday and midnight. The difference in morning and evening frequencies reflect commuter demands, and the varying timings may reflect the sharing of infrastructure with freight railway, and a lack of emphasis given to providing a regular service to commuters.

However, this timetable presentation gives only a macroscopic view of the detail of movements in the network. For example dwell times at platforms are not specified; platforms themselves are not specified. To be consistent in avoiding any conflicts, where two trains are planned to use a piece of infrastructure within too tight a time window, greater detail is required. When modelling the timetabling problem there are different levels of detail that may be considered. A high-level view may be taken where only times between stations are decided upon while exact resource utilization through stations, for example, are ignored, or a more detailed consideration may be undertaken.

To Wellington Monday to Friday

	AM				PM	
Masterton	5.45	6.21	6.48	10.25	3.40	8.20
Renall Street	5.48	6.24	6.51	10.28	3.43	8.23
Solway	5.51	6.27	6.54	10.31	3.46	8.26
Carterton	6.03	6.39	7.06	10.40	3.56	8.35
Matarawa	6.10	6.46	7.13	10.46	4.03	8.41
Woodside	6.16	6.52	7.19	10.51	4.09	8.46
Featherston	6.27	7.03	7.30	11.01	4.20	8.56
Maymorn	6.42	7.18	7.45	11.16	4.35	9.11
Upper Hutt	6.56	7.32	7.57	11.27	4.46	9.22
Waterloo	7.12	7.50	8.13	11.40	4.59	9.35
Petone	7.18	7.58	8.19	11.47	5.05	9.41
Wellington	7.28	8.08	8.29	11.59	5.17	9.55

Figure 2.5: Timetable for a train line into Wellington, New Zealand. Adapted from Metlink [2014]

Lusby et al. [2011] provide a review railway track allocation, which includes timetabling with a microscopic viewpoint.

Cacchiani and Toth [2012] review the literature of train timetabling, considering both nominal and robust approaches. In the nominal case, they identify two approaches; those based on models for the Periodic Event Scheduling Problem (PESP) of Serafini and Ukovich [1989], and approaches that are aperiodic. The authors suggest that one source of aperiodicity in a resultant timetable is in a multi-operator environment, where several operators each supply a preferred, periodic timetable, to an infrastructure manager. To create a complete, non-conflicting timetable the manager must modify those original timetables, making them aperiodic. An obvious example of such an occurrence is infrastructure shared between passenger rail services and freight rail services.

An example of an acyclic modelling of a timetable is work by Oliveira and Smith [2000], who model trains schedules on a single track corridor as a job-shop problem, where each train is required to perform a number of jobs on different machines. Here machines corresponding to infrastructure, jobs to utilizations, and the required sequence is defined by the train's intended trip.

2.4 Rolling stock management

A train timetable dictates train journeys and each must be operated by one or more physical train units. Rolling stock scheduling is the problem of determining which train unit(s) should operate on each of those train journeys.

A rail operator likely has several different types of rolling stock unit, and the line plan and timetable place some restriction on which units are appropriate for each journey. This may be due to required speed, infrastructure restrictions, or required passenger capacity. If multiple units are assigned to the same trip, they are physically attached to each other in a *composition*, which applies additional constraints.

Over a time horizon, each train unit is assigned some sequence of trips it will operate. Between trips, there must be sufficient time for the unit to negotiate the terminal station from its arrival platform to its departure platform. If the unit is part of a composition that will change, there must also be time for the physical detachment and attachment of vehicles. Fioole et al. [2006] give a model for train scheduling that also includes composition changes during journeys at network bifurcations. For example a train composed of two train units travelling from station A could stop at station B where the rail network splits. The units could be detached with the front continuing one direction to station C while the rear continues in another direction to station D.

In some rail systems, trains are not composed of individual self driving vehicles but of a powered locomotive and a number of passive carriage units. In this case many different compositions are possible.

Over some time horizon, many different considerations are important. Each unit may for example be required to visit cleaning facilities, refuelling facilities, de-icing facilities at regular intervals, located at different locations in the network. There may also be maintenance requirements necessitating taking units out of circulation after travelling a certain distance to visit a maintenance location. These requirements may then require that a unit is assigned a variety of journey types in different parts of the network rather than exclusively operating in some small region.

There are additional sub-problems that may be considered part of rolling stock planning. One example, though which could also be classified as an extension of the timetabling problem, is the allocation of platforms at large stations, termed by Train Platforming Problem by some authors [Caprara et al., 2011, for example], where arrival and departure times are specified by a timetable but there is freedom in assigning platforms to trains. Similarly, the parking of units for periods when they are not in use can be a standalone subproblem, where a station depot may consist of a number of long single tracks that several units can be parked on in a last-in first-out stacking manner. While if taking a more abstract or high level view of planning it may be assumed that any rolling stock plan that requires units being parked in and retrieved from shunting depot facilities will be feasible, it may in fact be impossible to store and retrieve units in the required order, or it may be possible only with significant unit movement "reshuffling" them in the depot. These problems must also be considered. Freling et al. [2005] define the Train Unit Shunting Problem as two subproblems; matching arriving and departing train units for shunting, and physically parking those units on shunting tracks. Kroon et al. [2008] consider these two sub-problems in an integrated manner.

2.5 Crew planning

All planned train movements have some corresponding requirement for crew. All movements require a driver, and some may also require conductors and other passenger-serving crew members. Crew planning is required to ensure that every trip is covered by the required crew members, each with the necessary skills or qualifications for that trip.

From an individual crew person's point of view, each requires a block of work (in some time period) consisting of a sequence of train trips with considerations given to requirements such as break time, shift length, and initial and final location. A common planning method is to first create anonymous crew schedules that cover every required train journey, and then later assign those schedules to individual crew members. The first step is (sometimes) referred to as crew pairing and the second step crew rostering.

Caprara et al. [2007] give an overview of railway optimization problems, and describe these two distinct stages of crew scheduling (pairing) and crew rostering. A natural way to model crew scheduling is as a set partitioning problem (or set covering problem). That is, to select from a large set of possible work-shifts (of trips) a subset such that each trip is contained exactly once in their union.

Crew planning for railway shares features to the well-studied crew planning problems in airlines, although there are some differences. For example airline work duties often include overnight stays in remote locations which can incur significant cost.

Optimization methods are heavily used in the crew scheduling process by some rail operators. For example Abbink et al. [2005] describe what they call a reinvention of crew scheduling at Netherlands Railways which includes the use of optimization software package TURNI, which is also used by DSB S-tog [Jespersen-Groth et al., 2009].

Kohl and Karisch [2004] present an overview of approaches for the crew rostering problem, and describe their crew rostering system that is used by eight European and American airlines, and by Swedish rail operator SJ and German rail operator Deutsche Bahn.

2.6 Disruption management

During daily operations, events often occur that result in plans being inoperable exactly as specified. These events may be minor, such as delays, that can be addressed by minor changes to later event times, but they may also be major incidents that require large changes to plans and possibly cancellation of train services. During recovery planning, decisions are made altering plans or creating new plans, during operation, to operate during the unexpected scenario and return to normal operations some time afterwards.

In examples such as infrastructure failure requiring temporary closure of a track section, immediate, significant changes to plans are required. Often the duration of the closure is unknown, and new incidents may occur, so the recovery must be performed in an ongoing and flexible manner.

This recovery planning can require significant changes to all types of plans, including the rolling stock and crew, and it may be true that formerly hard constraints can be slackened. Similarly, objectives may be different compared to when plans were originally created; now, the goal may be to create a feasible plan that can return operations to their "pre-incident" state easily.

Cacchiani et al. [2014] survey the literature on rescheduling in railway operations, considering those that focus on the timetable, the rolling stock, the crew, or take an integrated approach. As an example, Nielsen et al. [2012] present a rolling time-window method for the rescheduling of rolling stock units in the case of a track-closure disruption, where plans are updated as more information about the likely duration of the disruption becomes available. Here, a modified timetable is taken as input and updated (as input) as the scenario develops, while crew planning is left for a later component of rescheduling. Rezanova and Ryan [2010] focus on the crew rescheduling, considering the problem of repositioning crew who end duties at the wrong location to continue their next duty, due to rescheduling, and assigning new crew to trips lacking crew. As an example of an integrated approach, Walker et al. [2005] consider rescheduling the timetable and crew simultaneously, in a single track context with limited overtaking sidings.

Taking a passenger-orientated view, and in the context of smaller delays, Ginkel and Schöbel [2007] study the problem of whether trains (or buses) should have imposed delays and wait for feeder vehicles that are delayed, or not. If the vehicles are not made to wait then transferring passengers on the incoming delayed vehicle will miss their transfer, while if vehicles are made to wait then more passengers experience a delay.

An approach synergistic to disruption management is considering robustness in the earlier planning process. This is an attempt to include protection against some types of disruptive events in plans, so that plans can absorb things like delays with no or minor alteration, or so that rescheduling around large scale disruptions is easier or cheaper than if not considered. A more comprehensive description of robustness considerations in railway planning is given in Chapter 6.

CHAPTER 3

Methods Employed

3.1 Linear programming

Linear programming is a method of mathematical optimization which described generally is the selection of the best element of some defined set, given a function for measuring the value of an element. We may consider set $\mathcal{P}$, a function $f\colon \mathcal{P} \to \mathbb{R}$, and then define the mathematical optimization problem:

$$\underset{x \in \mathcal{P}}{\text{maximize}}\, f(x)$$

Many industrial decision problems can be expressed in such a way, at the risk of being so general as to be unsolvable in large scales. By restricting ourselves in the definition of $\mathcal{P}$ and f, we can describe a smaller variety of problems but ones that we can more readily solve.

Let x be a vector of unknown values to be determined; a *decision vector*. Let $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$ be known real vectors, and let $A \in \mathbb{R}^{m \times n}$ be a real matrix. An Linear Programme (LP) is then expressed in standard form as follows:

$$\begin{aligned} \text{minimize} \quad & c^{\mathsf{T}} x \\ \text{subject to} \quad & Ax \leq b \\ & x \geq 0 \end{aligned}$$

The set of all feasible solutions ($\mathcal{P}$) is all possible values x that satisfy the given conditions, and, if non-empty and bounded, there is an element (or some elements) that provide a minimum value $c^\intercal x$. The feasible set is a convex polytope.

LPs are easily formulated and easily solved in polynomial time (see Sipser [2006] for a reference on algorithm complexity, not generally considered further in this work). One common method for solving such problems is using the simplex algorithm, which finds an optimal extreme (corner) solution.

Chapters 7, 8 and 9 all make use of LP formulations in some way for problems in railway line planning, timetabling and rolling stock assignment. Generally, however, the LP formulations can not sufficiently describe the problems but rather a relaxation of the problems. A relaxation can provide useful bounding information for the more refined, complete formulation.

3.2 Integer programming

A weakness of LP models is that the decision variables all have real values, and due to the convexity of the set of feasible solutions, there are always solutions *between* other solutions. If a and b are in feasible set $\mathcal{P}$, then by convexity any $\lambda a + (1-\lambda)b$, where $0 \leq \lambda \leq 1$, is in $\mathcal{P}$.

This is a limitation when modelling many industrial problems that often feature discrete elements such as decisions or costs. For example if formulating a problem related to finding a route for a train unit in a railway network, where there are two distinct paths that may be chosen, the only valid solutions are to choose exactly one of the two paths, but there is no valid solution to half choosing each of the paths.

Using the tools of LP modelling, an obvious extension is to require elements of x to be integer; that is, given the polytope $\mathcal{P}$ we seek some $x \in \mathcal{P} \cup \mathcal{Z}^n$. If we ignore the integrality requirement and solve the problem as an LP (the LP relaxation of the Mixed Integer Programme (MIP)), we will likely find that the solution is not integer. However for any MIP there is some "best" formulation, where the set of feasible solutions exactly coincides with the convex hull of feasible integer solutions; therefore for any objective function the optimal solution to the LP will be integer.

Generally however our formulation does not naturally give LP solutions that are integer, but the objective value for the optimal LP solution provides a bound on the optimal solution (in that it can be no better than the LP solution).

A general approach to solving such problems is termed branch-and-bound; where branching refers to an exhaustive (potentially enumerative) search of the discrete solution space, and bounding refers to culling parts of the search using knowledge from the LP relaxation. Details are not given here, but solving MIP problems via branch-and-bound is well-established; an early survey of branch-and-bound methods is provided by Lawler and Wood [1966].

Another method is the generation of cutting planes: linear constraints that separate some discovered extreme optimal (but non-integer) solution from the convex hull of integer solutions. For a reference on cutting planes, see Wolsey [1998, Chapter 8]. Cut generation is implemented, with branch-and-bound methods, in commercial solvers for MIPs. There are specific methods, but at a high level we may say a *cut* is a linear constraint that we can add to a formulation that is violated by some non-integer solutions, but by no integer solutions. We ourselves however can create cuts, or linear constraints, that are violated by non-integers solutions; in the best case these coincide with the convex hull of integer solutions. We can also reformulate our problem so that the polytope we define is closer to the convex hull; in the LP relaxation, we will see this as an improvement in the bound.

Chapter 7 uses a MIP formulation for the line planning problem. Chapter 8 combines two MIP formulations of different problems; one for line planning and one for timetabling. Chapter 9 compares two different MIP formulations for matching arrival and departure trains at a station.

3.3 Delayed column generation

Some LP problems can be formulated in a way that results in the matrix A having many more columns than rows. In such cases, a possible method for solving problems is to delay the creation of all the columns initially, and generate them as needed. Ideally, the optimal solution will be found with far fewer columns generated than exist.

Exact details are not given here, but see Lübbecke [2010] for an overview of column generation. However, in short, delayed column generation mirrors the revised simplex method for solving LP problems, where at each iteration a subset of columns constitute a basis; a new column is identified to enter the basis and one to leave. This master formulation contains only a subset of all present columns, and new columns are (typically) introduced by be solving an optimization problem that creates one or more columns that could be basis-entering columns for the master formulation if any exist.

A common paradigm is to identify a problem with a certain structure and perform a Dantzig–Wolfe decomposition which is a reformulation of the problem as a master problem with several sub-problems, and solving by column generation. The method originally comes from Dantzig and Wolfe [1960] and has been applied to many problems. Here, the original problem is specified by an A matrix with block-diagonal structure, with an additional number of linking rows. More clearly, the following form is required:

$$\begin{bmatrix} B_1 & 0 & \dots & 0 \\ 0 & B_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & B_n \\ C_1 & C_1 & \dots & C_n \end{bmatrix}$$

Here, block B_i will correspond to one subproblem, and rows corresponding to coupling rows C will be present in the master problem. Each block is, independently and considering its corresponding right hand side, reformulated as a convex combination of its extreme points. If unbounded, a conic combination of extreme rays is also required.

In the case of integer programming, delayed column generation may also be used. In fact a reformulation may result in a tighter model (in terms of lower bound); this is for example true when extreme points for the original subproblem polytope are non-integer, but we can reformulate the problem in terms of a convex combination of *integer* extreme points.

As a very general example, consider the common set partitioning problem of maximizing some function $c^\intercal x$ and requiring $Ax = 1$. Here elements of A are either 0 or 1, as must be elements of x. A partition of a set S is the assignment of the elements of a set into distinct, disjoint subsets, such that every element of S exists in exactly one subset. Here, rows of A correspond to the elements of S and columns correspond to valid subsets, where a 1 in a row and column indicates the presence of appropriate set element in the subset.

In a column generation approach, we would begin with only some of a subset of columns, though taking care to guarantee feasibility, and solve the corresponding LP finding a non-integer solution x, an optimal basis, and a dual vector π. However this is only optimal give that only a subset of columns are present in the problem. For each non-basic column present we could check the reduced cost $c - \pi^{\intercal} z$ for every column c and none would be negative; however, there may be columns not yet present that would give a negative reduced cost. Using π, we attempt to find one or more new columns to introduce to the problem, and continue solving the LP. One identified column will immediate be an entering column for the simplex method.

3.3.1 Subproblem

The subproblem of finding new columns is formulated as an optimization problem. We solve the problem of finding some column z (of matrix A, which we may not have defined explicitly):

$$\underset{z}{\text{minimize}} \quad c - \pi^{\intercal} z$$

Where c is as defined for an LP or MIP above, and π is a dual variable vector provided from the master formulation.

A common problem structure is that in the master formulation variables correspond to paths, and the subproblem can then be formulated as finding a shortest path as an optimization problem.

The resource constrained shortest path problem is that of finding a shortest path in a graph between two given nodes, subject to one or more "resource" limitations on the path. Each arc in the graph is associated with a cost, to be minimized, and several resource measures. Each resource has a limit which must not be exceeded anywhere along the path (although if the arc resource measures are all positive and additive then the limit must simply be respected at the end of the path). Irnich [2008] examines resource constrained paths in depth, considering properties, and modelling and algorithmic implications for different types of resource extension functions, which are the functions associated with each arc for modifying the resource along the path. In a transportation context, resources could for example be related to a maximum distance a train unit could travel before requiring maintenance, or a maximum number of transfers a passenger would accept on a journey. A resource constrained shortest path problem can be formulated as a MIP, but it can also generally be solved with a labelling algorithm, where at each node a list of non-dominated labels must be maintained and pushed to neighbours.

Considering the master problem, if it is a MIP then to find integer solutions we have to implement a branching strategy, which must also be implemented in the subproblem (to avoid finding columns that do not confirm to the branching).

Chapter 9 formulates a problem in a way where delayed column generation is a natural solution method, and solves instances using a resource constrained shortest path sub-problem. However a heuristic method is taken to finding integer solutions to avoid the aforementioned challenge of implementing branching in the sub-problem.

3.4 Multi-objective optimization

We describe LP and MIP optimization above as particular special cases of mathematical optimization. In both cases there was a single objective function, and any two feasible solutions could be compared. However, in many real-world optimization problems, there is more than one relevant objective function, and not always any valid method to compare every pair of solutions. A train operator may wish to minimize both some function of passenger experience and their operator costs, for example minimizing both passenger wait time and crew hours worked. In both cases the two objectives are measured in different units and can not be directly compared.

Let us restrict ourselves to objective functions that can each be expressed in the form $c^\intercal x$. Overall, if we have n objective functions, we may express an overall objective in the form:

$$\begin{pmatrix} c_1^\intercal x \\ c_2^\intercal x \\ c_3^\intercal x \\ c_4^\intercal x \\ \vdots \end{pmatrix}$$

i.e., our objective function is now a linear map to $\mathbb{R}^n$, which we can refer to as the *objective space*. We should note, then, if considering an LP problem where the set of feasible solutions is convex, that the convexity is conserved by the map.

Generally, as the objectives are non-comparable and we do not have a preference for either one, then when comparing two solutions we may say: $x_1 \preccurlyeq x_2$ if, for every cost vector c_i, $c_i^\mathsf{T} x_1 \leq c_i^\mathsf{T} x_2$. We can further state that some solution x_1 *dominates* another solution x_2 if $x_1 \preccurlyeq x_2$, and, for some i: $c_i^\mathsf{T} x_1 < c_i^\mathsf{T} x_2$. Rather than seeking some single optimal solution, one may seek a set of feasible solutions that is not dominated by any other.

Consider Figure 3.1, where for some unspecified LP problem with two objectives the set of feasible solutions is shown in objective space. Any solution is either dominated by some solution on the dark black line, or lies on the dark black line and is not dominated by any other. Furthermore, any of the non-extreme non-dominated solutions can be expressed as a convex combination of the adjacent extreme solutions. Therefore, generally we need only find all extreme non-dominated solutions.

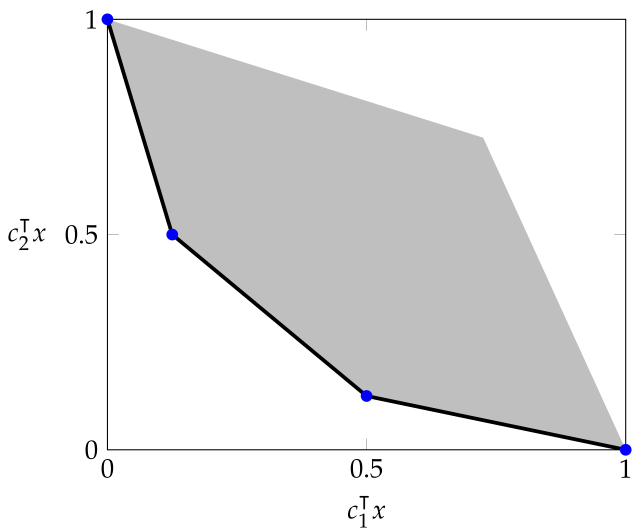


Figure 3.1: The feasible solutions to an LP, in objective space, shaded in grey. Non-dominated solutions sit on the dark black line, and non-dominated solutions are at the blue points.

It is also the case that if we reformulate the problem as a single-objective problem by weighting objectives, the optimal solutions will be one of the non-dominated extreme points. That is, we somehow select $\lambda_i > 0$ for every objective and minimise the single objective $\sum_{i=0}^{n} \lambda_i c_i^\top x$. This suggests an algorithm for finding all non-dominated extreme solutions by careful appropriate calculation of weights.

In the case of a MIP, there are more complications. See Figure 3.2 showing the convex hull of solutions. Unlike in the LP case, it is not sufficient to find non-dominated extreme solutions, as these alone are not sufficient to construct solutions dominating all others (as we can not simply take convex combinations of adjacent solutions). Now, non-dominated solutions do not necessarily sit on the convex hull of integer solutions. Potential locations of non-dominated but non-extreme solutions are shown in red on Figure 3.2; they can not be discovered by taking a weighted sum of objective functions as for any weighting an extreme point solution will be superior.

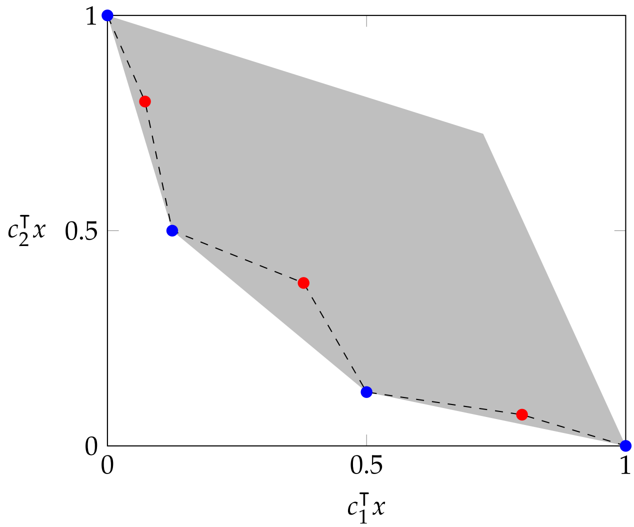


Figure 3.2: The convex hull of integer solutions to a MIP in objective space, shaded in grey. Non-dominated extreme solutions are in blue, with additional non-dominated solutions in red.

More complicated methods are required if all such solutions are to be discovered. Ehrgott [2006] gives an overview of different scalarization methods for MIP problems with multiple objectives.

Chapter 7 formulates a problem with two competing objectives, seeking solutions that are acceptable for both. However a full enumeration of all non-dominated solutions is not sought but a range of solutions that are close to being non-dominated is discovered.

3.5 Heuristics

The methods we have described so far for solving optimization problems are *exact*, in that they can guarantee an optimal solution if one exists, or guarantee that one does not exist. Heuristic methods are a different class of methods for approaching problems that do not provide such guarantees, but have other attractive features such as faster running time.

One common class of heuristic method consists of some constructive component and a subsequent search component. Initially a feasible solution must be constructed which for some problem types is not at all trivial. Subsequently, some search by modifying the solution occurs until a termination criteria is met. The simplex method in fact can be described in this way, as it first finds a basic feasible solution and then iteratively moves from solution to adjacent solution such that the objective function improves. Termination occurs when a solution is reached that has no improving adjacent solution. For an LP such a method is sufficient to find the optimal solution, but for many problems such an approach will not.

As a heuristic, such search where only improving solutions are accepted is referred to as a hill-climber, and will guarantee a locally optimal solution. However many classes of problems have the feature of local optimal solutions that are better than neighbouring solutions, but which are not globally optimal. It is in fact easy to construct optimization problems for which a locally optimal solution is arbitrarily worse than the globally optimal solution.

A description of several search heuristic approaches is given by Pirlot [1996]. The authors describe the meta-heuristic search methods of tabu search, genetic algorithms, and simulated annealing. These all avoid the problem of only finding local optimal solutions, to some degree, although all will in general not guarantee global optimality without substantial running time. Both genetic algorithms and simulated annealing are examples of a common theme of heuristics inspired by natural processes. Simulated annealing in particular is close to a simple hill climbing search, except worsening solutions are also accepted probabilistically. As the search progresses, a *temperature* parameter that controls this probability decreases, such that at the termination of the method no worsening solutions are accepted. Černý [1985] present early work applying such a method to the travelling salesman problem, of constructing a tour visiting a number of cities, with the inspiration from statistical thermodynamics clearly explained.

A recent approach is the hybridization of mathematical programming methods with meta-heuristics, creating what are termed *matheuristics*. Boschetti et al. [2009] survey the developments in the hybridization of meta-heuristics with other methods, some of which are related to, or solvable with, mathematical programming methods. A simple example is a meta-heuristic search where solutions are sought in some neighbourhood defined by the fixing of some problem variables leaving a small number free, and using a MIP solver to find a solution in this neighbourhood. Such a method permits the exploration of a complex neighbourhood with the powerful MIP solver, while avoiding the intractability of attempting to solve the entire problem as a MIP by restricting the problem size tackled by the solver.

In this thesis, we will consider only heuristic methods that are based on exact methods; that is, we generally start from some exact but intractable formulation for the problem and from this derive some heuristic method.

One simple example heuristic applied to a delayed column generation method is to solve the master problem as a LP, and then at optimality, find the best integral solution present given the currently discovered columns. In the worst case there may not even be a feasible integer solution, and if there is it may be far from optimal. However for many problems, reasonable solutions may be expected with such a method, and the quality of the solution can be assessed given the LP bound. This approach is taken in Chapter 9, avoiding the complexity of generating new columns while branching.

Appendix A describes a decomposition of a problem into several sub-problems we solve in sequence, some exactly and some heuristically, which is overall a heuristic method for the problem. The simulated annealing method described in the appendix moves to a neighbouring solution by a random insertion of the removed vehicle trajectories. However a different matheuristic method was also tested of removing the neighbouring vehicle trajectories, generating several possible trajectories, and using an exact MIP formulation for the reinsertion.

Chapter 7 describes a method based on an exact approach but with several heuristic elements, and Chapter 8 describes a method consisting of two exact models combined in a heuristic way.

Chapter 4

Problems Considered

The thesis concerns itself with several related projects in railway optimization, where different methods have been applied, with different objectives and approaches.

4.1 Robustness in railway planning

Chapter 6 reviews the literature of methods that include some notion of robustness in railway planning problems. Given the context of this thesis as part of the RobustRailS project, we considered it relevant to investigate the usage of robustness in railway optimization. In particular we considered the extent to which there are concepts of robustness specific to *railway*, that consistently span the different planning problems described in the previous chapter.

Here the motivation is to consider different approaches, consider their similarities throughout the planning horizon, and originally was to discover if there can be some all-encompassing definition of robustness for railway planning. We conclude, however, that even for a single planning problem in railway there are many different, sometimes contradictory, definitions of robustness. Despite some authors' attempts to *define* robustness for particular planning problems, we conclude that no single definition can exist that satisfies all stakeholders even for the particular problem, and certainly can not for railway planning in general.

The key findings here for railway robustness are that:

- given the many different and sometimes contradictory definitions of robustness, no clear single definition exists and is unlikely to be created;
- integrated approaches are becoming more common, not necessarily for solving problems but in considering robustness of one problem by its influence on another;
- the railway timetable is the most studied in terms of robustness, and the most accepted planning level where robustness is to be found.

We expect to see more integrated approaches, in railway planning in general and also those planning for robustness, and more consideration of robustness in all areas of railway planning.

4.2 Line planning

As outlined in the previous chapter, the line planning problem is that of selecting or discovering some pool of lines that meet some operational requirements and service passengers. Chapter 7 presents a line planning model and application to the S-tog commuter network in Copenhagen. The problem has some unique features that, despite the small size of the network compared to some other networks overseas, make it an interesting study. In addition we target the passenger cost with particular consideration for line switching, finding passenger routes based on input demand for travel between every pair of stations (origin-destination pairs).

The Copenhagen S-tog system is somewhat unusual in having trains with different stopping patterns, that are not classified separately as "inter-city" or "local" trains, as might exist in some other networks, but rather lines that may skip some stations but stop at different stations that other lines skip. The possibility of arbitrary stopping patterns greatly increases the solution space; for every route, there are 2^k different lines where k is the number of skippable stations on the route, while in some different problem the route alone would uniquely define the line.

Furthermore unlike some other networks where every line operates at the same frequency, or at any frequency, in the S-tog network there are distinct valid frequencies (which differ for different parts of the network). Therefore we model the frequencies discretely.

Finally, given the very different frequencies of lines in the network (as high as twelve trains per hour and as low as three per hour) we model passenger waiting times with a frequency dependence.

The model presented is a multi-commodity flow problem for passengers, with variables linking flows to line variables, and some additional constraints imposing the validity of the resultant line plan. Included in this are some consideration for ensuring the feasibility of creating a timetable for the plan but forbidding certain sets of lines that are determined to be incompatible for timetabling. Lines are chosen from a pool rather than allowing any possible route and stopping pattern, both significantly reducing the problem size and avoiding determining complicated rules for what constitutes a valid stopping pattern by defining the limited pool with the help of DSB. We chose to model passenger flows between origin and destination as non-integer and therefore splittable (i.e. multiple paths may be used between an origin and destination). Given that the original passenger data is non-integer (as it comes from modelled flows), and the assumption that some passengers would indeed choose to take a longer route if the shortest were crowded.

An obvious reformulation of the model, not presented in the chapter, is to a flow-based rather than arc-based model, with every possible path between every origin and destination being a decision variable, and for every such origin-destination requiring that in aggregate a path is chosen for each (but not one single path; again allowing splitting). In fact, in work not presented here we formulated such a model and rather than using a delayed column generation method, pre-generated every possible route but including some restrictions on valid routes. The restrictions resulted in roughly 2 million routes for the S-tog problem with 170 lines in the pool, which resulted in a problem with a solvable LP relaxation. However this did not provide a (significantly) tighter bound than the arc-flow formulation (and just the heuristic restriction of paths could explain any bound change as the problem is artificially tightened). Given that we do not seek integer passenger paths but rather integer line decisions, it is the linking between passengers and lines rather than the formulation of passenger flows that seems most relevant for targeting a bound improvement.

In Chapter 7, focus is given to the passenger-oriented objective with some mention of the other, operator-oriented objective. However given the presence of a different, competing objective for the operator, this could be seen as a bi-objective problem. Many solutions are shown which sit somewhat close to the efficient frontier of non-dominated solutions given the two objectives, but the finding of them is not clearly explained in the chapter. Refer to Appendix C for a brief overview of the algorithm used to find these solutions.

4.2.1 Future application

The model has only been applied to the S-tog case in Copenhagen. It would be interesting to apply a similar approach to a different network, considering freely-routed passenger travel time with a frequency dependent estimate of wait times to a different network. However, this only makes sense in a network with significantly different frequencies for different lines; if all line frequencies are the same or very similar, then taking a fixed wait penalty is more reasonable.

In the Copenhagen case, more work could be done considering the affect of there being multiple valid paths for some given passenger, with different travel times, but initially coincide. That is, the paths only differ after some initial journey, and the passenger may wait to select the route later in the journey when more information about train timings is apparent.

Finally, the predetermined Copenhagen line pool only contains 170 lines, which is diverse enough to ensure a large variety of solutions, but far fewer than the number of possible lines. Investigating more diversity would be interesting, but doing so would require both investigating what rules define the validity of individual lines, and a more comprehensive method for forbidding sets of timetable-incompatible lines (as currently, these sets are determined with the pool, by hand).

4.3 Integrating timetabling and line planning

As is discussed in Chapter 7, one concern when creating a line planning is that no feasible timetable can be created for it. Even if timetables can be created, they may all be of poor quality when considering cost to the operator, or robustness, or some other measure.

We adapt a timetable model designed for trains in Belgium to the Copenhagen S-tog problem. Given a line plan, specifying the line routes and frequency for trains, the timetable problem is then to assign times to every trip between stations along that trip. In work not presented in this thesis, we derive a single full model of the line planning and timetabling problem, combining elements of both the line planning model of Chapter 7 with the adapted timetable model. However due to its size and complexity we did not consider solving it directly. Instead we present a heuristic method for adapting a line plan to create on that is feasible, or better, for timetabling.

The combined method is presented in Chapter 8. Earlier results just comparing different line plans, and the significance of turning restrictions in the S-tog problem, are presented in Appendix B.

An additional measure in Appendix B: the number of train interactions, is in fact a valid objective for the line planning model despite being considered in a timetable context. However finding line plan solutions that minimize that measure did not lead to results of note and it is not used in Chapter 8.

In the S-tog network, with thirty trains travelling through a central corridor every hour, there are very tight buffers between trains. Coupled with the driving time and turning restrictions, many line plans are difficult to timetable, and necessarily have trains running very close together.

However, the skipped stations in the S-tog network present an opportunity for modifying the running time as skipping stations saves time (and stopping requires time). We use the timetable model to create timetable solutions, in Chapter 8 always requiring feasibility but also potentially considering infeasible plans, and identify lines to modify. However the modification is performed by the line planning model, creating a plan similar to the original but with some appropriate modification.

This iterative method is a partial integration of line planning and timetabling, and is a heuristic. No information on overall optimality is possible. However, targeting buffer times in the timetable model, we are able to modify line plans such that the upper bound for buffer times is reached while maintaining a line plan similar to that given originally.

Not presented in Chapter 8 or Appendix B, the timetable model allows validation of the waiting time estimates made by the line planning model. We find that, overall, the line plan estimate of waiting time is surprisingly accurate; for the several tested plans, the line plan estimate was within 5% of that taken from the resulting timetable. However unsurprisingly, some passengers wait significantly longer than estimated (waiting almost the full headway time) while others have very short transfers.

4.3.1 Additional and future application

The timetable model was originally created for the Belgium network, where timetable creation and robustness have been well studied [Sels et al., 2016, Vansteenwegen and Van Oudheusden, 2006, for example]. Here we have applied it to the Copenhagen S-tog network. It may be interesting to apply a similar integrated approach to the network in Belgium. However there are significant differences in the structure of a line plan, and it may not make sense to modify a line running time by adding or removing stations. Nevertheless, there may be other identifiable features in the timetable that can be implemented as a line plan modification.

More work could also be done on the single integrated model, both in simplifying it to be tractable and finding a good case study for its application. Significant complexity comes from the line plan model's treatment of passengers; a simpler modelling of passengers would certainly lead to a simple (but not necessarily tractable) model.

As an alternative experiment, we considered more significant changes to plans (and departure from realistic plans) for Copenhagen. As one example, with reference to Figure 4.1, we decomposed the network into an independently running shuttle service in the red central corridor, and lines in the blue finger tracks that either do not enter the corridor, or only partially enter the corridor. Here we find it very easy to create good quality timetables, and can explore a very large range of lines (taking a very large pool), as every finger may be solved independently (after some passenger pre-processing). However, the line plans require many more passenger transfers than before.

However this requires some significant and currently unrealistic turning capacity for trains at the ends of the central corridor. Currently it would not

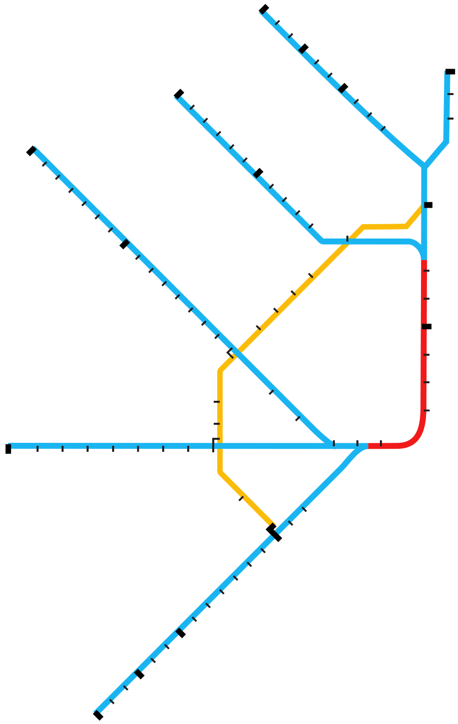


Figure 4.1: Structure of the S-tog network, with the red central corridor, blue fingers, and yellow circle. Adapted from DSB [2015b]

be possible for so many trains to turn at either end, as currently trains only travel through the stations.

4.4 Operational station planning

A major terminal station is visited by potentially thousands of trains during the horizon period for detailed planning. There are several related planning problems that must be considered. Here, several related problems are considered as part of a railway planning competition, and one of the individual problems is explored in more detail.

4.4.1 Problem overview

The ROADEF/EURO Challenge 2014 described such a problem to be considered. In the problem, train arrival times and departure times are strictly set (by a timetable). Arriving trains have partial routing pre-determined as they enter the station's area of consideration, but exact routes through infrastructure are not set and neither are platforms (although some platforms are preferred for some arrivals). Similarly, departure trains have partial outgoing route information set, but platforms and precise routes are not decided.

After entering the system, an arriving train must be assigned a conflict-free route through infrastructure to an assigned platform, where it must dwell for between a minimum and maximum time. It may be routed onward to either a depot facility, to one of several maintenance facilities, to single track parking infrastructure, or to a sequence of these. It may be later routed back to a platform to depart as a later departure train. Alternatively, it may remain in the station infrastructure at the end of the problem horizon.

The pairing between arriving train units and departure train units is not pre-determined and must be decided, while meeting certain requirements. For example every departure has a set of types of train unit that are appropriate for its journey. Also, some departures must be assigned more than a single unit coupled together (and some arrivals consist of several units coupled together); the coupling and uncoupling can be performed at certain infrastructure and takes some amount of time. Departures also specify some maintenance requirements, and some train units in the station will be ineligible for assignment to certain departures. However trains can be routed to maintenance facilities to have their (different types) of maintenance levels replenished.

Finally, train units that leave as certain departures in fact return as later arrivals; we are assigning trains to a sequence of train journeys (from the point of view of this station) and as maintenance levels decrease with every outward and returning journey, but a train may visit our station several times, there may be several visits in the planning horizon where we may decide to perform maintenance. Given that there are limitations on maintenance facility capacity, and routing in the station area causes congestion, decisions on when maintenance should be performed for a single (returning) train unit depends on other decisions.

Figure 4.2 (reproduced from Ramond and Nicolas [2014]) shows an example of an instance infrastructure, and it can be seen here that the stations studied are particularly large. In the figure, trackgroups 1, 5, and 6 are entrances to the system, while the others are for moving trains around. There are strict constraints for the usages of infrastructure, and, particularly in trackgroups, complex rules dictating the reservation times of infrastructure and the determination of whether different routes conflict.

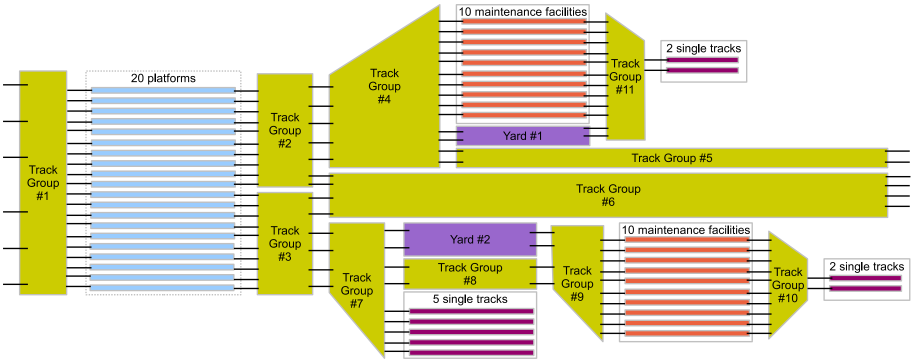


Figure 4.2: Example of ROADEF instance infrastructure

4.4.2 Overall approach

A description of our solution approach is given in Appendix A. The entire problem is very complex (specified in 39 page document: Ramond and Nicolas [2014]), with many details that are difficult to capture and model in some single formulation. We decomposed the problem into a sequence of subproblems we solve sequentially, which provides no guarantee of optimality but, given the ten minute runtime limit for competition entries, we decided it was unlikely that an approach targeting optimality would be possible.

Matching and platforming are formulated as MIPs, while routing is solved heuristically by adding and removing unit routes in a simulated annealing framework. A final heuristic attempts to use remaining time to re-assign costly trains.

An important feature of the problem is that, despite the competition subtitle "trains don't vanish", in the problem a penalty could be paid to make arriving trains "vanish" by cancelling trips, or to leave departures uncovered. For some test instances we could show that there was no feasible solution that avoided using this feature; i.e. there were no solutions that avoided cancelling any arrivals or departures. In our opinion this was a flaw in the problem instances or problem specification.

Our entry was awarded the second place prize for the PhD category.

4.4.3 Matching subproblem

Chapter 9 presents a derivation of and results for two methods for solving instances of the matching problem for the ROADEF challenge. We considered the matching problem to be interesting, and as it naturally formed a first step for a ROADEF challenge solution method, it could be studied alone with only the given (or created) instances. Later sub-problems such as routing depend on decisions made in the matching, platform allocation, maintenance to be undertaken etc. The matching decision, however, captures the complexity of the unit types, compositions and compatibilities, and incorporates maintenance decisions.

A complicating feature of the problem is that later arriving trains in the planning horizon are returning earlier train departures, and if viewed from a unit point of view, we seek a sequence of tours for train units that each begin and end at our station of concern. However not all arrivals are returning departures; some are simply new trains entering the system and becoming available at the station. A problem instance could in fact have no returning departures, and therefore all arrivals would be new well-defined trains to be matched to departures; or an instance could have a very high proportion of returning departures and a solution would imply many returning tours for every train unit.

There are then two "obvious" views of the problem. It can be viewed as a matching between arrivals and departures, requiring additional decisions and constraints for maintenance, unit classes etc. and some complicated additional features to capture the returning trains. Or it can be viewed as a per-unit sequence of tours, where the overall sequence must meet unit maintenance, unit class etc. requirements and the units' sequences must together correctly cover all departures. Given an instance with very few returning departures, the first viewpoint may be most appropriate and the second needlessly complex, while with many returning departures the second is more natural.

We formulate the matching problem from both points of view. In the first case, we formulate a MIP based on the assignment problem (i.e. assigning one departure to each arrival, and vice versa), with additional relatively simple additions to handle the unit classes, maintenance decisions, but rather complex (and weak) constraints to address returning departures. For the second case, we formulate a sequence based formulation, and use a delayed column generation method to create sequences as paths in a graph of arrivals and departures. We show that, given some assumptions on the minimum routing time required between an arrival and subsequent departure for a train, the LP lower bound we find for some instances implies that some ROADEF instances can not be completely solved, even from a matching point of view (as cancellation is required).

4.4.4 Additional and future application

The matching problem formulation has potential application to real matching problems in a real train unit management problem. The ROADEF problem was sponsored by French operator SNCF, presumably inspired by real data. However our investigations to the applicability of the matching problem to problems at DSB revealed some limitations in the problem modelling, at least in a Danish context, given its station-centric viewpoint.

In the problem, and in our matching formulation, all decisions are centred around a single station, and events outside the station are either predetermined or irrelevant. For example, the tours that correspond to a train being assigned to a departure that later return as a different arrival are fixed, and no maintenance is performed on these trips. However one may imagine that a departure that returns as a different arrival corresponds to a matching at a distant station between the departure's arrival there and the departure there of the subsequent arrival. It then seems strange to set all such matchings at every station except for one, and solve matchings at that one in isolation, rather than taking a wholistic view.

A small pilot study was undertaken considering the routing problem at Copenhagen central station, using a simpler matching heuristic than that presented in Chapter 9, and applying the routing heuristic of Chapter A.

However given the complexity of the infrastructure at Copenhagen central, and differences in the underlying assumptions, we concluded that an approach more tailored to the unique characteristics of the station would be more appropriate.

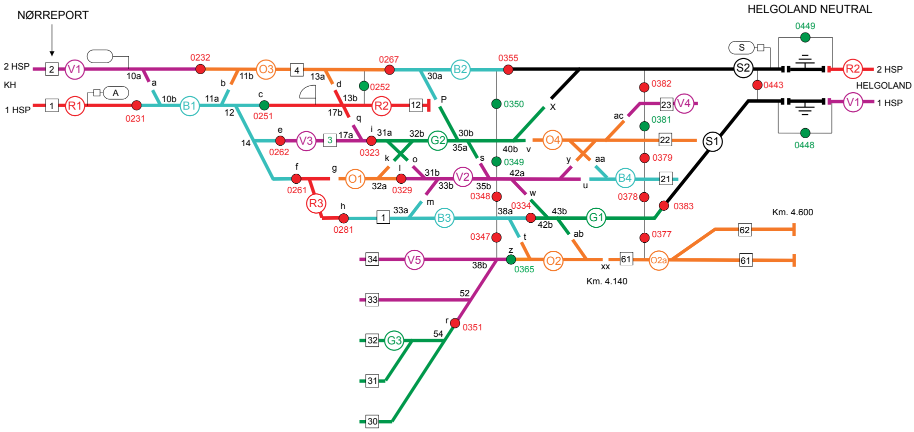


Figure 4.3: Østerport track layout for long distance trains

Figure 4.3 shows a schematic view of the infrastructure at Østerport station, with four platforms, which is connected by double track to nearby Copenhagen central station which has eight platforms. The interaction between these two stations and utilization of the limited connection is very significant, and effectively using the parking facilities at Østerport station (visible at the bottom of Figure 4.3) and at Copenhagen central considering both as a single station area.

4.5 Future work

Specific directions and applications are discussed in the individual chapters of Part II. Here we briefly discuss more general directions for future work.

In the application of optimization methods to railway planning, there is certainly scope for more depth in the exploration of problems. Due to the complexity of railway planning problems, better methods are still sought and generally, planning problems can not be considered solved. Furthermore, due to subtle differences between systems of different countries, and due to differences in underlying assumptions and objective measures used by different operators, new approaches are still relevant.

A clear direction for further work in railway optimization is in greater integration of the different planning problems. The commonly used sequential approach certainly has the potential to negatively impact solution quality. This integration is a common trend and is a current area of research, often considering two problems at once. Currently, due to the complexity of the planning problems alone, complete integration is far from feasible in the near future.

Robustness has become more relevant, and further investigation and implementation of robustness is an area where there is obvious scope for further research. Again, the sequential nature of much of railway planning can be a factor in failing to achieve overall robustness, and greater integration of robustness concepts through planning is an area worth exploring.

Chapter 5

Bibliography

Erwin Abbink, Matteo Fischetti, Leo Kroon, Gerrit Timmer, and Michiel Vromans. Reinventing crew scheduling at netherlands railways. *Interfaces*, 35(5):393–401. 2005.

Ralf Borndörfer, Martin Grötschel, and Marc E Pfetsch. A column-generation approach to line planning in public transport. *Transportation Science*, 41(1):123–132. 2007.

Marco A. Boschetti, Vittorio Maniezzo, Matteo Roffilli, and Antonio Bolufé Röhler. Matheuristics: Optimization, simulation and control. *Lecture Notes in Computer Science*, 5818:171–177. 2009.

Berit Dangaard Brouer, Fernando Alvarez, Christian Edinger Munk Plum, David Pisinger, and Mikkel M Sigurd. A base integer programming model and benchmark suite for liner-shipping network design. *Transportation Science*, 48(2):281–312. 2014.

Michael R Bussieck, Peter Kreuzer, and Uwe T Zimmermann. Optimal lines for railway systems. *European Journal of Operational Research*, 96(1):54–63. 1997.

Valentina Cacchiani, Dennis Huisman, Martin Kidd, Leo Kroon, Paolo Toth, Lucas Veelenturf, and Joris Wagenaar. An overview of recovery models and algorithms for real-time railway rescheduling. *Transportation Research Part B: Methodological*, 63:15–37. 2014.

Valentina Cacchiani and Paolo Toth. Nominal and robust train timetabling problems. *European Journal of Operational Research*, 219(3):727–737. 2012.

Alberto Caprara, Laura Galli, and Paolo Toth. Solution of the train platforming problem. *Transportation Science*, 45(2):246–257. 2011.

Alberto Caprara, Leo Kroon, Michele Monaci, Marc Peeters, and Paolo Toth. Passenger railway optimization. *Handbooks in operations research and management science*, 14:129–187. 2007.

Vladimír Černỳ. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of optimization theory and applications*, 45(1):41–51. 1985.

George B Dantzig and Philip Wolfe. Decomposition principle for linear programs. *Operations research*, 8(1):101–111. 1960.

DSB. Annual report. URL `http://www.dsb.dk/Global/PDF/%C3%85rsrapport/2015/DSB%20Annual%20Report%202015.pdf`. [Online; accessed April 1, 2016]. 2015a.

DSB. Linjeføringskort for s-tog 2015. URL `http://www.dsb.dk/Global/Trafikinformation/Kort/Linjef%C3%B8ringskort%20for%20S-tog%202015.pdf`. [Online; accessed April 1, 2016]. 2015b.

DSB. S-tog køreplaner 25 january 2016. URL `http://www.dsb.dk/Global/PDF/Koereplaner/S-tog/2016/S-tog-25jan.pdf`. [Online; accessed April 1, 2016]. 2016.

DSB S-tog a/s. DSB s-tog a/s annual report. URL `http://www.dsb.dk/global/om%20dsb/rapporter/dsb_s-tog_annual.pdf`. [Online; accessed April 1, 2016]. 2011.

Matthias Ehrgott. A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research*, 147(1):343–360. 2006.

Pieter-Jan Fioole, Leo Kroon, Gábor Maróti, and Alexander Schrijver. A rolling stock circulation model for combining and splitting of passenger trains. *European Journal of Operational Research*, 174(2):1281–1297. 2006.

Richard Freling, Ramon M Lentink, Leo G Kroon, and Dennis Huisman. Shunting of passenger train units in a railway station. *Transportation Science*, 39(2):261–272. 2005.

Andreas Ginkel and Anita Schöbel. To wait or not to wait? the bicriteria delay management problem in public transportation. *Transportation Science*, 41(4):527–538. 2007.

Jan-Willem Goossens, Stan van Hoesel, and Leo Kroon. On solving multi-type railway line planning problems. *European Journal of Operational Research*, 168(2):403–424. 2006.

Stefan Irnich. Resource extension functions: Properties, inversion, and generalization to segments. *OR Spectrum*, 30(1):113–148. 2008.

Julie Jespersen-Groth, Daniel Potthoff, Jens Clausen, Dennis Huisman, Leo Kroon, Gábor Maróti, and Morten Nyhave Nielsen. *Disruption management in passenger railway transportation*. Springer. 2009.

Niklas Kohl and Stefan E Karisch. Airline crew rostering: Problem types, modeling, and optimization. *Annals of Operations Research*, 127(1-4):223–257. 2004.

Leo G Kroon, Ramon M Lentink, and Alexander Schrijver. Shunting of passenger train units: an integrated approach. *Transportation Science*, 42(4):436–449. 2008.

Eugene L Lawler and David E Wood. Branch-and-bound methods: A survey. *Operations research*, 14(4):699–719. 1966.

Marco E Lübbecke. Column generation. *Wiley Encyclopedia of Operations Research and Management Science*. 2010.

Richard M Lusby, Jesper Larsen, Matthias Ehrgott, and David Ryan. Railway track allocation: models and methods. *OR spectrum*, 33(4):843–883. 2011.

Thomas L Magnanti and Richard T Wong. Network design and transportation planning: Models and algorithms. *Transportation science*, 18(1):1–55. 1984.

Metlink. Wairarapa bus & train timetable. URL `https://www.metlink.org.nz/assets/New-PDF-timetables/Train/WRC-Wairarapa-July2014-web.pdf`. [Online; accessed April 1, 2016]. 2014.

MTR. MTR system map. URL `https://www.mtr.com.hk/archive/ch/services/routemap.pdf`. [Online; accessed April 1, 2016]. 2016.

Lars Kjær Nielsen, Leo Kroon, and Gábor Maróti. A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, 220(2):496–509. 2012.

Elias Oliveira and Barbara M Smith. A job-shop scheduling model for the single-track railway scheduling problem. *RESEARCH REPORT SERIES-UNIVERSITY OF LEEDS SCHOOL OF COMPUTER STUDIES LU SCS RR*, (21). 2000.

Marc Pirlot. General local search methods. *European journal of operational research*, 92(3):493–511. 1996.

CB Quak. Bus line planning. *Master's thesis, TU Delft*. 2003.

François Ramond and Marcos Nicolas. *Trains don't vanish! ROADEF EURO 2014 Challenge Problem Description*. SNCF - Innovation & Research Department. 2014.

Natalia J Rezanova and David M Ryan. The train driver recovery problem—a set partitioning based model and solution method. *Computers & Operations Research*, 37(5):845–856. 2010.

Anita Schöbel. Line planning in public transportation: models and methods. *OR spectrum*, 34(3):491–510. 2012.

Peter Sels, Thijs Dewilde, Dirk Cattrysse, and Pieter Vansteenwegen. Reducing the passenger travel time in practice by the automated construction of a robust railway timetable. *Transportation Research Part B: Methodological*, 84:124–156. 2016.

Paolo Serafini and Walter Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581. 1989.

Lionel Adrian Silman, Zeev Barzily, and Ury Passy. Planning the route system for urban buses. *Computers & operations research*, 1(2):201–211. 1974.

Michael Sipser. *Introduction to the Theory of Computation*, volume 2. Thomson Course Technology Boston. 2006.

Pieter Vansteenwegen and Dirk Van Oudheusden. Developing railway timetables which guarantee a better service. *European Journal of Operational Research*, 173(1):337–350. 2006.

Cameron G Walker, Jody N Snowdon, and David M Ryan. Simultaneous disruption recovery of a train timetable and crew roster in real time. *Computers & Operations Research*, 32(8):2077–2094. 2005.

Laurence A Wolsey. *Integer programming*, volume 42. Wiley New York. 1998.

Part II

Research Work

Chapter 6

A Survey on Robustness in Railway Operations

With Richard M Lusby [1]
and Jesper Larsen. [2]
Submitted to International Transactions in Operational Research.

[1]Technical University of Denmark. rmlu@dtu.dk
[2]Technical University of Denmark. jesla@dtu.dk

6.1 Introduction

Operations Research (OR) plays a large and increasing role in the planning and execution of railway operations. As methods and approaches improve, and as rail utilization increases, it is increasingly important that solutions are not only of good quality in the normal case, but perform well when encountering unexpected situations or true realizations of estimated parameters. OR can provide the tools to both assess and quantify planning problem solutions under uncertainty, and to find solutions that perform well despite uncertainty. Such well behaved plans may be considered “robust”, and the quantification of performance may be a measure of robustness.

However given the large scope of the different problems in railway planning and operations, there are many different interpretations and definitions of robustness. Some railway planning is carried out with a view to the long term rather than immediate operations, and consequently any robustness considerations would be different to robustness considerations in planning the immediate operations. Within the scope of rail, however, we aim to examine the different approaches to robustness and in particular consider the degree to which approaches to robustness in rail can be unified by their common application, despite being different in scope and methodology. Here, we primarily consider passenger railway but in some specific cases refer to relevant or related work in similar application areas of freight railway or airline operations.

6.1.1 Robustness

Robustness is a concept that exists in many fields with similar general interpretations, but at a detailed level robustness may be significantly different. Certainly, in cases where robustness is defined by some application-specific metric the definition has little use outside the application area. However, generally, robustness refers to how a system or plan behaves in the presence of uncertainty. This can for example be that some plan is created using estimates of parameters, but operated with a realization of those parameters that can differ from the estimates. A plan that is well behaved under a wide range of realizations of parameters would be considered robust, whereas a plan that behaves very differently or even fails under different realizations would be less robust.

In the field of optimization, robustness has been considered in different ways. A common framework is termed *Robust Optimization*, which is surveyed by Ben-Tal and Nemirovski [2002]. This is strictly and clearly related to the conceptual idea that robustness is expressed in the context of an uncertainty of parameters or data for a problem. The following restrictions (among other) are imposed:

- The data are uncertain/inexact;
- The constraints must remain feasible for all meaningful realizations of the data.

For a Linear Programme (LP), with the nominal (non-robust) definition $\min_x\{c^T x : Ax \leq b\}$, a *robust optimization* version would permit that the values of A, b and c are not precisely known but lie in some *uncertainty set*. If that set is expressible with linear constraints, then the resulting optimization problem requiring a solution feasible for *any* realization of A, b, c, is also an LP. However, a potential problem with such modelling is that solutions can be somewhat pessimistic, by ensuring feasibility for any unlikely realization of uncertain parameters. Also, specifying the uncertainty set via linear constraints relating different uncertain parameters to each other may be difficult due to a lack of such data, whereas a lack of sufficient relationships between different uncertain parameters may result in solutions that are “robust” by being feasible in the case of particularly problematic but, in reality, impossible realizations of combinations of parameters.

Fischetti and Monaci [2009] defines the widely used “light robustness” framework. In short, light robustness is a maximization problem of the weighted slack of constraints, with a constraint on the nominal objective to be within some bound of the nominal optimal value, and where the slacks are bounded above by a calculated worst case for each constraint given the optimal nominal solution. The framework is related to “normal” robust optimization where there is uncertainty in constraint matrix $\boldsymbol{A}$; say each element has a nominal value a_{ij} but due to uncertainty, in a real instance will be in $[a_{ij}, a_{ij} + \hat{a}_{ij}]$. Then, for a given row in the $\boldsymbol{A}$ matrix, and with a bound on the number of elements that may be different from the nominal value (avoiding the over-conservatism of allowing any variation), and with a given solution, the worst case tightening of the constraint by uncertainty may be determined; this is then the maximum rewarded slack. The authors present railway timetabling as one application of light robustness.

Bertsimas and Sim [2004] defines a "price of robustness" probabilistically, related to traditional robust optimization (uncertainty in input parameters) but of a probabilistic nature. *... the trade-off between the probability of violation and the effect to the objective function of the nominal problem, ... is what we call the price of robustness*. In some row, suppose that the number of "modified" coefficients is bounded; then any solutions that would be feasible given that restriction are probabilistically protected from uncertainty dependent on the probability that at most that number of coefficients is modified. The price of robustness itself is the increase in objective value over that given by solving only the nominal problem. Here, there is the explicit recognition that requiring robustness "costs" something, when compared with a solution that is apparently feasible with no consideration for robustness. In a problem (such as rail) where the nominal problem defines the most likely realization of parameters, and variation is unlikely but potentially problematic, this cost is something that negatively affects the normal operation for a benefit in the unusual situation. Assessing the value of such solutions, and whether or not the "cost" is worthwhile, requires detailed probabilistic knowledge of the likelihood of such realizations, and some idea of the "cost" incurred when a nominal, non-robust, solution would encounter such realizations. On the other hand, when compared with the robust optimization described above, solutions are not necessarily so pessimistic.

Stochastic programming provides a different framework for finding solutions to optimization problems in the presence of uncertainty. Kall and Wallace [1994] provide a textbook on stochastic optimization. Here, probability distributions for unknown parameters can be used to find a plan or policy that has high expected objective value, and is feasible for all realizations. Commonly, stochastic programming defines a problem in *stages*, in which decisions may be made as uncertainty is revealed. If possible realizations are discrete in nature, this results in a tree structure of decision making with assigned possibilities. This tree may be fully enumerated, or, for tractability reasons, sampled as a limited set of discrete scenarios (see for example Kleywegt et al. [2002]). Rather than a full, fixed plan for operation, a solution is an initial plan dependent only on known problem parameters, and many contingency plans to be implemented as knowledge is revealed.

As a short comparison, generally robust optimization is a worst-case planning method ensuring feasibility in all circumstances. As an advantage it does not require probabilistic information, which may be difficult to capture, with the associated risk that created plans are robust against extremely unlikely circumstances. If plans must be kept as originally specified then this guarantee of feasibility is more reasonable than in problems where reactive modifications are possible during operation. The concept of light robustness formulates robustness as a maximization of the level of “protection” in changes of problem parameters. This then permits less pessimistic solutions that robust optimization, although again not including probabilistic information. Stochastic optimization, in contrast, is probabilistic in nature, which has the potential downside of requiring the gathering or estimation of many parameters, generally related to a discrete number of disruption scenarios. Stochastic optimization can be a more natural modelling method if changes are to be made to plans as uncertainty is revealed, and a fixed unchanged plan is not a requirement.

6.1.2 Railway operations

Passenger railway is a system of public transportation providing rail connections between stations. It includes a range of operations such as commuter trains, metro systems, light rail (trams), and regional and intercity trains. Passenger railway interacts with other public transport systems such as buses and ferries. Passenger railway decisions take into account other modes of transport such as freight railway, car traffic and buses, due to both shared infrastructure (e.g. freight rail and passenger rail on shared tracks; light rail, cars and buses on roadways), and also to passengers moving between modes.

The planning of railway operations differs between countries but in general there are several stages of planning from the most long term decision making to the day-to-day operations, and these stages are planned relatively independently and in sequence rather than as a single unified planning problem (though there may be an iterative process of feedback between stages). The size and complexity of each individual problem alone tends to mean that only a sequential approach can be applied in practice. A typical sequence of planning problems may be:

- **Network design**
  Determining the topological structure of the rail network; station locations and capacities; see Magnanti and Wong [1984] for an overview of network design in general transportation.
- **Line planning**

Determining the subsets of stations and a route between them that constitute train lines; deciding on frequencies, speeds, and rail stock types for each line. See Schöbel [2012] for a survey on line planning in public transport.

- **Timetabling**
  The determination of exact times for events that should occur for rail units such as driving between stations, the dwell times at platforms, and which specific station infrastructure is used. A survey of railway track allocation, which includes timetabling, is given by Lusby et al. [2011].

- **Rolling stock routing**
  The problem of allocating rolling stock units to the timetabled trips in circulations, considering compositions and depot parking. An example of a nominal model for the problem is given by Fioole et al. [2006].

- **Crew scheduling**
  The problem of creating rosters of duties to be performed by a single crew person over a time horizon. There are similar problems in airline crew scheduling (which may be termed pairing), and likely many similar problems in other application areas. A following problem is assigning these rosters to specific crew members. Caprara et al. [1998] describe modelling the problem, with an application to Italian railways. Kohl and Karisch [2004] survey the literature on similar and well-studied airline crew rostering problems.

These stages may be further divided up, as for example crew scheduling may consist of two steps: creating anonymous crew schedules followed by crew rostering. Or stages may be combined, such as integrated line planning and timetabling (e.g. Schöbel [2015]). However there are clear differences and distinctions between the stages; network design for example is rarely changed, and decisions must stay valid for potentially many years, whereas crew rosters may be different every day. See Figure 6.1 for a suggested overview of how the different problems in rail planning may be carried out, indicating some estimate of the time they are considered before the day-of-operation (but not to scale). This sequence is commonly seen in the rail industry, and figures very similar to Figure 6.1 appear in other work (e.g. Liebchen and Möhring [2007]; Lusby et al. [2011]). We also indicate recovery but there are in fact several problems considered in the present, during operations, related to dispatching and managing delay and disruptions. However we do not focus on these present operational problems here.

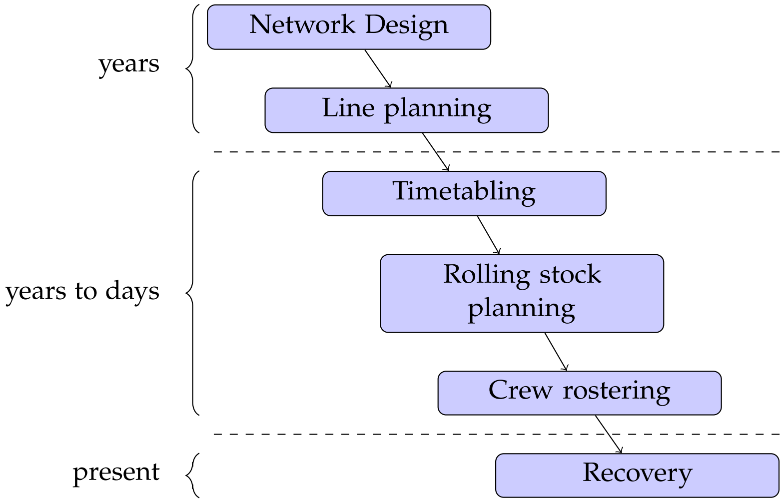


Figure 6.1: Overview of the possible steps of rail planning, indicating their relative time horizons.

Figure 6.2 (reproduced from Salido et al. [2012]) shows different considerations for a railway plan, and the trade-offs operators must generally make in creating a plan. High robustness can preclude (being close to) optimality, having high capacity utilization, and having a heterogeneous plan. In the general context of rail operations, capacity and heterogeneity do not necessarily have an obvious interpretation but capacity can be interpreted in relation to usage of infrastructure up to some maximum (and likely most fragile) level, and heterogeneity may be interpreted as a measure of the diversity of variety of rolling stock unit, or the variety in running times of trains on the same track, or the variety in the types of planned crew schedules. Low heterogeneity has robustness implications for both small disruptions (e.g. dissimilar running times lead to delay propagation) and during large disruption scenarios (e.g. dissimilar unit types may not be usable as replacements). Other factors may also be included; for example speed (in Salido et al. [2008]) may be reduced in a network, which also affects capacity and optimality with implications for robustness.

In the context of rail, robustness is a concept that is intuitively “obvious” or understandable, but there is no single clear definition. Further, any such proposed definition could not capture all elements of the intuitive “robustness” as understood by different stakeholders in public transport. There are in fact

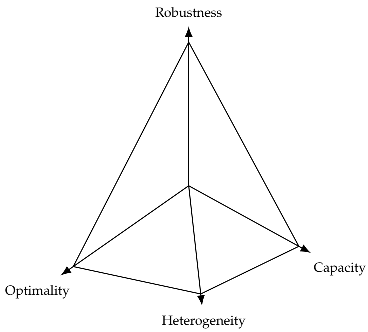


Figure 6.2: Overview of the possible steps of rail planning, indicating their relative time horizons.

many differing definitions of robustness in the literature on rail planning and operations, and indeed in other areas, but the core concepts are similar. In rail, as in most real-world problems, there are inherent uncertainties in aspects of the problem due either to inaccurate data, unpredictable occurrences, or stochastic processes.

Specifically referring to *passenger* robustness, De-Los-Santos et al. [2012] argue that generally agreed measurements of robustness do not exist. A passenger-robust timetable, the authors state, is one in which the travel time for passengers is not excessively increased when a particular link fails, and they define passenger-robust measurements indices. These are:

- the ratio of best-case travel time and the worst case (failure) for that connection;
- the ratio of best-case travel time and average case (failure) for that connection.

These can be calculated in the *with bridging* and *without bridging* cases, where bridging refers to the deployment of alternative routes by the operator (e.g. buses) rather than relying on passengers to re-route themselves or wait for the disruption to end. Dewilde [2014] investigates a definition of the robustness of a timetable, stating:

> A railway system that is robust against the daily occurring,

small disturbances minimizes the real weighted travel time of the passengers.

Dewilde [2014] considers the robustness of station areas. The author argues that as the purpose of rail networks is to serve passengers, a definition of robustness should (at least) consider passengers, while also arguing that there is no common measure of robustness in the literature. The weighted travel time assessment for passengers mentioned above, when the system is subjected to small disturbances, is argued to be the most relevant to users as it is a passenger-centric measure.

When planning railway operations, and when defining and quantifying robustness of operations, some authors take a point of view that indirectly benefits both the operator and the passenger (such as attempting to provide operations with little delay), while others explicitly take a viewpoint of the operator or of the passenger. While it is almost always true that it is the operator that makes decisions relating to robustness, the robustness itself can be viewed from the point of view of the operator or the point of view of the passenger, and the different perspectives do not always coincide. For example, a small delay to a train may not have a large affect on other operations, but the small delay may cause passengers to miss a transfer connection who may view the missed transfer as a symptom of poor robustness. Or, in contrast, to address an unexpected rolling stock problem an operator may be forced to use a different rolling stock unit than planned for, incurring additional operating cost, shunting and dead-head movements, but having no affect on passengers.

6.1.3 Disruption management

Operations are not always able to be carried out exactly as planned, perhaps due to some unforeseen occurrence, an accident, or closure of track. In these situations plans must be modified or *recovered* to create a plan that can be carried out. There may be several recovery stages undertaken in sequence following the original stages of operational planning itself; for example a temporary track closure due to maintenance may necessitate a new timetable, then requiring modifications or recreation of the rolling stock and crew plans. It may be that the changes are known long enough in advance that planning can be done as comprehensively as for the normal case, or only known during operations and therefore recovery must be done quickly. During the planning stages, contingency plans may be created for some possible disruptions, and recovery may consist of finding an appropriate contingency plan and adjusting it as necessary to fit the exact situation. Robustness implemented via recovery may be explicitly related to the previously defined planning problems above (i.e. network design, line planning, timetabling etc.) or may span several (e.g. both rolling stock and crew)

In a small scale delay scenario, there can be decisions related to train waiting such as deciding whether a non-delayed train wait so that planned coordination between the trains are maintained. This problem, for example, is studied by Schöbel [2007] for public transport vehicles (e.g. buses or trains).

In contrast, during operations there may be instances of track closure or blockage that require significant modifications to plans for a possibly unknown duration. As an example, Louwerse and Huisman [2014] study the problem of adjusting a passenger rail timetable

6.2 Overview of document

In the remainder of this document we consider robustness in different planning problems of railway described above. Robustness in network design and line planning are considered together in a single section. In railway timetabling, we consider the defining of robustness and optimization with robustness separately. Finally, robustness of rolling stock and crew planning are each considered individually.

6.3 Robustness in network design and line planning

In network design or line planning there has been little research into robustness, but some authors have attempted to define or incorporate robustness into line plans or even network design. The lack of a timetable at these stages make robustness measures related to exact passenger or train timings difficult to consider.

Goerigk et al. [2013a] investigates line plans or line "concepts", and their affect on resulting timetable quality and robustness. They question not whether the line plan *itself* is robust, but whether a resulting timetable for the line plan is or is not robust. The authors use two robustness measures: the relative increase in travel time for passengers under a disruption scenario, and the proportion of connections that would be missed in a disruption scenario. Their assessment is made via simulation, with a tool that both creates a timetable for a given line plan and provides the delay assessment. The authors examine two German intercity rail problems, with some provided and some estimated data and parameters, and the authors first find a line plan and then a timetable in an iterative way. Four different objectives or approaches for the line planning problem are used. Finally, the robustness of the resulting timetable is assessed. One conclusion is that when using an objective focused on the passenger, line plans offer nominally good solutions for passengers at the expense of robustness. One justification is that a good passenger solution can be considered one which provides many direct connections between stations, which is achieved with a variety of lines and tight transfers. There is then the potential for more missed transfers. In contrast, they observe that the (operator) cost oriented line plans have worse nominal passenger performance, but are more robust. The potential robustness of a resulting timetable is therefore very dependent on the goals of the line planning model.

Schöbel and Schwarze [2006] present a game-theory model for line planning, where individual lines seek to minimize their probability of delay, which depends on the presence and frequencies of other lines sharing infrastructure. The result is an overall line plan with an equally distributed probability of delay, which could be considered more robust than a plan where the delay probability is not equally distributed. As an exploratory numeric study the authors consider German intercity train lines, with some simplified details of the line planning problem, and show how their method is able to find equilibria for the problem.

Kontogiannis and Zaroliagis [2008] present a different view of robust line planning based on a network manager and independent line operators. The line operators compete to operate on shared infrastructure resources, managed by the single network manager. The incentive functions of the operators are the unknown parameters of the problem, and a robust line plan is one which maximises the aggregate utilities of the operators while being resilient to the unknown incentives of the line operators. The authors present a method for finding such robust solutions when certain conditions hold; however, the authors do not present a case study of the method.

Marín et al. [2009] consider robustness in both the line planning and network design problems. They consider robustness from two perspectives; the passenger and operator point of view, in the context of connection failures. Here, user robustness means that a failed connection should minimally affect travel time, and the passengers require geographically distinct routes but with similar travel times. Operator robustness, however, is measured in the cost of additional vehicles used to address such failures. The method developed is a heuristic that integrates both the network design and line planning problems. As a case study the authors consider the construction of a new high speed railway network in Andalucia (in southern Spain), and show how their method may be applied. They observe that both robustness measures (operator or passenger based) result in the same robust railway network but different line plans

Michaelis and Schöbel [2009] present a heuristic integrating line planning, timetabling and vehicle scheduling. The focus is not on robustness, but of including robustness implicitly with slack times for the timetable, and the authors state that such additional time can provide robustness. However the slack times are not distributed in some way guided by robustness concerns; it is simply implied that the presence of slack times provides robustness.

Bull et al. [2016] consider a line planning problem with a nominal passenger focus, but some mention is made of metrics that may be relevant for robustness of a subsequent timetable. The robustness of the line plan *itself*, however, is not defined or given mention; only the degree to which the line plan influences the potential robustness of a timetable. The case study considered is from the S-tog network Copenhagen.

There is not a clear picture of what exactly robustness in line planning or network design should be, or what uncertainty it is that plans should be robust against. This may be explained in part by the fact that robustness in line planning and network design have not been extensively studied. One natural concern with a line plan, from a robustness perspective, is whether or not a robust timetable can subsequently be created. This is also a consideration for optimality, disregarding robustness, and both inspire a more integrated approach. Railway timetabling, examined in the following section, is the most studied planning problem from a robustness point of view, and naturally robustness viewpoints in earlier problems look to the timetable to assess robustness. Nonetheless some approaches do consider the earlier problems' inherent robustness, such as the approach of Marín et al. [2009] addressing network failure by requiring geographically distinct connections. Elements of the network design could also influence robustness of subsequent plans, especially taking a detailed view of track layout, crossings, and locations of overtaking sidings. However, generally, from an optimization point of view, such approaches to network design are not taken.

6.4 Robustness in timetabling

In the area of railway planning, timetabling is the area most studied for implementing robustness. This is perhaps natural due to timetabling being placed in a moderate planning stage, being not too far removed from the exact operations (as line planning and network design may be), but having large scope for changes that are not already very constrained (as rolling stock and crew scheduling may be). However, the planned timetable affects the robustness of the subsequent rolling stock and crew planning.

Robustness in railway timetabling is summarised in Table 6.1 and Table 6.2, each corresponding to one of the two following sections and each introduced at the end of the corresponding section.

6.4.1 Defining and measuring robustness

Some work is concerned with the defining of robustness of a railway timetable, potentially for the assessment of different timetables, for the validation of methods producing robust timetables, or by defining metrics usable in models for creating railway timetables.

Many considerations of robustness relate to the distribution of buffer times. This is obviously more applicable to timetabling problems, where buffer times between trains can be included, and not directly to line planning or network design (as exact times or are generally not considered in those problems). Buffer times may also be included in rolling stock scheduling or crew scheduling though with a different interpretation: the buffer time between a particular unit finishing one service and beginning another. Buffer times are to mitigate the risk of the unknown precise “running time” of a train between two stations; running times are an input parameter but they are not necessarily known and are not fixed due to small delays. Buffer time supplementation is then an approach to create timetables that are still valid or operational for a range of uncertainties in input parameters. Up to some value over the anticipated running time, the timetable is still valid as the buffer time may be consumed. Smaller running times may also valid, if trains slow down or wait at stations. As pointed out by Dewilde [2014], much work on robustness in rail timetabling relates robustness to delay propagation, where a timetable that is unlikely to encounter propagating delay is more “robust”, and the buffer time is almost always related to the absorption of delay and subsequent avoidance of delay propagation.

Buffer time can be considered a directly applicable or measurable metric for robustness, for some small deviation from expected running times. However for larger deviations the buffer times alone are not necessarily an adequate measure, as the role of train dispatchers is neglected. For example, dispatchers may decide to change the order of trains leaving a station due to a delay, choosing to change the plan to avoid inducing delay for other trains. Also, additional buffer time does not necessarily mean additional robustness, as the introduced buffer time may be excessive or introduced in the wrong part of the timetable. For example buffer time at the end of the day may be less beneficial than buffer time early in the day, as early delays can have longer lasting effect than delays late in the day. Deciding on the amount of buffer time to use is a decision on the trade-off between (potentially) increased robustness and a decrease in utilization of the rail network and a decrease in passenger service. More buffer can for example require fewer trains operating and greater waiting time for passengers transferring between different trains.

As a support for determining the propagation of delay and addressing of it in planning, Flier et al. [2009] present methods for the derivation of delay dependencies in actual delay data. The authors formulate and identify possibly timetabled dependencies between trains where one train may wait for another, and then quantify the correlation between relevant event times for those trains using data from certain points of the Swiss SBB network. The authors show that correlation alone, without the correct underlying model of the relationship, can fail to identify delay relationships between trains, while with a model and data relationships can be discovered.

Carey [1999] discuss “heuristic” measures of the reliability of a (public transport) schedule which can be calculated in advance, and distinguishes heuristic measures from analytic and simulation methods. It is stated that, for measuring schedule robustness, knock-on delays (where one train being delayed causes another to be delayed) are most relevant. The measures are divided into those that are probabilistic in nature, deriving measures of knock-on delay based on probability density functions for departure times of trains in the absence of delay, and those that are not probabilistic and are largely related to headway. The headway, or the interval between subsequent trains, is used by many authors as a measure of robustness or a target for robustness in various ways in other measures of robustness. In this case, the measures are not validated with real data and instances.

Considering the interval between in a timetable events, Goverde [2007] propose methods for analysing properties of railway timetables, including timetable robustness. Their robustness measure is related to slack times between events; the slackness between two events can be quantified by identifying a *critical path* which is the path relating those events with the least slack time of all such event paths. They also propose a delay propagation model to quantify the effect of some initial delays, though these measures are not explicitly equated with robustness by the authors. As a case study the authors use the Dutch national railway timetable, including both passenger and freight rail, and show how delay propagation may be quantified and show the identification of critical timetable elements. The identification shows *critical cycles* of dependent events that begin at some event and return the same event in a later timetable period, that consume all of the time of the timetable cycling period. Without considering planned excess time within events, such cycles have no capacity to absorb additional delay meaning initial delay in such a cycle will continue into later timetable periods.

Hofman et al. [2006] present a simulation study on the robustness of commuter train timetables in Copenhagen, Denmark. There is no single precise definition of what robustness is, but the study considers the addition of slack time, and different recovery strategies for late trains. The authors observe an upper limit to the effectiveness of additional buffer time, above which there is low effect. Early turn-around is one recovery method the authors consider, which is a late train stopping at a non-terminal station and becoming a train of the same line in the opposite direction, and the strategy is shown to be just as effective as the more drastic cancellation of late trains for some cases.

Vromans et al. [2006] propose methods to increase the *reliability* of railway services. Their approach is to increase homogeneity of the timetable, by reducing instances of trains running at different speeds in track sections. The authors are interested in small disturbances, and in fact state that *no* timetable is robust enough to handle larger disturbances (without major online adjustments). As a case study the authors consider a single line in the Netherlands, and compare a real "heterogeneous" timetable, and a "homogenized" version, and see a decrease in delay measure by simulation, and state that the improvement is predictable with the measures Sum of Shortest Headway Reciprocals (SSHR) and Sum of Arrival Headway Reciprocals (SAHR). The authors define two measures of heterogeneity: SSHR and SAHR. For an ordering of trains running on a sequence of tracks, suppose train i precedes train $i+1$, and h_i^- is the minimal headway between trains i and $i+1$ in the track sequence (and final train N precedes train 1 in a cyclic manner).

$$\text{SSHR} = \sum_1^N \frac{1}{h_i^-} \tag{6.1}$$

Claiming that headway at arrival in a section is more significant that headway later in the section, then authors define h_i^A to be the headway between trains i and $i+1$ at arrival in the track sequence.

$$\text{SAHR} = \sum_1^N \frac{1}{h_i^A} \tag{6.2}$$

In the homogeneous case, where train speeds are all the same, both measures are equal and have minimal value when all trains are evenly spaced in the cyclic interval and has maximum value if they are "bunched".

Another train buffer related measure that appears in several works (such as Andersson et al. [2013], Fischetti et al. [2009], Kroon et al. [2007], Vromans [2005]) is Weighted Average Distance (WAD) of time supplements added to trips along a line, as a measure of how the supplements are distributed. The measure is intended to indicate the degree to which supplement or buffer is biased towards the start of the line, the end, or evenly distributed. If there are $N+1$ trips on a line, from trip $t=0$ to trip $t=N$, and each trip has supplement time s_t, then we may define WAD as follows:

$$\text{WAD} = \frac{1}{N} \frac{\sum_{i=0}^N i \cdot s_i}{\sum_{i=0}^N s_i} \tag{6.3}$$

Here, a WAD value of 0 would indicate that all buffer is allocated to trip $t=0$; a WAD value of 1 would indicate that all buffer is allocated to trip $t=N$, and a WAD of 0.5 would indicate that there is equal buffer in both the first half and second half of the trip.

Andersson et al. [2013] define a robust timetable as one which can maintain its planned operation, when subjected to some small delays, and also focus on the headways between trains. They define “critical points” in the timetable, that are a relationship between two trains at a location and time in the timetable, from which delay can propagate because the trains are running in the same direction on the same track, or where one is planned to overtake another. Critical points come from a timetable study by the same authors; Andersson et al. [2011]. The critical points are used to make timetable modifications such as headway increases. The authors measure the robustness of their original and modified timetables with other timetable-robustness measures from the literature. The comparative measures from literature are: WAD of runtime supplements from the start of a line (from Kroon et al. [2007], Fischetti et al. [2009], Vromans [2005]); SSHR (citing Salido et al. [2008], Vromans et al. [2006]). The measures of Andersson et al. [2013] are:

- Maximum runtime distance (MRD)
  Defined for parts of the network where the parts “are naturally bounded by the traffic structure”
- Total amount of runtime margin for each individual train (TAoRM)
- Robustness in critical points (RCP)
  A quantification of flexibility at the identified critical points, calculated for a pair of subsequent trains as the runtime margin between trains, the runtime margin for the earlier train to be earlier at the point, and for the later train to exit later from the point.

As a case study, the authors apply their methods to part of the Swedish mainline (a 200km double track in southern Sweden, between Malmö and Alvesta), and demonstrates the identification of potential modifications to increase robustness, showing how the different robustness measures may be interpreted in practice.

Landex and Jensen [2013] develop measures for analysing rail operations at stations, including two methods for calculating the complexity of the timetable at a station which the authors relate to the robustness of the operation. The authors suggest two methods that they specifically relate to robustness. The first is buffer time threshold, defined as a ratio of high risk conflicts, where buffer times between train are below some given threshold, to the total number of such potential conflicts. The second is and a complexity measure based on a probabilistic determination of trains delaying other trains. As an example application the authors apply the measures to Skanderborg station in Denmark, with a real timetable instance and two instances derived from the real timetable to achieve goals related to buffer times. The authors find that the modified timetable where small buffers are increased has the lowest timetable complexity (and best robustness), as measured by the delay probability, and also deduce that that particular measure is the best for timetable comparison while noting that an analysis of more stations should be done for a better picture of overall timetable complexity.

Salido et al. [2008] and Salido et al. [2012] develop methods for comparing the robustness of timetables, where robustness is described as being necessary *to absorb short disruptions*. The application is to single railway lines, and the authors derive both a simulation-based approach and an analytic approach. The quantification the authors present is of total delay encountered by other trains, given some initial input delay, and for both the case of homogeneous trains and heterogeneous trains the authors see agreement in total delay between the simulation and analytic approaches. The analytic methods consist of calculating the total delay of all trains for some initial delay in a corridor based on the absorption of the delay through headway, its propagation to subsequent trains, and by considering the possibility of overtaking. A timetable is said to be more robust than another if its total delay for the same initial input delay is lower. The analytic methods have the advantage of being computationally efficient compared to the simulation method and the authors suggest they are applicable in the generation of robust timetables.

Corman et al. [2014b] examine robustness of a train timetable as the performance of a timetable when faced with large scale disturbances (examples given are multiple train delays, speed restrictions, track blockages). The authors use different measures and compare different timetables under a range of disruption scenarios reporting things such as train delay and passenger travel time as a result of a disruption, taking those as indicating how robust the timetable was. The focus is not on the creation of timetables but rather of comparing timetables and responses to disruption with a robustness view. As a case study the authors use a section of the Dutch railway network, and compare the regular timetable and a shuttle timetable in which trains drive back and forward between pairs of major cities. For comparison the authors compare the disruption management methods of re-timing, where train order is maintained but times are altered; and rescheduling by changing train order using the method of Corman et al. [2014a]). The authors show results for both train delay and for passenger delay, showing the potentially conflicting experience of robustness passengers and the operator may have.

Takeuchi and Tomii [2006] (and Takeuchi et al. [2007]) define a robustness index for train timetables based on passenger disutility. The authors claim that any robustness measure should be defined from the passengers' perspective, not from the operator's perspective. The authors define passenger disutility as a weighted sum of congestion discomfort, number of transfers, waiting time at stations, and boarding time into trains (dependent on the number of other passengers boarding). The robustness index is the expected *increase* in this passenger disutility, given a sampling of delay scenarios, an estimate of the resultant train operations, and passenger re-routing given the new train operation. The authors give experimental results for an unspecified real urban train line, and compare the actual train timetable and a proposed modified timetable by re-allocating running time supplements. The authors show that their robustness index indicates better performance for the modified timetable. The method, however, is for the quantification of the robustness index and not for the modification itself.

Dewilde et al. [2011] present a claimed all-embracing, generic definition of the robustness of a railway timetable:

> A railway timetable that is robust against small delays minimizes the real total travel time of passengers, in case of small delays. Limited knock-on delays and a short settling time are necessary but not sufficient conditions for a timetable to be robust. Furthermore, different weights can be assigned to different kinds of travel time prolongation.

Here the authors have a clear passenger focus; their robustness measure only relates to passengers; not to the increase in operator cost in the case of delays or in the nominal case to guarantee the robustness. The authors apply their method to method to the 2010 Belgian timetable, using both real and artificially modified timetables with increased running time supplement. The authors show that, while the timetables with additional running time supplement have fewer delays, they have worse robustness as calculated using their definition. The measure is also strictly related to small delays. In fact, many measures and metrics encountered in this section are implicitly or explicitly related to small delays only. However in the following section, applying different methods, more authors seek robustness as large-scale disruption robustness.

Table 6.1 summarises work defining or measuring railway timetable robustness, classifying whether they have a passenger or operator focus and noting whether there is a clear application to some real timetable problem instance.

Table 6.1: Summary of literature measuring or defining timetable robustness

Reference	Focus	Description	Application
Andersson et al. [2013]	Operator	Metric based assessment based on critical points from which delay can propagate.	Swedish single line
Corman et al. [2014b]	Mixed	Analysis of train and passenger delay in the case of major disruption.	Southern Dutch network
Dewilde et al. [2011]	Passenger	Define robustness as realized passenger travel time, encountering small delays.	Belgian timetables
Goverde [2007]	Operator	Timetable assessment based on identification of critical paths and their relation to delay propagation.	Single dutch line
Hofman et al. [2006]	Operator	Assessment of timetables and recovery strategies; considers buffer times	Copenhagen commuter timetable
Landex and Jensen [2013]	Operator	Station-centric robustness analysis; buffer time and probabilistic delay propagation measures.	Danish station
Salido et al. [2012]	Operator	Analysis based on delay propagation.	-
Takeuchi and Tomii [2006]	Passenger	Method for comparing timetable based on delay-induced passenger travel time.	Real, unspecified
Vromans et al. [2006]	Operator	Metric-based assessment, introducing two measures of heterogeneity and spreading of trains.	Dutch single line

6.4.2 Optimization with robustness

Cacchiani and Toth [2012] surveys the literature on the rail timetabling problem, in its regular and robust forms. The following approaches to robustness are identified:

- stochastic programming [Kroon et al., 2008, Fischetti et al., 2009].
- light robustness [Fischetti and Monaci, 2009, Fischetti et al., 2009].
- recoverable robustness [Liebchen et al., 2009, Cicerone et al., 2009, D'Angelo et al., 2009, Cicerone et al., 2008a].
- delay management [Liebchen et al., 2010, Cicerone et al., 2007b]

Fischetti et al. [2009] are concerned with the modification of railway timetables to introduce robustness, where robustness is related to the absorption of minor delays, and explicitly *not* delay requiring major adjustment. The robustness is validated with a separate validator that tests a solution on a number of delay scenarios, permitting limited adjustment but again stating that robustness is not related to major disruptions. Four methods are proposed: one effective but slow stochastic programming formulation; two "slim" stochastic programming methods, and a method based on light robustness. The full stochastic programming formulation minimizes delay time, similar to the validator, whereas the two slim formulations minimize weighted unabsorbed delay time (differing only in weighting). Instances are single-line problems and come from the Italian railway operator. As one measure, the authors use WAD and are able to draw a relationship between WAD and their measured robustness. The authors show that with a weighting putting more importance on earlier events than later events, both the light robustness formulation and the slim stochastic programming formulation perform well.

Cacchiani et al. [2012a] present a lagrangian relaxation heuristic for robustness problems, applying the method to the train timetabling problem. The method finds solutions that are approximately pareto optimal considering nominal efficiency and an estimate of robustness (against small delay). The solutions are validated using the tool of Fischetti et al. [2009], reporting average total delay. In the method itself, however, robustness is estimated as a dynamically weighted profit for buffer time for certain types of train event. The method is applied to corridor-focused Italian Railway instances, and also compared with results of Fischetti et al. [2009] showing better results in many cases found with less computation time.

Recoverable robustness (Liebchen et al. [2009]) is a new approach to robustness that, briefly, considers a nominal problem with scenarios and recovery possibilities; a solution to the nominal problem is *recoverable robust* if for any scenario, it can be recovered to feasibility using the given recovery possibilities.

The authors give two rail applications as examples where recoverable robustness may be applicable. The authors contrast recoverable robustness with both robust optimization and stochastic optimization. Robust optimization can result in unnecessarily pessimistic solutions, because solutions must accommodate for every uncertainty simultaneously and without change. Stochastic programming instead permits some aspects of the solution to be fixed and solves the uncertain ones later in a second step; a 2-stage expansion in the scenarios. However, the authors claim that the complexity of their presented train timetabling problem results in problems that are too large and complex to approach with stochastic programming. Recoverable robustness, however, is claimed to combine "the flexibility of stochastic programming with the performance guarantee and the compactness of models found in robust optimization".

Cicerone et al. [2008a] expand the concept of recoverable robustness, considering the recoverability of a timetable (or more generally some schedule) in the face of several recovery steps. This is applied to the train timetabling and delay management problem. Cicerone et al. [2009] consider the timetabling problem using the concept of recoverable robustness and quantifying the "price of robustness". Several algorithms are proposed, which improve upon similar work and algorithms by Cicerone et al. [2008b]. The authors are explicit in stating the relationship between robust timetabling and delay management; Cicerone et al. [2008b] state that:

> The field of robust optimization is still in its beginning. This work can be considered as a first step in the study of recoverable robust timetables.

The authors do not make an experimental study applying their methods to a real world problem, but note the value in doing so as future work. Outside the scope of passenger rail, Cicerone et al. [2007a] apply similar concepts of recoverable robustness to the shunting of trains.

Goerigk and Schoebel [2014] define robustness in terms of a "recovery to optimality"; a solution is robust if, faced with the realization of uncertain parameters, can be recovered to an optimal solution with low recovery cost. Here, then, robustness is related to large scale disruption requiring intervention and recovery; not solutions that are still valid in the presence of small scale disruption. The authors state that their intention is to create a solution that is not strictly robust but rather one that can be recovered easily to an optimal, or good quality, solution for any disruption scenario. As an example the authors present an aperiodic timetabling application (an LP), with a specific created instance based on the German intercity rail system, and compare several different timetables:

1. A nominal (non-robust) solution
2. A strictly robust (robust optimization) solution
3. A lightly robust solution
4. A recoverable robust solution
5. A recoverability to optimality solution
6. A uniformly buffered solution (with 6% extra buffer time applied everywhere)

Testing the solutions using two different samplings of uncertainty, they show and conclude that their approach can find solutions that have good nominal objective values, and perform particularly well for a worst case recovery to optimality, and not as well simply recovering to feasibility. The authors suggest that their approach is effective in cases where a large-scale disruption is expected to occur, with the exact details unknown but with set of possible scenarios determined, such as knowledge that one of several disruptive construction plans will go ahead. Their approach leads to a solution that may be recovered or modified to a good solution in all cases, with low cost measured as difference from the nominal plan. However the authors note that generally, periodic timetables are sought and nominal periodic timetabling is (or can be) modelled as a Mixed Integer Programme (MIP), and not as a LP problem. They conclude that their method would require adaption.

Schöbel and Kratz [2009] use a bi-objective approach to robust aperiodic timetabling, using as one objective the nominal timetable quality, and as the other a robustness measure. The authors give three possible measures of robustness: the largest initial delay for which no passenger misses a transfer; the maximum number of passengers who could miss a transfer if all delays are bounded by some given value; the maximum accumulated passenger delay that could occur if all delays are bounded by some given value. The first would be maximized, and the second and third minimized. These, then, are examples of passenger-centric robustness approaches. The authors show how pareto-optimal solutions may be found for the problem by a formulation as a timetable problem with a robustness level parameter, and other objective is total passenger travel time. Solutions to this problem, given some robustness level, are pareto optimal for the bi-objective formulation considering passenger travel time and robustness as objectives. The authors do not show results for a particular case study but rather consider formulations, and propose the idea of using a bi-objective for robustness in general.

Liebchen et al. [2010] introduce robustness into timetable planning by finding a periodic timetable with high delay resistance. The authors describe their robustness approach as an extension of light robustness. For the construction of timetables, the authors use a weighted sum of passenger travel time and expected deviation (delay) from the published time. The expected delay is computed in a simplified way in the constructive model using scenarios with given probability and input delays, and subsequent passenger delay is calculated using a no-wait policy where non-delayed trains all operate on time (which may be infeasible in practice). However, results are validated using a more detailed delay-management model. Delay in the authors context is many, small delays. Their solutions are assessed with (multiple) input dealy of at most 20 minutes, while in the creation of timetables delays of up to 40 minutes are considered. As an example, the authors apply their method to passenger railway lines in the Harz region of northern Germany and show timetables can be created that substantially reduce expected passenger delay with only minimal increase in nominal cost.

Vansteenwegen and Van Oudheusden [2006] use a passenger-centric approach to the cost of a timetable, and use a method to modify times of a given timetable to reduce expected passenger wait times when trains are delayed. The authors state that a robust timetable is one that performs well in non-ideal circumstances, and therefore their planning by including expected delay can be considered more robust than one that is created without considering the delay. The authors apply the method to passenger trains in a small part of the Belgian railway network, and their results show an improvement in overall waiting time although they also see an increase in passengers who miss their connection. The method uses arriving train delay probability, justifying their choice of distribution with delay data from the rail operator. The methodology is improved by Vansteenwegen and Van Oudheusden [2007] and applied to the full Belgian network. Their results show that the timetable created with their LP method outperforms the original timetable for several performance metrics, including passengers missing connections, and also show a reduction in long waiting times even in the no-delay case.

Kroon et al. [2008] consider the railway timetabling problem with stochastic disturbances, and equate robustness with the ability of the timetable to cope with those disturbances. They address the problem by a (re-)allocation of running time supplements, planning for certain events to take longer than their minimum required time. They state that their approach improves robustness to small delays only, and furthermore state that most robustness work in literature is focused on such small delay resilience. The work is an allocation of buffer time. Given a timetable, buffer time can best be allocated to achieve robustness to delay. The model itself is a two-stage stochastic optimization model, consisting of the creation and assessment of the timetable (considering average weighted delay of trains), by a sampling of possible delay scenarios. As a case study the authors apply their method to the train timetable for a northern section of the Dutch railway network, and compare the original to a timetable they create where running times of trains may change by only one minute, and another where running times may change by any (feasible) amount, and see improvement in (train) delay and punctuality over the original timetable even with only the small modification permitted. The authors sample input delay from a distribution that permits up to 10 minutes of delay.

Also taking a stochastic optimization approach to railway timetabling for small disturbances, Kroon et al. [2007] study an optimal placement of running time supplements along a train line. One measure used is WAD, as defined above, and the authors observe that with different amounts of total slack allocated optimally, the distribution favours early buffer supplement (low WAD) when less total buffer supplement is available.

Burggraeve et al. [2016] consider an integrated approach to robustness of the line planning and timetabling problems, in a heuristic framework. The S-tog network of Copenhagen is again studied, and the line plans are modified by stopping pattern to alter the running times of lines using information from a timetable model that aims to build a robust timetable for the line plan. The overall aim is to create a robust timetable but, where this is not possible due to the line plan, modify the line plan to further facilitate timetable robustness. The claimed robustness improvement is related to buffer time in the created timetable, where, in creating the timetable, the goal is to maximize the smallest buffer time between trains.

Finally, considering the timetable, but entirely from a passenger's perspective, Goerigk et al. [2013b] considers the concept of robustness in the *timetable information* problem, which is the problem of finding a passenger path, given a timetable, which minimises travel time and/or the number of transfers. A robust form of the problem is one which guarantees an unchanged, or minimally changed travel time, or likely unchanged travel time, given some set of disruption scenarios. The authors give a strictly robust formulation, where a path maintains validity in all delay scenarios, and a light robustness formulation where the path length is bounded to be close to the optimal path length, but additionally should have as few unreliable transfers as possible. Data instances are from the German rail schedule, and comprise either the full network or only high speed trains. One conclusion is that strict robustness is too costly to the passenger in terms of travel time to be used in practice, while in contrast the light robustness approach leads to solutions that are often as good in terms of robustness and suffer a much lower penalty in travel time.

Table 6.2 presents a summary of work incorporating robustness in railway timetables, classifying whether they have a passenger or operator focus and noting whether there is a clear application to some real timetable problem instance.

6.5 Robustness in rolling stock

Rolling stock planning is another area where robustness is studied and incorporated into railway planning.

Alfieri et al. [2006] mention robustness of railway rolling stock operation briefly; they relate a minimum turning time requirement to the robustness of a rolling stock plan, which comes with a cost. They also briefly mention the impact of required shunting on robustness, where required shunting in a plan has a risk of delay, and is therefore less robust than a plan with less required shunting. However the work is more generally focused on rolling stock circulation without a clear theme of robustness, and is applied to an intercity line in the Netherlands as a case study.

Abbink et al. [2004] present a model for the allocation of rolling stock units to timetabled trips, with an objective of minimising relative seat shortages. However they do indicate the variety (i.e. lack of homogeneity) of rolling stock unit types as being indicative of a lack of robustness. Via constraints,

Table 6.2: Summary of literature implementing robustness in timetable optimization

Reference	Focus	Description	Application
Cacchiani et al. [2012a]	Operator	Lagrangian heuristic for creating robust timetables for small delay with low cost to the nominal objective.	Italian "corridor" instances
Cicerone et al. [2009]	Mixed	Study of the timetabling problem in terms of recoverable robustness; theoretical.	-
Fischetti et al. [2009]	Operator	Modify timetables to be more robust to small delay; present methods based on stochastic programming and light robustness.	Italian single line instances
Goerigk and Schoebel [2014]	-	Finds aperiodic timetables that are recoverable to optimality in disruption scenarios; theoretical.	-
Kroon et al. [2008]	Operator	Stochastic-based modification to a timetable to improve expected punctuality in the face of input delay.	Northern Dutch network
Liebchen et al. [2010]	Passenger	Light robustness (inspired) method to create timetables resistant to delays	North German network
Schöbel and Kratz [2009]	Passenger	Bi-objective approach to robustness, with application to aperiodic timetabling; theoretical. Gives passenger-based nominal and robustness definitions.	-
Vansteenwegen and Van Oudheusden [2006]	Passenger	Modification to a timetable to reduce expected passenger delay.	Belgian network section
Vansteenwegen and Van Oudheusden [2007]	Passenger	Modification to a timetable to reduce expected passenger delay.	Belgian network
Burggraeve et al. [2016]	Operator	Heuristic integration of line planning and timetabling, maximizing minimum headway in the timetable by altering the line plan.	Copenhagen S-tog network

the number of types of units allocated to a train line is limited, arguing that such limits lead to improved robustness at some cost to the nominal objective, although this cost of robustness is not quantified. The authors apply their model to an instance of regional trains in the Netherlands.

Cacchiani et al. [2012b] present an optimization model based on recoverability when faced with different disruption scenarios (noting that in the nominal problem, robustness considerations may be included by a count of composition changes arguing that such events can lead to delay propagation). In their full formulation, robustness is considered by simultaneously solving the rolling stock circulation plan and a recovery plan (with an estimate of the recovery cost) for a number of disruptions They also consider the solutions afterwards on a larger range of scenarios. The authors consider the worst case recovery scenario, rather than an expected value and therefore do not require probabilities for scenarios. The authors apply their method to an intercity line in the Netherlands, and when compared to a nominal optimal solution, can find solutions that experience fewer cancelled trips and have lower shunting costs for recovery when assessed with many scenarios. Their Benders' decomposition method is solved in a heuristic way, and estimates the true recovery cost for scenarios. The authors observe that the estimate is generally accurate, although note some cases where it makes a significant underestimate.

Cadarso and Marín [2010] address the rolling stock routing problem for the rapid transit routing problem, and consider some robustness aspects. Robustness is included with two measures; the first is based on penalising propagated delay, which depends on expected arrival delay and assigned slack time to absorb the delay (using historic arrival delay distributions). The second is by penalising certain operations which require additional crew. The authors apply their method to two instances of the Madrid suburban rail network; one of a single line, and another of two related lines; and show a reduction in expected delay when compared with the nominal solution. Cadarso and Marín [2011] also consider the same two instances from the Madrid suburban rail network, considering the rolling stock assignment problem. Robustness here is said to be introduced by penalising solutions that require composition changes during rush hours, and empty train movements during rush hours, as both are claimed to risk leading to propagation of delay. The robust solutions presented do indeed reduce the number of empty movements at rush hour significantly, and the number of composition changes can be reduced, with a price for robustness paid for in increased operator cost but not in the service offered to the passenger. The same two instances from the Madrid suburban rail network are considered by Cadarso and Marín [2014], solving the rolling stock assignment and train routing problem for rapid transit simultaneously using a method based on a Benders' decomposition (solved heuristically). Robustness considerations are introduced by penalising risky empty train movements and shunting movements, and expected delay, in a weighted objective with operator costs. Two other robustness considerations are a per-passenger-in-excess penalty on trains as very full arrivals at stations can lead to congestion, and finally restricting the number of unit types assigned per line (similar to Abbink et al. [2004]). In the case study of the two Madrid instances, the authors show improved solution quality over the nominal solution, and also find better solutions with less cost for including robustness over a sequential approach solving assignment and routing separately.

6.5.1 Train routing

Here, we consider works concerned with the routing of train units in station areas. This could also be classified with the timetabling works, but there we have focused more in macroscopic timetabling whereas here, we discuss works taking a more fine-grained view where particular train unit information is relevant.

Caimi et al. [2005] consider the routing of trains through a station, where the timetable and station infrastructure layout is given but exact routings must be determined. The authors seek delay-resistant routings, measuring the maximum deviation from schedule an individual train may encounter without conflict if every other train is on time, and use a weighted sum over all trains as an objective to maximize in a local search heuristic method. The authors present results for application to Bern station in Switzerland and can find much improved solutions, as measured by their objective.

Caprara et al. [2010] also consider the train routing problem at a station, determining platforms and routes through the station infrastructure. The plans are assess using external delay and considering subsequent delay, when using different strategies to address the delay. The plans themselves, however, are created with one of two different ideas intended to introduce robustness. The first is to increase the robustness to small delay, by ensuring events have a large capacity to absorb delay. The second is a method of reserving both a platform and a backup platform for trains, as a contingency plan in the case of delay. In their Genova case study, this backup platform method is shown to be effective at reducing overall delay.

In an approach integrating routing and platforming with timetabling, Dewilde et al. [2013] consider the improvement of robustness at a single station. The authors argue that passenger service should be measured by realized total travel time, and that a robust railway system is one that minimizes real total travel time in the face of small, frequently occurring disturbances. The authors define a **weighted travel time extension** as

$$\frac{\text{realised passenger travel time} - \text{nominal travel time}}{\text{nominal travel time}}$$

This is for some system a measure of its robustness, and this may be compared to other systems by the percentage difference. They would say that system 1 is x-percent more robust than system 2 if the weighted travel time extensions of the systems compare in that way. In modelling the problem, the authors use a measure of the spreading of trains and assess their true defined robustness with a simulation method when a solution is found. The method itself consists of a routing module, a timetable module, and a platform assignment module in a heuristic framework. The authors study two station areas: Brussels (containing three closely linked stations), and Antwerp, both in Belgium, and in both cases show that they can find solutions that are 8% more robust than the original solutions.

Jin et al. [2014] present work on the *resilience* of a metro network to disruptions, analogous to many definitions of robustness. They introduce resilience by integrating the metro operation with another transport mode (buses), such that in the case of a metro disruption alternative capacity and routing may be provided by the unaffected alternate mode.

6.6 Robustness in crew

Crew rostering at railways can be an optimization research problem; for example Caprara et al. [1998] apply methods to the crew rostering problem with an application to Italian railways, and more recently Abbink et al. [2005] apply operations research to the crew scheduling problem at Netherlands Railways. However, robustness in crew scheduling has not been as thoroughly considered as it has been in other areas of railway. In contrast, in the more studied field of airline crew scheduling, robustness has been an area of study, such as the bicriteria approach of Ehrgott and Ryan [2002].

In the field of freight railway, Jütte et al. [2011] describe the modelling and implementation of a crew scheduling system for the DB Schenker German freight network. Robustness is not a focus of the work, but some robustness considerations are included. For example, buffer times are included when a crew person changes train, and the changes themselves are (optionally) penalised to avoid duties for a crew person that included multiple train changes. Another consideration is that employment rules dictate that a maximum of ten hours may be worked in a duty, and if during operation this would be exceeded (for example due to delay), then remaining tasks must be reallocated which is costly and can cause further delay. The authors consider restricting the maximum number of work hours in a duty to be less than ten hours to reduce the risk of reallocation being necessary, showing results in ten minute increments for maximum work of between nine hours and ten hours. Robustness is improved as measured by a lower risk of such disruption, based on historic delay data, with an increase in operator cost as the price for the robustness increases.

Potthoff et al. [2010] present a column generation algorithm for the rescheduling of crew in disruption cases with data from Netherlands Railways. The authors use a weighted objective with one consideration being robustness; they suggest penalising any change to the original schedule (as fewer drivers need to be informed of changes), and penalising any short transfers between consecutive tasks on different rolling stock units. However in the presented experiments the penalty for short transfers is set to zero. The authors show the results for their algorithm using ten instances and different problem parameters; however the exact affect of the robustness considerations are not exactly quantified or described.

6.7 Conclusion

Robustness has become an increasingly significant factor in both research and application for railway planning problems. There has been a clear focus on robustness of the timetable, but more authors are applying similar ideas to other areas of railway planning such as rolling stock planning. Robustness work also appears in railway crew planning, and in the related field of airline crew planning robustness features.

Some work has appeared considering the integration of different planning problems of railway, and consequently some work features robustness in integrated problems. For example the integration of network design and line planning has been studied with robustness considered, as have the line planning timetabling problems.

There have been several metrics and measures for robustness derived for railway systems, mostly but not exclusively focusing on the timetable. However it is not evident that a single metric will ever be derived that satisfies all interested parties, as different metrics focus on different stakeholders and some are even contradictory.

Without integrating planning problems, robustness planned in the individual planning steps may be lost in the overall plan. For example a rolling stock plan designed to have recoverable-robustness, where in large scale disruption scenarios the rolling stock plan can always be recovered to a good plan, may in fact lose its robustness if planned crew schedules can not feasibly cover the required recovery train movements. A robust line plan may not facilitate robust timetables to be planned; or a robust timetable may lead to non-robust rolling stock and crew plans.

Measures of robustness indeed generally focus on a particular part of the plan such as the timetable or rolling stock plan. They are robustness of the timetable, or robustness measures of rolling stock; not robustness measures of passenger railway. While a completely integrated planning approach may not be achievable, an integrated metric of robustness may be more feasibly derived and provide insight into the possible loss of robustness from non-integrated planning. We do not yet see the conceptualization of robustness of railway operations in a wholistic manner, but as metrics and concepts become more developed for the individual problems we may see more parallels between them.

We may expect to see more formulations that treat robustness as an additional objective function and therefore more multi-objective optimization approaches to planning with robustness. As robustness is generally not measured in units comparable to other costs, methods that provide several pareto-optimal solutions for manual planners to assess. Certainly this is an area attracting present research, and approximately half the works cited in this review are from the year 2010 or more recent.

6.8 Bibliography

Erwin Abbink, Matteo Fischetti, Leo Kroon, Gerrit Timmer, and Michiel Vromans. Reinventing crew scheduling at netherlands railways. *Interfaces*, 35(5):393–401. 2005.

Erwin Abbink, Bianca Van den Berg, Leo Kroon, and Marc Salomon. Allocation of railway rolling stock for passenger trains. *Transportation Science*, 38(1):33–41. 2004.

Arianna Alfieri, Rutger Groot, Leo Kroon, and Alexander Schrijver. Efficient circulation of railway rolling stock. *Transportation Science*, 40(3):378–391. 2006.

Emma Andersson, Anders Peterson, and Johanna Törnquist Krasemann. Robustness in swedish railway traffic timetables. In *4th International Seminar on Railway Operations Modelling and Analysis, Sapienza-University of Rome, February 16-18, 2011*. 2011.

Emma V. Andersson, Anders Peterson, and Johanna Törnquist Krasemann. Quantifying railway timetable robustness in critical points. *Journal of Rail Transport Planning and Management*, 3(3):95–110. 2013.

Aharon Ben-Tal and Arkadi Nemirovski. Robust optimization–methodology and applications. *Mathematical Programming*, 92(3):453–480. 2002.

Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations research*, 52(1):35–53. 2004.

Simon H Bull, Natalia J Rezanova, Richard M Lusby, and Jesper Larsen. An optimization based method for line planning to minimize travel time. Technical report, Technical University of Denmark. 2016.

Sofie Burggraeve, Simon H Bull, Peter Vansteenwegen, and Richard M Lusby. Integrating robust timetabling in line plan optimization for railway systems. Technical report, Tecnical University of Denmark. 2016.

Valentina Cacchiani, Alberto Caprara, and Matteo Fischetti. A lagrangian heuristic for robustness, with an application to train timetabling. *Transportation Science*, 46(1):124–133. 2012a.

Valentina Cacchiani, Alberto Caprara, Laura Galli, Leo Kroon, Gábor Maróti, and Paolo Toth. Railway rolling stock planning: Robustness against large disruptions. *Transportation Science*, 46(2):217–232. 2012b.

Valentina Cacchiani and Paolo Toth. Nominal and robust train timetabling problems. *European Journal of Operational Research*, 219(3):727–737. 2012.

Luis Cadarso and Ángel Marín. Robust routing of rapid transit rolling stock. *Public Transport*, 2(1-2):51–68. 2010.

Luis Cadarso and Ángel Marín. Robust rolling stock in rapid transit networks. *Computers & Operations Research*, 38(8):1131–1142. 2011.

Luis Cadarso and Ángel Marín. Improving robustness of rolling stock circulations in rapid transit networks. *Computers & Operations Research*, 51:146–159. 2014.

Gabrio Caimi, Dan Burkolter, and Thomas Herrmann. Finding delay-tolerant train routings through stations. In *Operations Research Proceedings 2004*, pages 136–143. Springer. 2005.

Alberto Caprara, Laura Galli, Leo Kroon, Gábor Maróti, and Paolo Toth. Robust train routing and online re-scheduling. *Openaccess Series in Informatics*, 14:24–33. 2010.

Alberto Caprara, Paolo Toth, Daniele Vigo, and Matteo Fischetti. Modeling and solving the crew rostering problem. *Operations research*, 46(6):820–830. 1998.

Malachy Carey. Ex ante heuristic measures of schedule reliability. *Transportation Research, Part B (Methodological)*, 33B(7):473–494. 1999.

Serafino Cicerone, Gianlorenzo D'Angelo, Gabriele Di Stefano, Daniele Frigioni, and Alfredo Navarra. Robust algorithms and price of robustness in shunting problems. In *ATMOS*, pages 175–190. Citeseer. 2007a.

Serafino Cicerone, Gabriele Di Stefano, Michael Schachtebeck, and Anita Schöbel. Dynamic algorithms for recoverable robustness problems. In *ATMOS*. 2008a.

Serafino Cicerone, Gianlorenzo D'Angelo, Gabriele Di Stefano, Daniele Frigioni, and Alfredo Navarra. On the interaction between robust timetable planning and delay management. In *2nd Annual International Conference on Combinatorial Optimization and Applications (COCOA'08)*. 2007b.

Serafino Cicerone, Gianlorenzo D'Angelo, Gabriele Di Stefano, Daniele Frigioni, and Alfredo Navarra. Delay management problem: Complexity results and robust algorithms. In *Combinatorial Optimization and Applications*, pages 458–468. Springer. 2008b.

Serafino Cicerone, Gianlorenzo D'Angelo, Gabriele Di Stefano, Daniele Frigioni, and Alfredo Navarra. Recoverable robust timetabling for single delay: Complexity and polynomial algorithms for special cases. *Journal of combinatorial optimization*, 18(3):229–257. 2009.

Francesco Corman, Andrea D'Ariano, Dario Pacciarelli, and Marco Pranzo. Dispatching and coordination in multi-area railway traffic management. *Computers & Operations Research*, 44:146–160. 2014a.

Francesco Corman, Andrea D'Ariano, and Ingo A Hansen. Evaluating disturbance robustness of railway schedules. *Journal of Intelligent Transportation Systems*, 18(1):106–120. 2014b.

Alicia De-Los-Santos, Gilbert Laporte, Juan A. Mesa, and Federico Perea. Evaluating passenger robustness in a rail transit network. *Transportation Research Part C*, 20(1):34–46. 2012.

Thijs Dewilde. *Improving the robustness of a railway system in large and complex station areas*. Ph.D. thesis, Arenberg doctoral school, Faculty of Engineering Science, KU Leuven. 2014.

Thijs Dewilde, Peter Sels, Dirk Cattrysse, and Pieter Vansteenwegen. Defining robustness of a railway timetable. In *25th Annual Conference of the Belgian Operations Research Society*, pages 108–109. 2011.

Thijs Dewilde, Peter Sels, Dirk Cattrysse, and Pieter Vansteenwegen. Robust railway station planning: An interaction between routing, timetabling and platforming. *Journal of Rail Transport Planning and Management*, 3(3):68–77. 2013.

Gianlorenzo D'Angelo, Gabriele Di Stefano, and Alfredo Navarra. Recoverable-robust timetables for trains on single-line corridors. In *Proceedings of the 3rd International Seminar on Railway Operations Modelling and Analysis (RailZurich)*. 2009.

Matthias Ehrgott and David M Ryan. Constructing robust crew schedules with bicriteria optimization. *Journal of Multi-Criteria Decision Analysis*, 11(3):139–150. 2002.

Pieter-Jan Fioole, Leo Kroon, Gábor Maróti, and Alexander Schrijver. A rolling stock circulation model for combining and splitting of passenger trains. *European Journal of Operational Research*, 174(2):1281–1297. 2006.

Matteo Fischetti and Michele Monaci. Light robustness. *Lecture Notes in Computer Science), Lect. Notes Comput. Sci*, 5868:61–84. 2009.

Matteo Fischetti, Domenico Salvagnin, and Arrigo Zanette. Fast approaches to improve the robustness of a railway timetable. *Transportation Science*, 43(3):321–335. 2009.

Holger Flier, Rati Gelashvili, Thomas Graffagnino, and Marc Nunkesser. Mining railway delay dependencies in large-scale real-world delay data. In *Robust and online large-scale optimization*, pages 354–368. Springer. 2009.

Marc Goerigk, Michael Schachtebeck, and Anita Schöbel. Evaluating line concepts using travel times and robustness. *Public Transport*, 5(3):267–284. 2013a.

Marc Goerigk, Marie Schmidt, Anita Schöbel, Martin Knoth, and Matthias Müller-Hannemann. The price of strict and light robustness in timetable information. *Transportation Science*, 48(2):225–242. 2013b.

Marc Goerigk and Anita Schoebel. Recovery-to-optimality: A new two-stage approach to robustness with an application to aperiodic timetabling. *Computers and Operations Research*, 52:1–15. 2014.

Rob MP Goverde. Railway timetable stability analysis using max-plus system theory. *Transportation Research Part B: Methodological*, 41(2):179–201. 2007.

Mads Hofman, Line Madsen, Julie Jespersen, Jens Clausen, and Jesper Larsen. Robustness and recovery in train scheduling-a case study from DSB s-tog a/s. In *ATMOS*. 2006.

Jian Gang Jin, Loon Ching Tang, Lijun Sun, and Der-Horng Lee. Enhancing metro network resilience via localized integration with bus services. *Transportation Research Part E: Logistics and Transportation Review*, 63:17–30. 2014.

Silke Jütte, Marc Albers, Ulrich W Thonemann, and Knut Haase. Optimizing railway crew scheduling at DB schenker. *Interfaces*, 41(2):109–122. 2011.

Peter Kall and Stein W Wallace. *Stochastic programming*. Springer. 1994.

Anton J Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502. 2002.

Niklas Kohl and Stefan E Karisch. Airline crew rostering: Problem types, modeling, and optimization. *Annals of Operations Research*, 127(1-4):223–257. 2004.

Spyros C Kontogiannis and Christos D Zaroliagis. Robust line planning under unknown incentives and elasticity of frequencies. In *ATMOS*. 2008.

Leo Kroon, Gábor Maróti, Mathijn Retel Helmrich, Michiel Vromans, and Rommert Dekker. Stochastic improvement of cyclic railway timetables. *Transportation Research Part B: Methodological*, 42(6):553–570. 2008.

Leo G. Kroon, Rommert Dekker, and M. J C M Vromans. Cyclic railway timetabling: A stochastic optimization approach. *Lecture Notes in Computer Science*, 4359:41–68. 2007.

Alex Landex and Lars Wittrup Jensen. Measures for track complexity and robustness of operation at stations. *Journal of Rail Transport Planning & Management*, 3(1):22–35. 2013.

Christian Liebchen, Marco Lübbecke, Rolf Möhring, and Sebastian Stiller. The concept of recoverable robustness, linear programming recovery, and railway applications. In *Robust and online large-scale optimization*, pages 1–27. Springer. 2009.

Christian Liebchen and Rolf H Möhring. The modeling power of the periodic event scheduling problem: railway timetables—and beyond. In *Algorithmic methods for railway optimization*, pages 3–40. Springer. 2007.

Christian Liebchen, Michael Schachtebeck, Anita Schöbel, Sebastian Stiller, and André Prigge. Computing delay resistant railway timetables. *Computers & Operations Research*, 37(5):857–868. 2010.

Ilse Louwerse and Dennis Huisman. Adjusting a railway timetable in case of partial or complete blockades. *European Journal of Operational Research*, 235(3):583–593. 2014.

Richard M Lusby, Jesper Larsen, Matthias Ehrgott, and David Ryan. Railway track allocation: models and methods. *OR spectrum*, 33(4):843–883. 2011.

Thomas L Magnanti and Richard T Wong. Network design and transportation planning: Models and algorithms. *Transportation science*, 18(1):1–55. 1984.

Ángel Marín, Juan A Mesa, and Federico Perea. Integrating robust railway network design and line planning under failures. In *Robust and online large-scale optimization*, pages 273–292. Springer. 2009.

Mathias Michaelis and Anita Schöbel. Integrating line planning, timetabling, and vehicle scheduling: a customer-oriented heuristic. *Public Transport*, 1(3):211–232. 2009.

Daniel Potthoff, Dennis Huisman, and Guy Desaulniers. Column generation with dynamic duty selection for railway crew rescheduling. *Transportation Science*, 44(4):493–505. 2010.

Miguel A Salido, Federico Barber, and Laura Ingolotti. Robustness in railway transportation scheduling. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pages 2880–2885. IEEE. 2008.

Miguel A Salido, Federico Barber, and Laura Ingolotti. Robustness for a single railway line: Analytical and simulation methods. *Expert Systems with Applications*, 39(18):13305–13327. 2012.

Anita Schöbel. Integer programming approaches for solving the delay management problem. *Lecture Notes in Computer Science*, 4359:145–170. 2007.

Anita Schöbel. Line planning in public transportation: models and methods. *OR spectrum*, 34(3):491–510. 2012.

Anita Schöbel. Integration of line planning and timetabling. In *Proceedings of the Conference on Advanced Systems in Public Transport*. 2015.

Anita Schöbel and Albrecht Kratz. A bicriteria approach for robust timetabling. *Lecture Notes in Computer Science*, 5868:119–144. 2009.

Anita Schöbel and Silvia Schwarze. A game-theoretic approach to line planning. *FIXME*, page 688. 2006.

Yoko Takeuchi and N Tomii. Robustness indices based on passengers' utilities. *WCRR2006, Montreal*. 2006.

Yoko Takeuchi, Norio Tomii, and Chikara Hirai. Evaluation method of robustness for train schedules. *Quarterly Report of RTRI*, 48(4):197–201. 2007.

Pieter Vansteenwegen and Dirk Van Oudheusden. Developing railway timetables which guarantee a better service. *European Journal of Operational Research*, 173(1):337–350. 2006.

Pieter Vansteenwegen and Dirk Van Oudheusden. Decreasing the passenger waiting time for an intercity rail network. *Transportation Research Part B-Methodological*, 41(4):478–492. 2007.

Michiel Vromans. *Reliability of Railway Systems*. Ph.D. thesis, Erasmus Research Institute of Management (ERIM), Erasmus University Rotterdam. 2005.

Michiel JCM Vromans, Rommert Dekker, and Leo G Kroon. Reliability and heterogeneity of railway services. *European Journal of Operational Research*, 172(2):647–665. 2006.

Chapter 7

An Optimization Based Method for Line Planning to Minimize Travel Time

With Natalia J Rezanova [1]
and Richard M Lusby [2]
and Jesper Larsen. [3]

Submitted to Transportation Research Part B.

[1]DSB. najr@dsb.dk
[2]Technical University of Denmark. rmlu@dtu.dk
[3]Technical University of Denmark. jesla@dtu.dk

Abstract. *The line planning problem in rail is to select a number of lines from a potential pool which provides sufficient passenger capacity and meets operational requirements, with some objective measure of solution line quality. We model the problem of minimizing the average passenger system time, including frequency-dependent estimates for switching between lines, working with the Danish rail operator DSB and data for Copenhagen commuters. We present a multi-commodity flow formulation for the problem of freely routing passengers, coupled to discrete line-frequency decisions selecting lines from a predefined pool. We show results directly applying this model to a Copenhagen commuter rail problem.*

7.1 S-tog problem description

The S-tog network in Copenhagen is a commuter rail network serving 84 stations and between 30,000 and 40,000 passengers per hour at peak times. The S-tog network is operated by the Danish state railway operator: DSB. The trains in the network operate on published lines which each have an hourly frequency, and run according to a published timetable. We consider the lines and the frequencies, but not the exact timetable.

See Figure 7.1 for an example of the lines that may operate in the S-tog network. Here each coloured path refers to a different line that is operated at some frequency, and on each a train visits every station marked on the line in each direction according to that frequency. An important feature of the network is that a train may not necessarily stop at every station it passes; for example the red and orange lines (C and H) run parallel to each other in the top left of the figure, and to the same end station, but the red line stops at fewer stations and is therefore faster between stations. This is of benefit to passengers travelling past those stations, but passengers travelling to or from the skipped stations are left with fewer options. For our purposes, the possibility of different stopping patterns greatly increases the problem size.

Given the fact that lines may not stop at all stations, a route or path in the network is not sufficient to define a line. We define a line here as a sequence of tracks which a train passes, and the stations on those tracks that are stopped at and not stopped at. We refer to the sequence of tracks as a line route, and the stations stopped at and not stopped at are the line stopping pattern. Paired with every line (route and stopping pattern) is an hourly frequency, and the set of lines with their frequencies defines the line plan.

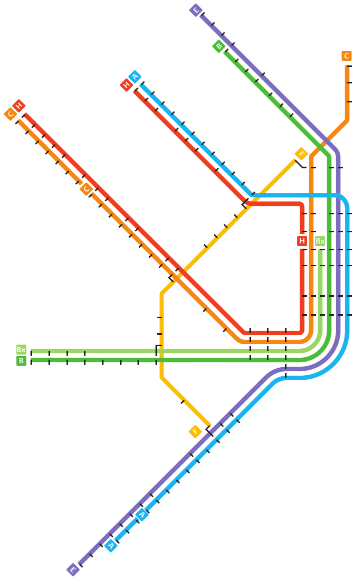


Figure 7.1: An example of the lines operated in the S-tog network, showing different lines in different colours. Each is identified by letter, and the presence of a dash indicates a stop.

Of the roughly 7,000 pairings of stations we consider non-zero demand for passengers between just over 4,600 of the pairs, which is around 65% of the possible demand pairings. As input we take a set of 174 valid lines, each with one or more valid frequencies at which the line could run; in total 350 line-frequency combinations are considered. Each line services between 11 and 39 stations with an average of 23 stations served per line, and almost all lines can operate at exactly two frequencies, while some very small number have more possible frequencies. However we also experiment with more frequencies.

Our demand data is for a specific peak period of the day, where in reality demand varies throughout the day. Real S-tog line plans have lines that operate at different frequencies at different times during the day, and operate modified line plans during the weekends. We make the following two assumptions, which are true for the S-tog problem:

- a line plan created for a peak time is valid at other off peak times, possibly operating at lower frequency;

- a line operates in both directions at the same frequency.

For the second point, we model both directions of the line as having the same frequency, where in practice it may be possible to operate each direction at a different frequency, though balancing vehicle movements may be more complicated.In practice current plans operate both directions of a line at the same in frequency, and we model the problem in that way. However, for capacity reasons the different directions of a line may not use the same rolling stock unit type; here we do not model rolling stock units but instead take a fixed capacity of the largest rolling stock unit type available. In practice there are no other differences between units that are relevant for our purposes here (for example plans are made for a fixed driving speed, not dependent on unit type) and through discussions with the operator we are confident that assuming all units are of the maximum size is a valid simplification. Assumptions on rolling stock unit are necessary because the line planning problem is solved early in a sequence of rail planning problems, usually followed by timetable creation and then rolling stock planning, and exact rolling stock details are only known at that stage. Without a timetable which itself depends on the line plan, we can only make estimates or assumptions about those details.

The presence of mixed stopping pattern lines running on the same infrastructure lines has the potential to negatively impact the timetable due to the mixed driving times of trains. This could be a source of a lack of *robustness* for the timetable and line plan. However, we do not calculate a timetable to assess this impact and make no estimate or derive any metrics for this particular feature.

The line planning problem we consider is that of selecting a set of lines from a larger pool of lines, and for each selected line, determining an hourly frequency at which the line should operate. A line is defined as a route in the infrastructure network with a stopping pattern, and several lines may share the same route but have different stopping patterns. The selected line plan must meet certain criteria, such as provide a minimum hourly service at each station, not exceed hourly limits on trains using certain track segments, visiting certain stations and turning at certain stations. These criteria are required due to the traffic contract between the operator DSB and the Danish Ministry for Transport

The line plan must also have sufficient capacity to transport all expected passengers, providing all with a good path from their origin to destination. As an objective measure we want to model the entire travel time of passengers, and include a frequency-dependent cost component to penalize occurrences of passengers switching to lines at low frequency, in favour of switching to lines at high frequency.

We take the line pool as a fixed input, and this is a subset of every possible route and stopping pattern. The lines in the pool are all assumed to be feasible alone, and are a sensible restriction of the entire pool excluding lines that are not considered to be appropriate or feasible. The pool is still sufficiently large to ensure the presence of a large range of feasible line-plan solutions.

In this paper, we present an integer program formulation for the line planning problem, formulated to allow a primary objective of total passenger travel time defined as time travelling in train and time switching between lines. We present a flow based formulation for routing passengers, and show that, despite being large, we use the formulation directly for real-world instances without requiring a path-based decomposition. However to reasonably solve the model, we present some contractions of the flow graph, present how we can aggregate passengers to reduce the problem size, and present additional constraints that improve the lower bound.

We present our results for the Danish S-tog case study, where we show a range of different line plans and how they can be compared with different metrics.

7.2 Related Research

There is much work in the literature on the line planning problem, with different details and objective measures. Schöbel [2012] gives an overview of line planning in public transport, and classifies different problem instance characteristics and models for addressing them.

In many line planing problems, the line and the route are interchangeable; if a line follows a certain route, then every station on the route is serviced. Goossens et al. [2004] take an operator-cost oriented approach, with discrete frequencies and carriage types selected for each line. The authors note that, due to passengers preferring to switch to a faster type of line as soon as possible, in a network with multiple line types a decomposition into problems considering each type is valid (and therefore in their problem all lines are of the same type). Goossens et al. [2006] however present several models for the train line planning problem where a route may have different stopping patterns (i.e. different line types), with a cost focus rather than a passenger focus. Other work at DSB for the S-tog problem [Rezanova, 2015] has used a version of one presented model to find low cost line plans.

There is a more recent focus on the passenger, and on minimizing the total trip time for passengers (and so modelling their moving and switching times) as we would like to do for the S-tog system. Schöbel and Scholl [2006] present a model with a station-line graph in which passengers are freely routed with both travel times and switching penalties, with continuous frequency decision variables for lines, and use a decomposition such that decisions are made in terms of path selection for every pair of stations. Nachtigall and Jerosch [2008] present a model where passengers are routed freely, measuring travel time with fixed penalties for switching lines, and with an integer decision variable per line. Borndörfer et al. [2007] similarly presents a formulation freely routing passengers (though ignoring transfers) and also dynamically generating lines, using continuous frequency decision variables. In contrast we wish to model the switching cost with a frequency dependence. Also, the line frequencies in the S-tog system are not so free that we can model them continuously as there are only discrete frequencies that are considered valid. In addition to passenger travel time, Borndörfer et al. [2007] use operator running cost and fixed line-setup costs into account in their model.

A simpler passenger-focused measure is using a direct-travellers objective, maximizing the number of direct travellers the network may transport (i.e. passengers who may travel with no transfers). Bussieck et al. [1997] present such an approach while, like many authors, selecting lines from a pool (by selecting for each line in the pool an integer frequency, which may be zero). A pool-based approach is used by many authors, and is used in this work.

Goerigk et al. [2013] take an approach selecting an integral decision variable for each line in a pool, taking either an operator cost approach; a direct travellers approach, or an approach ensuring an even distribution of frequencies in the public transport network. Here, unlike in much work, the authors' main concern is the impact the plan has on a subsequent timetable. However in this case the authors draw a positive correlation between more direct travellers and lower overall travel times, because with fewer overall transfers the transfers can be scheduled tightly in a subsequent timetable.

7.3 Lines Model

We take as input a set of valid lines, $\mathcal{L}$, and for each line l there is a predefined set of discrete frequencies at which the line could operate: $\mathcal{F}_l$.

Ignoring passengers, we may simply find line plans (pairings of lines with frequencies) which satisfy all operational limits, and consider how well they serve passengers. In general such solutions do not even guarantee sufficient capacity for all passengers, though often they are very close; the minimum visits requirement per station in many areas provides more capacity than there are passengers travelling to, from or passing by the station. However, even if a solution does provide sufficient capacity, it is possibly a very poor quality solution for passengers.

We decide which lines and frequencies we will select from the line pool $\mathcal{L}$, where each has valid frequencies $\mathcal{F}_l$ (defined for each $l \in \mathcal{L}$). We let the binary decision variable $y_{lf} \in \{0, 1\}$ denote selecting line l at frequency f.

Simply selecting a valid set of lines is not in itself trivial; the selected lines must be compatible, must meet certain service levels, and must not exceed some fixed operating budget. The service level requirements can all be expressed as a minimum number of trains visiting a single station per hour, or operating on a particular track sequence per hour. Similarly, the compatibility requirements can be expressed as a maximum number of trains per hour visiting stations, turning at stations, and operating on particular tracks. Selecting a line at frequency f contributes f trains per hour towards the relevant service level constraints, and so we can enforce such constraints by summing over every line frequency decision with the frequency itself as the coefficient.

Every contractual requirement or operational limit can be expressed by determining exactly those lines which contribute toward the requirement or limit such as lines visiting the relevant station or using the relevant track sequences. Consider such a set $\mathcal{Z}$ of lines. The contractual requirement or operational limit for $\mathcal{Z}$ may have either a lower limit or an upper limit or both for the number of trains per hour. For simplicity in definition we assume both; let these be $\alpha(\mathcal{Z})$ and $\beta(\mathcal{Z})$ for the lower and upper bounds, respectively. Now, let $\mathcal{C}$ be the set of all such sets $\mathcal{Z}$; every element of $\mathcal{C}$ is a set of lines $\mathcal{Z}$ with a lower ($\alpha(\mathcal{Z})$) and upper ($\beta(\mathcal{Z})$) hourly limit.

Additionally, certain sets of lines are inherently incompatible for various reasons not explicitly related to the line plan but for other operational reasons. Let $\mathcal{I}$ be the set of all incompatible sets of lines, where any element of $\mathcal{I}$ is a set of lines from which only one line can appear in a valid line plan.

Finally, every line has a cost when operated at a particular frequency: c_{lf}, and we impose a maximum budget for the line plan $c_{\max}$. This generalized cost may not necessarily scale with frequency; selecting a line at frequency $2f$ may cost more or less than selecting the line at frequency f.

Now, the following constraints define a valid line plan, considering only the lines themselves but ignoring passengers.

$$\sum_{f \in \mathcal{F}_l} y_{lf} \leq 1 \qquad \forall l \in \mathcal{L} \tag{7.1}$$

$$\sum_{l \in \mathcal{L}} \sum_{f \in \mathcal{F}_l} c_{lf} \cdot y_{lf} \leq c_{\max} \tag{7.2}$$

$$\sum_{l \in \mathcal{Z}} \sum_{f \in \mathcal{F}_l} y_{lf} \leq 1 \qquad \forall \mathcal{Z} \in \mathcal{I} \tag{7.3}$$

$$\sum_{l \in \mathcal{Z}} \sum_{f \in \mathcal{F}_l} f \cdot y_{lf} \geq \alpha(\mathcal{Z}) \qquad \forall \mathcal{Z} \in \mathcal{C} \tag{7.4}$$

$$\sum_{l \in \mathcal{Z}} \sum_{f \in \mathcal{F}_l} f \cdot y_{lf} \leq \beta(\mathcal{Z}) \qquad \forall \mathcal{Z} \in \mathcal{C} \tag{7.5}$$

$$y_{lf} \in \{0,1\} \qquad \forall l \in \mathcal{L}, \quad \forall f \in \mathcal{F}_f \tag{7.6}$$

Constraints (7.1) ensure that a given line is chosen at most once disallowing a single line at multiple frequencies (because, for example, some line might be permitted at 3, 6, or 12 times per hour but not at 9 times per hour, so combinations may not be permitted). Constraint (7.2) ensures that the total lines cost is no greater than the given budget. Constraints (7.3) permit only one line for each of the sets of incompatible lines. Similarly, constraints (7.4) provide minimum service levels for the same visits, turnings or track usages. Constraints (7.5) provide all operational constraints that can be expressed as a maximum number of trains visiting or turning at a station, or using a specific sequence of track.

The formulation (7.1)–(7.6) defines a valid line plan. It completely ignores passengers; some feasible solutions to the formulation will fail to provide sufficient capacity for all passengers in the network, and those that do provide sufficient capacity may nevertheless provide a poor solution for many passengers. However, solving the formulation will find a line plan with some capacity that services all stations, so it can be assessed to determine whether or not it does provide sufficient capacity. If so, we can assess how well it serves passengers, and if not we can identify where capacity is lacking.

7.4 Passengers

7.4.1 Graph

We model passenger travel as a movement in a graph, where the existence of components of the graph depends on the presence of a line in the solution. We could model each line-frequency pair as a completely distinct component of the graph. However, this leads to a very large graph, especially if we want to experiment with many frequencies for each line, and much of the information depends on the line itself and not its frequency.

Consider Figure 7.2 showing the structure of a single line at a single frequency visiting three stations. For each station (1, 2, 3) there are three vertices; a source vertex, a sink vertex, and a platform vertex (s_i^1, s_o^1, p^1 for station 1, respectively). All passenger paths originate from some source vertex, and terminate at some other sink vertex, travelling on dashed line travel edges or switching lines using a platform vertex. To capture the information we want about frequency-dependent aspects of the line, we could simply duplicate this structure for every frequency at which the particular line operates. That is, we would have a parallel structure to $l_1, l_2, \ldots$ vertices representing the same line with route and stopping pattern, but operating at a different frequency. However, much of the information would be redundant, and when experimenting with large numbers of frequencies per line the graph becomes very large. Alternatively we could simply have one such structure that represents every frequency, except that then the cost of a particular path could have no dependence on frequency of lines used. In our problem we want to penalise switching to low frequency lines more than high frequency lines. However capacities of edges, though dependent on frequency, can still be maintained even with a single structure by summing the capacities of the frequency-line decisions that would contribute toward them. This suggests that it is possible to partially aggregate the line-frequencies into simply lines, being careful to accommodate the frequency-dependent switching cost between lines.

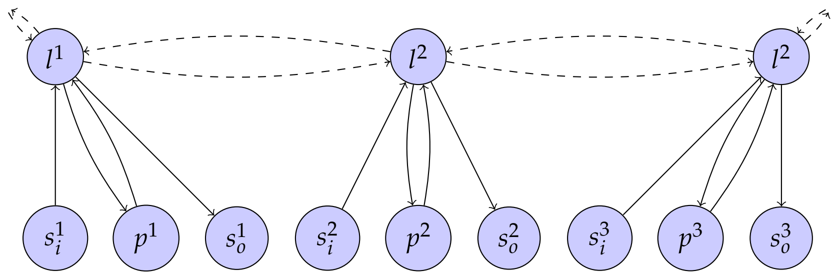


Figure 7.2: The structure for a single line at one frequency visiting multiple stations.

The aggregated graph then contains three types of node:

- a source and sink for every station;
- a platform for every station;
- a station-line for every station a line visits, for every line.

The graph also contains several types of edges:

- A travel edge between every adjacent pair of station-line edges for every line, in each direction;
- A get-off edge from every station-line to every station sink;
- A get-off edge from every station-line to every station platform;
- A get-on edge from every station source to every station-line;
- A get-on edge from every station platform to every station-line at every frequency.

Note that this is an aggregation of the line/frequency combinations, though without being able to aggregate those boarding frequency edges. It means the capacity of an edge (the station-line to station-line edges) is dependent on a summation over all frequency decisions for that line.

The graph structure is similar to the stop-and-go graph described by Schöbel and Scholl [2006]. See Figure 7.3 for the structure of the problem graph for passengers. The figure shows a single station with its three nodes, and two distinct lines which visit the station at two frequencies each. Here depicted as a multi-graph, the graph can be made simple with auxiliary nodes and edges. For each passenger, a path through the graph from their origin station s_i^1 vertex to their destination station s_o^2 vertex must be found, which incurs the travelling time (on dashed edges) and switching time costs (on red edges). Differing from the Schöbel and Scholl [2006] problem structure, in our case the discrete frequencies a line may operate at are an important feature, and we want to model different passenger time costs for switching to lines at different frequencies, so our graph has additional station structure.

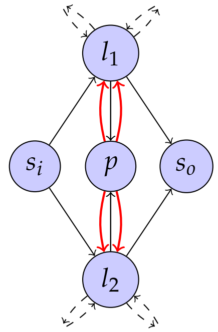


Figure 7.3: The graph structure for two lines at a station. Each station has three vertices, s_i and s_o, which representing either entering or exiting the system at this station, and p, which represents switching lines at the platform. Vertices l_1 and l_2 represent the two lines visiting the station. The solid black edges have zero cost, while the dashed black edges cost the travel time to the next station along a line. The thick red edges represent the costly switching from one line to another, and depend on the frequency of the boarded line (one edge per frequency the line may operate at).

7.4.2 Flow decisions

The problem can be represented as a multi-commodity flow problem, with one commodity per OD pair, with additional constraints linking flows to line presence and capacity. However, with the roughly 4,600 OD pairings in our regular problem instance and the relatively large graph we describe above, the problem would be very large. Let us refer to such a model, which we do not formulate here, as a per-OD arc flow model, where for every OD we would select a proportion of passengers who use every edge in the graph such that every OD has one path from origin to destination, and those edges used correspond to selected lines and frequencies. We have tested the per-OD arc flow model for small instances (such as with only the *lines* of a known feasible solution, but unspecified frequency still to be determined) and, though solvable, the model is very large and would not scale to having very many lines.

The proposed per-OD arc flow model would have one flow variable for every OD combination, for every edge in the graph, and we would require one path with capacity sufficient for that OD demand for every OD combination, respecting every other OD path. As an aggregation, we can combine flows that have the same origin (or alternatively the same destination), and instead have one type of flow for every origin. The number of flow decisions is then lower by a factor of $|\mathcal{O}|$. Instead of requiring one path per OD, we will require the aggregation of those paths scaled by passenger counts; that is, we will require a network flow from each origin which supplies the sum of passengers from the origin to every destination, and each destination from that origin consumes just the passenger demand from the origin to that destination. The flow variables are $x_o^e \geq 0$; the number of passengers from origin o using edge e. Note that we do *not* require integer flows, and we do not require a single path between every origin and destination. In fact we see for currently used plans that it is infeasible for every OD pair to use a single path, as there is insufficient capacity. Instead some proportion of passengers on some OD trips are forced to take less favourable paths than the best available due to limited capacity on their most attractive path. Here, we note that the capacity we take for an operating line is assuming the largest possible rolling stock unit is operating the line, while in reality DSB operates different units with different capacities. In discussions with DSB we determined that this simplification was appropriate and, while it potentially over-estimates the true capacity achievable for a line plan (as DSB has too few of the largest units to use them everywhere), we do not see solutions where this capacity is required on every line simultaneously.

In the graph described in Section 7.4.1, let $\mathcal{V}$ be the set of vertices and $\mathcal{E}$ be the set of edges. For every vertex in the graph, we define a demand for passengers for every origin station in the network. Let the demand for passengers at vertex v who originate from station s be a_v^s.

$$a_v^{s_1} = \begin{cases} d_{s_1 s_2} & \text{if vertex } v \text{ is a sink vertex for station } s_2 \\ -1 \cdot \sum_{s_2} d_{s_1 s_2} & \text{if } v \text{ is the source vertex for station } s_1 \\ 0 & \text{otherwise} \end{cases}$$

We also require constraints that link the flow variables to the line decision variables y_{lf}, ensuring both that if any flow uses a line, then the line is present, and that every connection of the line has sufficient capacity for all flows which use it. Further, we will require that the edges corresponding to the frequency-dependent boarding of a line are only used if the line is present at the correct frequency. Constraints linking the flow variables to the capacity of the selected lines are in fact sufficient, and it is not necessary to impose additional constraints to link simply a usage of a line to a line decision variable. To do this, let $\mathcal{E}^l$ be the set of all edges in the graph that depend on the presence of line l at undetermined frequency. Let $\mathcal{E}_f^l$ be the set of all edges in the graph that depend on the presence of line l at exactly frequency f.

We impose the following constraints:

$$\sum_{(u,v)\in\mathcal{E}} x_s^{(u,v)} - \sum_{(v,w)\in\mathcal{E}} x_s^{(v,w)} = a_v^s \qquad \forall s \in \mathcal{O} \quad \forall v \in \mathcal{V} \tag{7.7}$$

$$\sum_{o\in O} x_o^e \leq \sum_{f\in\mathcal{F}_l} P_f y_{lf} \qquad \forall l \in \mathcal{L}, \forall e \in \mathcal{E}^l \tag{7.8}$$

$$\sum_{o\in O} x_o^e \leq P_f y_{lf} \qquad \forall l \in \mathcal{L}, \forall f \in \mathcal{F}_f, \forall e \in \mathcal{E}_f^l \tag{7.9}$$

Constraints (7.7) ensure that flow is conserved in the graph, for every origin defining a network flow moving the required number of passengers from each origin to every destination. Constraints (7.8) ensure that for every edge on a line ($\mathcal{E}^l$), the edge provides sufficient capacity for all flows using it. The constant P_f is the capacity of any line at frequency f, which we take as a constant. An obvious simple extension is to have a line-specific capacity, but that is not present in our data. Finally, constraints (7.9) ensure that for those boarding edges from a platform that are frequency dependent (edges $\mathcal{E}_f^l$ for line l at frequency f), again sufficient capacity must be present. In effect, the difference between constraints (7.8) and (7.9) is that (7.8) are for the aggregated frequency edges and therefore we sum the y_{lf} variables for all frequencies.

These constraints (7.7)–(7.9) define the flows and link them to the line decision variables. The full formulation then is constraints (7.1)–(7.6), and (7.7)–(7.9).

7.5 Objective functions

There are a number of possible objective functions we could use, either related to passengers, to the operator, or to some combination. Above with constraints (7.1)–(7.6) we gave the operator cost as a constraint with a fixed budget, and our primary goal is to minimize a passenger-based objective. However, other measures are possible. Here we give our primary measure and some other alternative measures we use.

7.5.1 Passenger travel time

As already stated, we are interested in penalising switching time for passengers with emphasis on discouraging switching to lines operating at low frequency. As we do not know the timetable in advance, we can't know the exact time required for a switch. In the ideal case, for every switching occurrence, both trains would arrive at a station at the same time and the station layout would permit passengers to switch from either train to the other, losing no time. However, generally this is impossible in the S-tog system. The best case at most stations is that one train arrives shortly before another, in such a way that passengers may switch from the earlier train to the later time with minimal waiting time, but then passengers switching in the opposite direction have almost a worst-case wait time for their next train.

Overall, we consider passenger travel time to be the most appropriate measure. In tests, if we ignore switching time and minimize only moving travel time, we find solutions with many undesirable switches required. Conversely if we ignore travel time and consider only minimizing some measure of switch cost, we find solutions which do not use "fast" lines appropriately and have higher overall average total travel time. Travel time therefore includes both the moving travel time on train lines and an additional estimate of the wait time. However, in addition to this we include a separate expression for the "unpleasantness" of switching lines which we express as a time, in effect calculating a weighted sum of estimated travel time and the number of switches.

For every edge in the graph, we assign some cost to the passenger. Let t_e be the time cost to one passenger for using edge e. For every travel edge on a line (the dashed lines in Figure 7.3), the edge time cost is the exact, known, travel time for trains between the two stations. However, for the frequency-dependent switching edges (the red edges on Figure 7.3), the edge time cost includes an estimate of the waiting time and the penalized fixed cost of switching. For such edges e at frequency f, let $t_e = p_{\text{fixed}} + \lambda \frac{1}{f}$, where $\lambda \in [0, 1]$. That is, the time cost is a fixed term with a fraction of the worst case wait time (where for example in the worst case, a line operating twice per hour has a worst case switch time of $\frac{1}{2}$ an hour). We take, as a parameter, a fixed penalty of six minutes and $\lambda = 0.5$, or an average case wait time estimate. For any other edges let $t_e = 0$. Now, we can define our objective function as simply:

$$\sum_{e \in \mathcal{E}} \sum_{o \in \mathcal{O}} t_e x_o^e \tag{7.10}$$

7.5.2 Minimum lines cost objective

We are not only interested in line plans which have minimal passenger cost, but also those that have low operating cost while still providing a good passenger service. An operating cost objective can easily be used with the formulation (7.2)–(7.6), iteratively finding many low cost solutions and assessing their passenger cost. However it is likely that such plans are not feasible; minimal line cost solutions likely provide low passenger capacity. We can use the following as our objective function, which expresses operating line cost:

$$\sum_{l \in \mathcal{L}} \sum_{f \in \mathcal{F}_l} c_{lf} \cdot y_{lf} \leq c_{\max} \tag{7.11}$$

We observe that there are many similar cost solutions with the same lines but with different frequencies, where the aggregated sum of frequencies is the same (or similar). If we iteratively find solutions and forbid them (which we can do with constraints described in a later section), we see many similar solutions. Also, we observe that if a low-cost solution of lines with some frequencies is infeasible for passengers, then a different low-cost solution with the same lines at different frequencies is also likely to be infeasible for passengers. However there is also likely to be other solutions with the same lines at different frequencies that *are* feasible, but not low-cost. We can also find these similar solutions by taking a limited line pool consisting only of the lines present in a low cost solution, but at any frequency, and solve a problem with passenger cost as the objective. A feasible solution is guaranteed to be present as every line and frequency of the provided solution is present, but we may find there are better feasible solutions with respect to the passenger objective.

7.5.3 Direct travellers objective

In model (7.2)–(7.6) we can instead use a (pseudo) direct travellers objective, seeking to maximize the number of direct travellers. This will not be entirely correct as it will count the number of passengers who have a direct path in the line plan, but without the guarantee that all such passengers may take their direct path (due to limited capacity).

To measure the direct travellers we introduce a new binary variable: let $z_{od} \in \{0,1\}$ denote whether the line plan provides a direct connection between stations o and d. Then, let $\mathcal{L}_{od}$ be the set of lines which provide a direct connection between stations o and d, which can easily be precomputed. We then maximize the objective function:

$$\sum_{(o,d)\in\mathcal{O}} d_{od} \cdot z_{od} \tag{7.12}$$

where d_{od} is the demand between stations o and d.

We require the following additional constraints:

$$z_{od} \leq \sum_{l\in\mathcal{L}_{od}} \sum_{f\in\mathcal{F}_l} y_{lf} (o,d) \in \mathcal{O} \tag{7.13}$$

That is, z_{od} may have value 1 if there is one line providing a direct connection between o and d in the line plan.

As mentioned in the previous section, we may iteratively find solutions by forbidding those we do discover with appropriate constraints. We then assess the feasibility and travel time cost of the line plans. The results are remarkably poor; the direct traveller objective is not a good approximation of our true travel time estimate objective. For the S-tog problem we assessed the first three thousand solutions; all had very high line operating cost, and almost all had high travel time cost as well. None was a noteworthy solution; all discovered solutions are dominated by other solutions, even (some of) those found with minimal operator cost in the previous section.

Examining the discovered solutions, there are two explanations for the poor quality of the solutions favouring direct travellers. Firstly, lines that skip stations in our network provide a small travel time benefit to potentially many passengers, but provide fewer direct connections than lines with the same route stopping at all stations. If measuring only direct travellers then these faster "skipping" lines are less valued, and almost none appear in any solutions we discover. Secondly, the solutions have a very wide selection of lines, with more lines being present overall and those lines that are present all terminate at extreme end stations of the network, not intermediate end stations. This then has the problem that the line frequencies must all be low to accommodate many lines turning at each of the extreme end stations, and then by our travel time estimate, those passengers that do *not* have a direct connection are then required to switch onto lines operating at low frequency, incurring higher cost. We see that, indeed, fewer passengers are required to make connections (roughly 30% fewer connections in total than plans targeting total travel time), but that is still then a large number of passengers and most are switching to low frequency lines.

The direct travellers objective is used by other authors, and in fact has been noted as being a source of a lack of delay robustness for a timetable [Goerigk et al., 2013]. The direct travellers objective results in line plans with very many lines, and therefore those transfers that are required are tight in the timetable and susceptible to being missed. In the S-tog network there is not the possibility for particularly many lines, although a direct traveller objective does discourage using the "intermediate" end stations on the fingers and instead using the end depots at greater capacity. However a direct travellers objective in the S-tog case discourages skipped stations which leads to more homogeneity in driving speed, which might be expected to provide better delay robustness.

7.6 Other modeling considerations

7.6.1 Additional linking constraints

In the previous section we defined a formulation where the linkage between edge flows and line presence is imposed only by the capacity of a line and the sum of all usages of every element of the line.

In general, in our problem, the demand between some particular pair of stations is lower than the capacity of a line operating at only the lowest frequency. In a non-integer solution only a small fractional line decision variable (y_{lf}) is required to provide capacity for some OD pair to make use of the line, if no other OD pair uses that line. Suppose for an OD based arc flow model there are variables x^e_{od} deciding the flow on edge e for flow from o to d. In addition to summing all such flows for every OD pair for the usage of the line y_{lf} which contains e to constrain the capacity of the edge, the following constraint could be used:

$$x^e_{od} \leq d_{od} \cdot \sum_{f \in \mathcal{F}_l} y_{lf}$$

where d_{od} is the demand for pair (o, d). This would provide a tighter linkage between the flow variables and the line variables, at the expense of requiring very many constraints, though it is not necessarily required that such constraints are included for every edge of a line.

Unfortunately, we do not have individual flow variables x^e_{od} as we have aggregated the variables by origin; we have only x^e_o. Unlike previously, where the maximum flow on any edge for one (o, d) flow was d_{od}, now the maximum demand of flow on any edge from one origin is $\sum_d d_{od}$, which is *not* in general significantly smaller than the capacity provided by one line; in fact it can often be that any one line capacity is less than the aggregated demand. The analogous constraint is much weaker:

$$x^e_o \leq \sum_{f \in \mathcal{F}_l} y_{lf} \cdot \sum_d d_{od}$$

In general, the flow originating from an origin has much higher edge usage than any single d_{od}, and close to the origin itself it may in fact be as high as $\sum_d d_{od}$. However the flow from that origin terminates at many destinations and at those destinations the flow is much lower; exactly d_{od} flow terminates at a particular destination from some origin, and that corresponds to usage of an edge in our graph that belongs to a specific line and is *only* used by flow terminating at that destination. That can be seen on Figure 7.3, where edges to s_o can each be associated with a single line, and where there are no edges out of s_o. Therefore, for every such specific edge e we can include the following constraint:

$$x_o^e \leq d_{od} \cdot \sum_{f \in \mathcal{F}_l} y_{lf}$$

Let t_{ld} be the terminating edge of line l at destination station d (if there is destination d on line l). Then, we impose the following for every line and for every (o, d):

$$x_o^{t_{ld}} \leq d_{od} \cdot \sum_{f \in \mathcal{F}_l} y_{lf} \tag{7.14}$$

Given our tight operational constraints, as well as the budget constraint (constraints (7.2) and (7.5)), such constraints improve the bound given by solving the LP relaxation of the model, as in general the forcing of some line variables to have higher value must cause a decrease in others, and then some passengers must use less favourable lines. However this comes with the addition of many new constraints; one for every OD pair, for every line that visits the destination of the pair (up to $|\mathcal{L} \times \mathcal{O} \times \mathcal{O}|$ constraints).

The formulation is constraints (7.1)–(7.6), (7.7)–(7.9) and (7.14).

Figure 7.4 shows how the LP lower bound differs with and without constraints (7.14). Here the problem itself is the S-tog problem, and more precise details are given later. The effect differs with the budget limit: without the constraints only a very strict budget limit has any effect on the LP bound, while with the constraints the bound is affected more with a wider range of budget limit levels. The difference between the two bounds is most pronounced at lower budget limits than higher, but in all cases is less than half way from the worse lower bound to a best IP solution (though the IP solutions were only obtained with a time limit of 1 hour and so not all are optimal). Note that for the IP solutions, the actual lines cost may not correspond exactly to one of the cost constraints applied, as with the applied constraint the best solution found may have a lower lines cost.

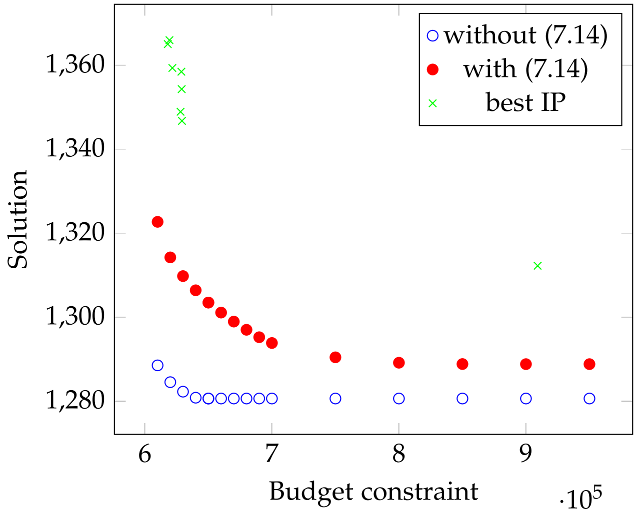


Figure 7.4: The LP relaxation lower bound found for different cost constraints with (blue open circle) and without (red filled circle) constraints (7.14). Green cross-marks show some best integer solutions found with a one hour time limit for a range of cost constraints.

7.6.2 Forbidding a solution

Let $\mathcal{S}$ be a set of lines in a particular solution, and let f_l be the frequency of line l in $\mathcal{S}$. The following constraint forbids this exact solution:

$$\sum_{l \in \mathcal{S}} y_{l,f_l} \leq |\mathcal{S}| - 1 \tag{7.15}$$

This has the potential problem that it does not forbid a solution containing only some of the lines in $\mathcal{S}$, or conversely that it does forbid solutions which contain the lines of $\mathcal{S}$ and additional lines. As an alternative, a solution with the given solution lines at *any* frequency can be forbidden with the following constraint:

$$\sum_{l \in \mathcal{S}} \sum_{f \in \mathcal{F}_f} y_{lf} \leq |\mathcal{S}| - 1 \tag{7.16}$$

As before, this does not forbid a solution containing only some of the lines in $\mathcal{S}$, and does forbid solutions with additional lines. However, we find that to be acceptable for our problems.

Forbidding solutions in this way is used to remove solutions known to be infeasible for later planning; and iteratively, to find a large pool of solutions by repeatedly solving the model, forbidding the last found solution, and solving again.

7.6.3 OD grouping

Stations can be grouped together if they are served similarly by all lines. Consider two adjacent stations, s_1 and s_2, which lie on a track sequence, and all lines in the line pool stop at either *both* s_1 and s_2 or *neither* s_1 nor s_2. That is, they are served identically by all lines. Then, if there is a third station s_3 with demand to both stations s_1 and s_2 we can treat the two demands as a single combined demand to (say) s_1. Any demand for travelling directly from s_1 to s_2, or vice versa, which would be discarded, can be reserved by requiring sufficient aggregated extra capacity on the lines visiting both stations. This may under-reserve capacity on the connection between s_1 and s_2 if s_1 is closer to s_3 than s_2 is, or over-reserve capacity if s_1 is further from s_3. We optionally apply a pessimistic grouping strategy which reduces the problem size (by reducing the total number of OD pairs), but given the potential error in underuse or overuse of some connections, we only consider low magnitude OD pairs and always assess solutions found using all OD pairs.

With reference to Figure 7.1, the S-tog network has a central region and several "fingers". In the presented plan, on each finger, one line serves every station and the other line serves only some stations, but this is not necessarily true for any solution plan. However it is in the fingers where grouping is most relevant; for example the final few stations on every finger may not be skipped by any line, and are therefore visited identically by every line and may be grouped.

7.7 Instance size

The OD data is not necessarily well-suited to all applications in that the hourly demand between many station pairs is non-integral (being the output of a demand model). Table 7.1 shows the total passenger demand for different rounding strategies of the individual OD demands between pairs of stations.

Table 7.1: Total OD demand for different roundings of individual OD demands

method	passenger number
rounded up	+6.52%
rounded down	-6.53%
rounded	-0.02%

Always rounding down results in a loss of over 6% of passengers, while rounding up adds more than an additional 6% passengers in total. Some solutions (including real solutions) are very tight in capacity in certain sections, and the additional passengers in the round-up scenario can make such solutions infeasible. Rounding down has a similar problem, in that solutions may be found which have very limited capacity and insufficient capacity for the true total number of passengers. We instead round all demands to the closest integer number of passengers which results in a total passenger demand very close to our raw input data, and is less likely to have such capacity problems.

Now that all components of the problem are defined, we can more explicitly define the standard S-tog instance size. Table 7.2 shows the size of various sets defined earlier.

Table 7.2: Sizes of problem instance parameters, and some derived values.

parameter	explanation	size
$\lvert\mathcal{L}\rvert$	Number of lines	174
$\sum_{l\in\mathcal{L}} \lvert\mathcal{F}_l\rvert$	Number of line-frequency decisions	350
$\lvert\mathcal{I}\rvert$	Number of marked incompatible lines	258
$\lvert\mathcal{C}\rvert$	Number of operational requirements	189
$\lvert\mathcal{OD}\rvert$	Number of (o, d) pairs	4645

Figure 7.5 shows the cumulative number of passengers for (o, d) pairs, when sorted by the number of passengers demanding a route for each (o, d). A small proportion of the (o, d) pairs account for the majority of the passengers. A potential simplification may then be to simply ignore some proportion of (o, d) pairs with small demands; however experiments solving reduced problems and then assessing solution quality considering all ODs gave poor results, as

those low-demand ODs cover a diverse range of station pairings that are given insufficient consideration.

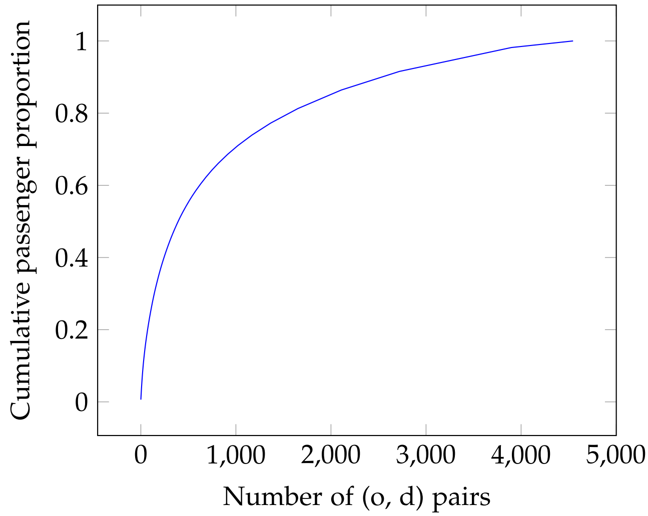


Figure 7.5: The cumulative proportion of passengers when considering (o, d) pairs sorted in decreasing order by the size of their passenger demand.

7.8 Method

Primarily, we use the formulation as described, with a constraint on the operator cost which we vary to find different results, and the total passenger time estimate as the objective function. Solving the resulting mixed integer program (MIP) directly can often provide good quality solutions, but we propose two additional methods for finding solutions.

In the following, we will refer to two real solutions: R1 and R2. These are both real historic line plans and frequencies as operated by DSB.

All testing is implemented with Gurobi 5.6 as the MIP solver on a machine with 8 GB of memory, and a four-core 2.5 GHz i7 processor.

7.8.1 Refining a given solution with a limited pool

From a given solution, we can create a limited pool of lines to use as the input to the model, which may or may not contain all the lines of the given solution and possibly also has the original solution as the optimal solution. The advantage of using a limited pool is that if the pool is sufficiently small, the model is solvable to optimality in reasonable time.

Suppose a solution is given, and let $\mathcal{S}$ be the set of lines present in the solution, with unspecified frequency. A simple restricted instance is to solve the model with only the lines $\mathcal{S}$, but with the definition of $\mathcal{F}_l$ for each line l in $\mathcal{S}$ unchanged. This is then a very small line pool (with generally one or two frequencies per line), but there is guaranteed to be at least one feasible solution present and likely to be others. We see that solving this limited model for any given solution can quickly find very similar but better solutions, especially in the case of real past line plans which were generally not planned with the same objective we use, and likely respect additional requirements. The similarity of any solution found to a real past solution is potentially useful, avoiding solutions that are significantly different to a plan that is not only feasible but known to operate in practice, which we can not in general guarantee.

To determine a wider but limited line pool, consider a set of lines from the entire line pool that are similar to a given line; let $N(l)$ be a set of "neighbouring" lines to line l which only differ in some small way to l. Now, we may use the following as a limited line pool:

$$\bigcup_{l \in \mathcal{S}} N(l)$$

We assume that $l \in N(l)$, and therefore $\mathcal{S}$ is a subset of this limited line pool.

The definition of $N(l)$ has a large effect on the problem size and solution quality. For example if $N(l) = \{l\}$ for all lines in $\mathcal{S}$, then this limited line pool is the same as simply taking the given solution lines. Alternatively, if $N(l)$ is very large then the resulting problem may have every line in the entire pool $\mathcal{L}$, and the line pool would not be "limited" at all.

Table 7.3 shows the problem sizes if we apply these two options, either taking lines but unspecified frequencies, or expanding with neighbouring lines, to the two real line plans R1 and R2. Here we indicate the solution with fixed frequencies as R1, the problem with the lines of R1 but open frequencies as R1$^+$, and the problem with all neighbouring lines to R1 as R1* (and the equivalent for R2). For each, we report the number of considered line-frequency combinations, and for the expanded problems report the solve time and the percentage improvement in the moving time, train switching time, and line cost. Note that the line cost is not considered in the objective, and as expected it increases, while we see modest improvements in the components of the total travel time.

Table 7.3: Solve times and objective improvements for different limited line pools. Cost improvements are the improvement in total moving time and switching time, and the improvement in operator line cost (which in fact becomes worse as operator cost may increase if it below the budget constraint).

	Lines			Cost improvements		
Problem	$\lvert\mathcal{L}\rvert$	$\sum_{l\in\mathcal{L}} \lvert\mathcal{F}_l\rvert$	Solve time (s)	Moving	Switching	Line
R1	9	9	-	-	-	-
R1$^+$	9	19	1	0.1%	9.7%	-3.3%
R1*	29	59	395	0.1%	9.7%	-3.3%
R2	8	8	-	-	-	-
R2$^+$	8	17	1	0.2%	5.5%	-2.4%
R2*	25	52	175	1.5%	4.5%	-5.3%

Solve times for the R1* and R2* instances are much greater than the R1$^+$ and R2$^+$ instances, and in the case of R1* no improvement is seen over R1$^+$. However, we can see that we can relatively quickly find line plans that "neighbour" a given plan, and here we can improve over these real line plans with very similar line plans (in the case of R1$^+$ and R2$^+$ the found solutions modify only the line frequencies). We can see here that there is more scope for improvement in switching time than travel time, given these reduced problems. Note, however, that here we report the percentage improvement for each individually, but the magnitude of those changes is very different, and in our problem a small relative improvement in travel time can be more significant than a large relative improvement in switching time.

When we solved both R1$^+$ and R2$^+$, the moving time improved even though we had exactly the same lines and can alter only frequencies. This may seem impossible, as the travel time between any pair of stations is unchanged. However the reason that we see small improvement is that either some passengers did not take their quickest (moving time) route due to a costly low frequency connection, or because some passengers could not take their fastest (moving time) route due to a lack of capacity, but with higher frequency (and therefore capacity) can now take that route.

7.8.2 LP heuristic

We propose the following as a simple heuristic for finding solutions. Initially, we solve the LP relaxation of the model and consider exactly those y_{lf} variables that have non-zero value. Then, we restrict the problem to only those lines present but find the optimal integer solution with the restricted problem. We compare the value of the optimal integer solution to the initial lower bound to the LP that we had, and, if we wish, we can re-introduce the missing line-frequency decision variables and allow the solver to tighten that lower bound and potentially discover better integer solutions. The advantage we see is that it is much faster to solve to optimality when there is a restricted pool of lines, and so we can relatively quickly find good solutions. In fact, in some experiments the solutions are optimal or near optimal. We hope that the smaller resulting problems have acceptable solve times but still have solutions of good quality. Also, as a possibility, we can expand the lines in the LP solution using the ideas from Section 7.8.1.

We compare four different formulations, summarized in Table 7.4. The formulations differ in the presence of the additional constraints (7.14), and in whether or not the grouping of ODs from Section 7.6.3 is applied.

Table 7.4: Four different formulations, differing in the presence of additional constraints and their grouping of OD pairs.

	Without cons. (7.14)	With cons. (7.14)
No grouping	M1	M3
Grouping	M2	M4

We try solving the problem with the LP heuristic for each of the four methods. The solve times are summarized in Table 7.5, referring to firstly the solve time for the LP, and then the additional solve time to reach (potentially) the optimal solution. However it can be seen that only formulation M4 can solve both the LP and the subsequent IP to optimality in reasonable time. The others can all provide the LP solution but none can prove optimality for a solution in reasonable time. We allowed 5000 seconds for attempting to solve the resulting reduced IP for each model. After this time, neither M1 nor M2 both had an incumbent solution, whereas M3 had nearly found and proven an optimal solution. In fact, the line-frequencies solution provided by M3 was exactly the same as that provided by M4.

Table 7.5: LP and IP solve times for different formulations for an LP non-zero heuristic, where the IP is solved only considering non-zero variables in the LP solution. Termination with no incumbent marked 1; termination with a 0.8% gap marked 2.

	Times (s)	
Formulation	LP	IP
M1	325	5000^1
M2	42	5000^1
M3	4728	5000^2
M4	178	1086

We see that M1 and M2 do not solve to optimality or even find any feasible solutions in reasonable time. However, the comparison is potentially misleading because each is solved using the lines and frequencies found to be non-zero at LP optimality, and as we might expect (due to M1 and M2 lacking the additional constraints (7.14)) their LP solutions potentially have more non-zeros than the LP solutions for M1 and M2.

Consider Table 7.6 where we show the number of line-frequency combinations in the LP solutions to M1 and M3, and then consider expanding those as we did with integer solutions in Section 7.8.1.

From the table, it can be seen that the additional constraints (7.14) reduce the number of non-zero variables in an LP solution significantly. Also, in contrast to the LP solution without constraints (7.14), the solution with the constraints features the majority of lines at every valid frequency (due to there being only an increase from 59 to 62 line-frequency variables when the missing frequencies are included). Without the constraints, however, lines tend to occur at only a single frequency but a far greater variety of lines is present. For M1 the expansions of the problem are unlikely to give the benefit we would like; rather than resulting in a still small problem, it is expanded to almost every line, and so we would gain little over attempting to solve the entire problem.

Table 7.6: The problem size in line-frequency combinations given by taking non-zero elements of an LP, or expanding that with additional frequencies (denoted $^+$), or with neighbouring lines (denoted *)

Problem	$\sum_{l \in \mathcal{L}} \lvert \mathcal{F}_l \rvert$
M1 LP	172
M1 LP$^+$	324
M1 LP*	350
M3 LP	59
M3 LP$^+$	62
M3 LP*	156

As can be seen from the table, the LP solution when using M1 has many more non-zero elements than the LP solution of M3. The same relative difference occurs comparing M2 and M4. As the heuristic method was to then take only those non-zero elements, it is perhaps not surprising that it was difficult to solve M1 with the 172 line-frequency combinations to IP optimality, as it was to solve M4 to IP optimality with its 59 line-frequency decisions. However, as a further experiment, we instead solved M2 to LP optimality, discarded the non-zero line-frequency elements of the problem not in the LP solution, and then added constraints (7.14). In this case, with a 5000 second time limit, we were able to solve the model to within 1.3% of optimal. This solution was in fact 2.8% *worse* than the solution found with the 59 non-zero line-frequencies of the M3 LP solution. This reveals a weakness of the LP heuristic method, in that it may be possible that there are no good integer solutions given the restricted problem, and possibly no feasible integer solutions at all.

Finally, we simply attempted to solved M4 as a MIP with all lines and frequencies, which, given 5000 seconds found a solution 1.7% worse than the solution provided by M4 with the LP heuristic. Allowing significantly more time, the solver can show that the M4 LP heuristic solution is within 1% of optimal, and it finds the exact same solution itself, but not better solutions.

7.9 Results

Though we assess line plans with total passenger travel time and by line cost to the operator, there are other metrics we may use to distinguish between different line plans. We note that to be implemented in practice a real line plan must meet other requirements we have not captured, such as facilitating good timetables. We see that with our formulation of line plan requirements, examples of real line plans are not optimal for either passenger travel time or lines operator cost, or for any weighted sum of those measures. However, we might expect that a usable line plan is one which has an appropriate trade-off between those two measures and falls within certain bounds for many other metrics.

Here, we consider the following metrics for line plans described in Table 7.7.

To explore the range of values we might see for these metrics, we generate a set of solutions that we assess as being "interesting" due to either their acceptable trade-off between the costs to the operator or the passenger, or for being particularly good for some other measure. These are primarily generated iteratively, by solving the problem with different operator cost constraints and with a passenger cost objective, and storing incumbent solutions found by the solver. Good solutions were also refined as described in Section 7.8.1. For some solutions, also tighten some operational limit constraints (Constraints (7.5)) to find solutions that are more conservative but might be more likely to be operated by DSB as true line plans. Figure 7.6 shows every solution we consider, plotted by their cost to the operator and to the passenger. The majority of the considered solutions sit close to the frontier of solutions that are optimal for some weighted sum of these two measures, while a small number appear to be poor for both measures but are optimal or close to optimal for some other measure. These unusual solutions were found with a primary objective maximizing the number of direction travellers, as in Section 7.5.3. A real S-tog solution, R1, is marked with an open circle, showing that it is neither optimal for operator cost nor passenger cost, and there are solutions that are better for *both* measures. In reality, there are additional concerns we do not model that may mean real solutions are more favourable, or extra requirements that may invalidate the "better" solutions.

On Figure 7.6, the optimal solutions considering only operator cost or only passenger cost can be seen. Due to the competing nature of the two measures, the solution providing the best value for one measure is poor for the other, though the best solution for the operator is not the worst solution for the passenger, and vice versa. Let us refer to these extreme solutions as SO, the operator-optimal solution; and SP, the passenger-optimal solution. Finally, we will refer to one of the solutions that dominates R1 for both **line cost**

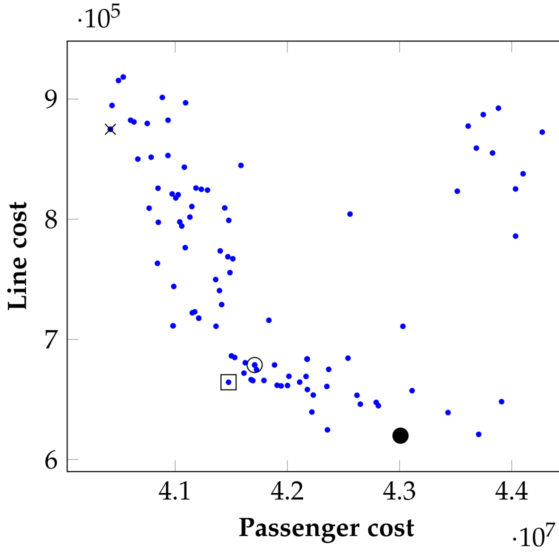


Figure 7.6: The passenger cost and operator cost for the considered set of noteworthy solutions. The open circle indicates real line plan R1.

and **passenger cost** as SM (in fact four solutions strictly dominate R1 in this solution set).

Table 7.7: The metrics used for line plan comparisons

Metric	Description
Line cost	The estimated cost to the operator, per hour, for running the line plan.
Hourly turn time	The minimum time per hour spent by trains dwelling/turning in end stations.
Hourly run time	The total moving time of trains in the line plan, per hour.
Hourly skipped stations	The number of stops that see a train pass by but not stop, per hour, in the line plan.
Unit requirement	The minimum number of units that would be required to operate the line plan (calculated using the minimum circulation time for a unit operating on the line).
Mean station visits	The average number of trains that visit a station every hour (or the total number of hourly station visits in the line plan divided by the number of stations).
Low-service stations	The number of stations which see no more than some threshold number of trains per hour. Here we use a threshold of 6.
Direct trips	The number of direct connection trips possible in the network, or the number of pairs of stations which feature together on at least one line.
Passenger cost	The total cost to the passenger, where cost is travel time with switching estimate and switching penalty.
Average travel time	The average per-person traveling component of the passenger cost, excluding switching.
Average switch cost	The average per-passenger switching cost consisting of the switching estimate and the switching fixed cost penalty.
Average switching wait	The average per-passenger time spent switching trains in the line plan.
Average switch penalty	The average per-passenger fixed switching penalty in the line plan.
Total transfers	The number of passengers who must transfer trains in the line plan.

Table 7.8: The minimum, maximum, and mean values for each metric, and the actual solution values for every metric for the four solutions: R1, SO, SP, and SM.

Indicator	Minimum	Maximum	Mean	R1	SO	SP	SM
Line cost	6.198e+05	9.184e+05	7.522e+05	6.786e+05	6.198e+05	8.748e+05	6.643e+05
Hourly turn time	3.672e+04	4.752e+04	3.869e+04	3.96e+04	3.924e+04	3.672e+04	3.816e+04
Hourly run time	2.454e+05	3.231e+05	2.71e+05	2.48e+05	2.454e+05	2.983e+05	2.563e+05
Hourly skipped stations	0	276	189.9	186	192	258	204
Unit requirement	82	108	89.82	84	82	97	85
Low-service stations	0	38	26.26	37	38	22	34
Mean station visits	33.71	53.36	41.85	37.29	36.86	44.67	41.74
Direct trips	2832	4482	3516	3132	3096	3752	3506
Passenger cost	4.042e+07	4.427e+07	4.184e+07	4.171e+07	4.301e+07	4.042e+07	4.148e+07
Average travel time	1027	1130	1050	1042	1048	1033	1038
Average switch cost	99.83	173.1	125.3	130.2	160.8	103.3	127.2
Average switching wait	45.37	107.7	68.23	66.92	99.67	50.78	73.62
Average switch penalty	42.13	72.16	57.09	63.33	61.1	52.53	53.63
Total transfers	4165	7133	5643	6260	6040	5193	5301

Table 7.8 shows the minimum, maximum, and mean values for the different measures for all solutions in the set we discuss here. The table also shows the metric values for four solutions: true solution R1, solution SO that is best for **line cost**, passenger-optimal solution SP, and the fourth SM that is better than R1 for both **line cost** and **passenger cost**.

Figure 7.7 shows normalized boxplots for all of the metrics, with the values for four noteworthy solutions marked on each. The real line plan R1 is marked on Figure 7.6 with an open circle, and we can see it is below average for **line cost** and above average for **passenger cost**. Noteworthy observations are that it has a rather low unit requirement (which is a component of the total **line cost**), but has a particularly high number of total transfering passengers compared with most other solutions, and an above average passenger (moving) travel time. In contrast, the solution SO (optimal for **line cost**) also has the lowest unit requirement of all presented solutions, but has the highest number of **low-service stations**. It also has a higher total cost to the passenger compared to solution R1 although it has fewer overall transfer passengers, and surprisingly it provides a very similar number of **direct trips** as SO. Solution SP, the passenger-optimal solution, has, perhaps unexpectedly, a very large number of **hourly skipped stations**, which corresponds to many passengers saving travel time. However this is not completely at the expense of those passengers at those skipped stations being required to transfer as the solution has a below-average (but not exceptionally low) number of transferring passengers. Finally, we mark a fourth solution which is better than R1 for both **line cost** and **passenger cost** (with an open square). We observe that it is similar to solution R1 for some metrics but surprisingly different for others; for example, it provides more **direct trips** and has fewer transferring passengers than R1, but is greater for metric **average switching wait**. The interpretation of that is that while there are fewer overall passengers transferring, those that do transfer are transferring to lines with greater headway and therefore are estimated to wait longer. The **hourly skipped stations** is indicative of a lack of homogeneity in trains; with zero **hourly skipped stations** all trans stop at every station while if this value is high, many stations are bypassed. Bypassed stations must still be served, and so there must be parallel lines running on the same infrastructure stopping at the stations, therefore stopping more and running slower. This could have implications for the robustness of the subsequent timetable; some authors [Vromans et al., 2006, for example] equate high robustness with high homogeneity in running speed. We can see that the real line plan R1 has lower **hourly skipped stations** that the other marked plans, but substantially lower values are present in other plans. The measure

hourly turn time could also be relevant for a robustness perspective, as the measure indicates the total minimum hourly utilization of turning facilities at stations, either on-platform turning or using dedicated turning track. A higher value indicates greater utilization of turning capacity and likely therefore less buffer between turning events, and subsequently less robustness for absorbing delay.

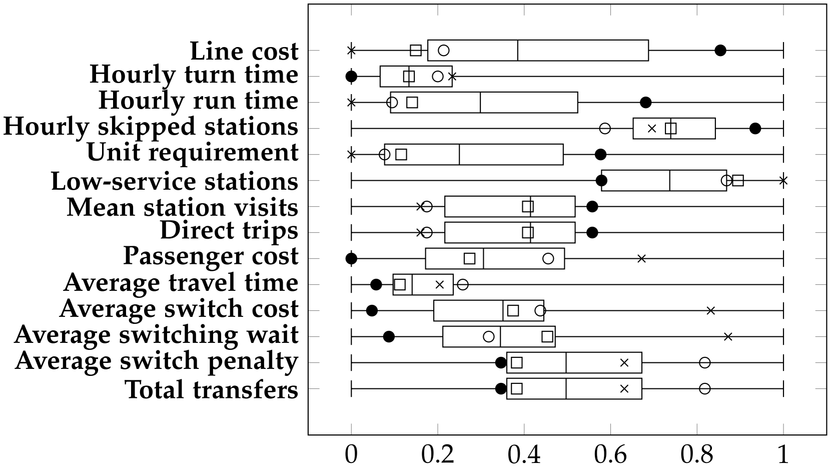


Figure 7.7: Normalized distributions of values for the performance indicators. The real line plan R1 is marked in each row with an open circle. The asterisk and the closed circle indicators indicate the optimal **line cost** and **passenger cost** solutions, respectively (SO and SP). The open square indicates a solution which dominates the real solution for those two measures.

It may be expected that a solution with many **direct trips** would also be a solution with low **passenger cost**, due to the limiting of the line-switching cost in the total passenger cost. However, that does not take into account the fact that solutions with more **direct trips** tend to have fewer missed stops, resulting in longer travel times for some passengers, and that not all line switching is equally penalised as we estimate the wait time by the line frequency. Figure 7.8 shows **passenger cost** and the number of **direct trips** for the range of solutions. Surprisingly, there is a set of solutions that have many direct trips but are also high in passenger cost. Excluding those, there may be some negative correlation between **passenger cost** and the number of **direct trips**, but this is not strong. The real solution R1 is marked, and is not particularly extraordinary for either measure. We see from the figure that simply maximizing the number of direct travellers, a measure used by some other work in the field (for example Bussieck et al. [1997]), is not appropriate for this particular problem, as those solutions with the highest number of direct travellers are particularly bad for the passenger.

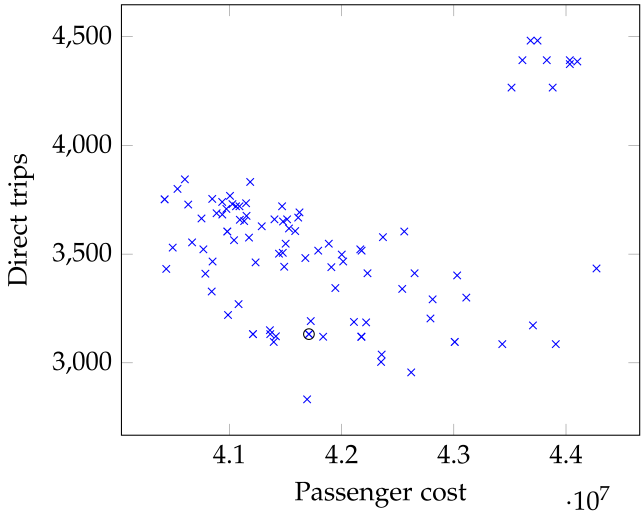


Figure 7.8: The total passenger cost and the number of direct trips for the set of solutions. The real line plan R1 is marked with an open circle.

We may consider the proportion of the total cost to the passenger that is attributable to travel time, and to switching lines. Further, we can consider the components of the cost of switching lines. That is, we consider as a fraction of the per-person passenger cost, the cost attributed to moving (**average travel time**) and the cost attributed to waiting to transfer (**average switch cost**). Of the waiting time cost, there is the estimate of the waiting time itself (**average switching wait**) and fixed penalty (**average switch penalty**). Table 7.9 shows a summary of the percentage of passenger cost that can be attributed to the different constitutive components. For these solutions the vast majority of the total passenger cost is attributable to actual moving time in trains. All passengers spend time travelling while only some must switch trains, and for many journeys with a switch, the actual travel time is greater than the switching time estimate and penalty combined, so it is unsurprising that travelling time is greater overall. We see that, for all considered solutions here, at least 85.81% of passenger cost is attributable to actual travelling in a train, while up to 14.09% is due to the cost of switching trains.

Table 7.9: Summary of the percentage of total passenger cost attributable to passenger travel time, switching cost, and the components of switching cost.

Component	Minimum	Mean	Maximum
Average travel time	85.91	89.33	91.88
Average switch cost	8.12	10.65	14.09
Average switching wait	3.92	5.79	8.77
Average switch penalty	3.40	4.86	6.16

7.10 Solving only lines model

Solutions are a valid line plan if they satisfy constraints (7.2)–(7.6), operationally and contractually, though these do not guarantee sufficient capacity for all passengers. However, the minimum-visit requirements at many stations do mean that most line plans satisfying (7.2)–(7.6) have close to sufficient capacity, or sufficient capacity, for passengers. To determine the passenger-feasibility of a solution, and the cost to the passenger, we can construct the graph described in Section 7.4.1 for only those lines in the given solution, and solve just the passenger flow problem. Alternatively, as in Section 7.8.1, we may take the lines but not their frequencies from a given solution, or construct a neighbourhood of lines, and then solve (7.1)–(7.6), and (7.7)–(7.9) to, with greater likelihood, find a feasible solution for passengers similar to the line plan given. It is possible to forbid specific solutions, either by specific line-frequency (constraint (7.15)), or by just line at any frequency (constraint (7.16)), as described in Section 7.6.2. A potential method for finding solutions is to iteratively solve the MIP (7.2)–(7.6) to optimality with some objective, apply a constraint ((7.15) or (7.16)), and repeat the process. The found solutions can be assessed in terms of passenger quality by solving the passenger flow problem, if frequencies are known, or a limited line model if the frequencies are still to be determined. It is possible that with a well-chosen objective for the reduced problem, good solutions can be found. As the limited problem of finding solutions is small (even with hundreds or thousands of additional solution-forbidding constraints), many solutions can be discovered quickly. The passenger flow problem is also relatively fast to solve for a fixed solution and so such solutions can be quickly assessed for quality and feasibility for passengers.

Of the metrics introduced in Section 7.9, the following do not depend on passenger flows:

- **Line cost**
- **Hourly turn time**
- **Hourly run time**
- **Hourly skipped stations**
- **Unit requirement**
- **Low-service stations**
- **Mean station visits**
- **Direct trips**

As we describe in Section 7.5.2 and Section 7.5.3, we can use other objective functions (**line cost** or **direct trips**, respectively). In fact we could use all of the metrics here, some more readily than others, as an objective function. However we observe that none is a good substitute for the total passenger cost objective we primarily use, and none gives solutions that are either particularly good for the passenger or that are similar to real DSB solutions. As already seen, the real DSB solutions are not extraordinary for any of the metrics we define, and so using any one metric as an objective function does not give similar solutions. Furthermore, although we can quickly find, qualify and forbid solutions and therefore assess a large number in reasonable time, we also see that there is a very large number of possible solutions to the problem. Such an approach, with the wrong objective, tends to find many solutions that are not of particular interest. An exception is when using the **line cost** objective; instead we see many infeasible solutions (for the ignored passengers), but the few that are feasible are interesting for having minimal **line cost**. However we note that when ignoring passengers, the first feasible solution is number 1305 discovered (i.e. 1304 other solutions were discovered with better **line cost** that were not feasible for the passenger). Subsequent feasible solutions occurred as solution numbers 2018, 2251, 2155 and 2274 in the first 2500 solutions discovered.

7.11 Conclusions

Here we use an arc-flow formulation to attempt to solve a line planning problem for the S-tog network, focusing on passengers. The integer programme we formulate can generally not be solved to optimality for our instances, but we can find relatively good solutions in reasonable time with an LP based heuristic method. Given more time the full formulation itself can also find good quality solutions but not generally prove optimality (without excessive additional time). We also show we can find good solutions quickly when we restrict ourselves to lines that are similar to currently-operated lines, and this is perhaps a natural restriction as it is unlikely that the operator would change all lines at once.

The passenger focus means that the lines are of good quality for the passenger and tend to be at the upper limit of whatever cost limit we enforce. By reducing the cost limit to be lower than real operated plans, we can show that there are plans which are both better for the average passenger and cheaper in line cost (though our line cost does not necessarily reflect all components of the true operating cost or other important measures). Considering some other measures, but not including them as constraints or objectives for the optimization, we can compare solutions by more than just one or two objectives, and see that there are significant differences between apparently similar solutions

We show that for this problem, the arc-flow model, though large, can be directly applied and solved to find solutions of reasonable quality, and we show a simple LP based heuristic approach to find good solutions more quickly. In our experiments we have found that the model is also applicable to the same problem but with many more frequencies per line, without becoming unsolvable. However, given that the operational requirements are created considering the given frequency options, there are fewer interesting solutions with additional frequencies.

One limitation of the model is that, while we try to minimize switching time, we can only estimate this time as we have no timetable. In fact, the lines constraints (constraints (7.1)–(7.6)) do not capture everything necessary to ensure that a timetable can be created for the line plan at all; it may be that our proposed line plans are infeasible. However, assuming that a valid timetable does exist, we can still only estimate the wait time.

Another limitation is that we penalise the cost of switching to one specific line. However, for many trips, when boarding a subsequent line on a trip, a passenger can have several similar options in some line plans. For example, a passenger may begin on line l_1 and exit at some station to wait for a train to their destination, and there are two lines l_2 and l_3 stopping at both their intermediate and destination stations, and a real passenger would likely board the first of those to arrive (even if the first train provided longer travel time). Therefore if the two lines operate at a low frequency, our long estimated wait time is pessimistic because the combined frequency of the lines is not low.

Finally, we restrict ourselves to a predetermined line pool which is not an exhaustive set of every feasible line. However the more limited number of lines ensures a reasonably sized problem, and also provides more certainty that every line in the pool is feasible (alone) because they may all be explicitly checked by an expert.

7.12 Future work and extensions

As we have said, we restrict ourselves to a predetermined line pool of 174 lines for the S-tog problem. For some different problems not expanded upon here, but, for example, finding a night-time line plan, we use a different pool of lines due to the different rules defining a feasible line and line plan. All such limited pools can be viewed as being subsets of the set of any possible line, which, due to the possibility of arbitrary stopping patterns, is very large. However, we have experimented with expanding our limited pool by considering new lines not in the original pool but with some similarity to pool lines. By fixing some of the lines in a solution, but for the non-fixed part introducing a large variety of new, out-of-pool lines, we can find line plans with some different lines relatively quickly. By performing such moves within a heuristic framework, there seems to be potential to explore all possible lines in a reasonable way that is not possible with the MIP formulation we present.

The line plan solutions we find may possibly, but not necessarily, be operated in practice by DSB. Despite meeting all operational and contractual requirements, and not containing any explicitly forbidden line pairs which cause problems for creating a timetable, it can still be the case that it is not possible to create a feasible timetable for a line plan. Even if there are feasible timetables for a line plan, there may be no *good* timetables for the line plan; the measures used to compare timetables are not necessarily the same as those we estimate for the line plans (operator cost and passenger cost) and may include other measures such as minimal headways. On the other hand, our estimated operator cost and estimated passenger waiting times can be more precisely assessed when the timetable is determined. One direction for future work is more closely integrating the creation of a line plan with the subsequent timetable problem that takes a line plan as input. This would avoid the risk of finding infeasible line plans (from the point of view of timetable creation), and could allow more precise operator cost and waiting time estimates. Looking further in the planning process, the timetable itself may not facilitate the creation of good or even feasible rolling stock plans. Closer integration of all planning stages of rail is obviously desirable to achieve overall optimality of plans, but the complexities of each problem alone prove to be a challenge, and so complete integration of every problem stage is unlikely.

7.13 Bibliography

Ralf Borndörfer, Martin Grötschel, and Marc E Pfetsch. A column-generation approach to line planning in public transport. *Transportation Science*, 41(1):123–132. 2007.

Michael R Bussieck, Peter Kreuzer, and Uwe T Zimmermann. Optimal lines for railway systems. *European Journal of Operational Research*, 96(1):54–63. 1997.

Marc Goerigk, Michael Schachtebeck, and Anita Schöbel. Evaluating line concepts using travel times and robustness. *Public Transport*, 5(3):267–284. 2013.

Jan-Willem Goossens, Stan Van Hoesel, and Leo Kroon. A branch-and-cut approach for solving railway line-planning problems. *Transportation Science*, 38(3):379–393. 2004.

Jan-Willem Goossens, Stan van Hoesel, and Leo Kroon. On solving multi-type railway line planning problems. *European Journal of Operational Research*, 168(2):403–424. 2006.

Karl Nachtigall and Karl Jerosch. Simultaneous network line planning and traffic assignment. In *ATMOS*, page 105. 2008.

Natalia J Rezanova. Line planning optimization at DSB. In *13th Conference on Advanced Systems in Public Transport*. 2015.

Anita Schöbel. Line planning in public transportation: models and methods. *OR spectrum*, 34(3):491–510. 2012.

Anita Schöbel and Susanne Scholl. Line planning with minimal traveling time. In *ATMOS 2005-5th Workshop on Algorithmic Methods and Models for Optimization of Railways*. Internationales Begegnungs-und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl. 2006.

Michiel JCM Vromans, Rommert Dekker, and Leo G Kroon. Reliability and heterogeneity of railway services. *European Journal of Operational Research*, 172(2):647–665. 2006.

Chapter 8

Integrating Robust Timetabling in Line Plan Optimization for Railway Systems

With Sofie Burggraeve[1]
and Peter Vansteenwegen[2]
and Richard M Lusby. [3]
Submitted to Transportation Research Part C.

[1]University of Leuven. sofie.burggraeve@kuleuven.be
[2]University of Leuven. pieter.vansteenwegen@kuleuven.be
[3]Technical University of Denmark. rmlu@dtu.dk

Abstract. *We propose a heuristic algorithm to build a railway line plan from scratch that minimizes passenger travel time and operator cost and for which a feasible and robust timetable exists. A line planning module and a timetabling module work iteratively and interactively. The line planning module creates an initial line plan. The timetabling module evaluates the line plan and identifies a critical line based on minimum buffer times between train pairs. The line planning module proposes a new line plan in which the time length of the critical line is modified in order to provide more flexibility in the schedule. This flexibility is used during timetabling to improve the robustness of the railway system. The algorithm is validated on a high frequency railway system with little shunt capacity. While the operator and passenger cost remain close to those of the initially and (for these costs) optimally built line plan, the timetable corresponding to the finally developed robust line plan significantly improves the minimum buffer time, and thus the robustness, in eight out of ten studied cases.*

Keywords: railway line planning; timetabling; robustness; mixed integer linear programming.

8.1 Introduction

Railway line planning is the problem of constructing a set of lines in a railway network that meet some particular requirements. A line is often taken to be a route in a high-level infrastructure graph ignoring precise details of platforms, junctions, etc. In our case, a line is a route in the network together with a stopping pattern for the stations along that route, as a line may either stop at or bypass a station on its route (which saves time for bypassing passengers). We define a line plan as a set of such routes, each with a stopping pattern and frequency, which together must meet certain targets such as providing a minimal service at every station.

Timetabling is the problem of assigning precise utilization times for infrastructure resources to every train in the rail system. These times must ensure that trains can follow their routes in the network, stop at appropriate stations where necessary, and avoid any conflicts with other trains. A conflict rises where two trains want to reserve the same part of the infrastructure at the same time, for example at a switch, platform or turning track. If timetabling is performed separately from line planning, the line plan specifies the lines and the number of hourly trains operating on each line but not the exact times for those trains and not the precise resources that a train on a line will utilize. Those timings and utilizations are decided as part of the timetabling.

Traditionally, a railway line plan is constructed before a timetable is made. However, an optimal line plan does not guarantee an optimal or even a feasible timetable [Kaspi and Raviv, 2013]. An integrated approach can overcome this problem. Nevertheless, since line planning and timetabling are both separately already very complex problems for large railway networks [Michaelis and Schöbel, 2009, Goerigk et al., 2013], solving the resulting integrated problem is in most cases not computationally possible [Schöbel, 2015]. We propose a heuristic algorithm that constructs a line plan for which a feasible timetable exists. We call a line plan *timetable-feasible* if there exists a conflict-free timetable for that line plan. Moreover the algorithm improves the robustness of the line plan by making well chosen changes in the stopping patterns of the lines while the existence of a feasible timetable remains assured.

There are different interpretations of robustness in railway research. According to Dewilde et al. [2011], a railway planning is *passenger robust* if the total travel time in practice of all passengers is minimized in case of frequently occurring small delays. The focus of this definition is twofold, as both short and reliable travel times have to be provided by the planning. Passenger robustness is also what we want to strive for with our approach.

The context of this research is a high frequency network where trains are forced to turn on their platform in their terminal stations due to a lack of shunting area. The proposed integrated approach originates from insights on why some line plans do not allow feasible timetables and why some line plans allow more robust timetables. A first insight is that a line can be infeasible on its own, which we call *line infeasibility*. A second insight is that line combinations can be infeasible due to their frequencies. We call this *frequency combination infeasibility*. In Section 8.3 we explain these insights. Furthermore, we present a technique to develop a line plan that guarantees a feasible timetable. We introduce a timetabling model based on the Periodic Event Scheduling Problem (PESP) to create passenger robust timetables. We illustrate with a case study that a smart and targeted interaction of both techniques develops a line plan from scratch which guarantees a feasible and passenger robust timetable. Moreover, the integrated approach can also be used to improve the robustness of an existing line plan. The line planning and timetabling technique and the integrated approach are explained in Section 8.4.

Related work is discussed initially in Section 8.2. The case study is described in more detail in Section 8.5. In Section 8.6 the results of the case study are presented and examined and the integrated approach is illustrated in an example. The paper is concluded and ideas for future research are suggested in Section 8.7.

8.2 State of the art

The planning of a railway system consists of several decisions on different planning horizons [Lusby et al., 2011]. The construction of railway infrastructure and a line planning are long term decisions. A timetable, a routing plan, a rolling stock schedule and a crew schedule are made several months up to a couple of years in advance. Decisions on handling delays and obstructions in daily operation are made in real time. Each of these decisions affects the performance of the other decisions. Ideally, a model that optimizes all these decisions simultaneously is preferred. Each of the separate decision problems, however, is NP-hard for realistic networks [Schöbel, 2015]. In practice these planning decisions are usually made one after the other, although the solution from a previous decision level problem does not even guarantee that a feasible solution exists for the next level problem [Schöbel, 2015]. In the case that the output of the previous decision level leads to infeasibility at the next planning step, there are several possible approaches for looking for a feasible solution to both planning levels together. First, the outcome of the previous level can be replaced by a second best outcome in the hope that a feasible solution for the next level exists. Secondly, the outcome of the previous level can be specifically oriented towards making a feasible solution for the next level possible, by using case dependent restrictions specifically for this goal. Thirdly, the constraints on the outcome of the next level can be loosened. These approaches increase the possibility of finding a feasible solution for the next level, but not necessarily guarantee a good outcome for both levels together. A few integrated approaches for two or three of the typical decision problems in railway research are described in the literature and clearly outperform the hierarchical approach [Goerigk et al., 2013]. Most of these solution algorithms are heuristics to overcome the high computation times of an exact approach for a realistic railway network. As in this paper we propose an algorithm towards the integration of line planning and timetabling, we elaborate on existing integrated approaches for these two planning problems in the first part of this literature review. We also explain the place of the individual timetabling and line planning modules that are used in our integrated approach within existing literature on timetabling and line planning.

8.2.1 Integration of line planning and timetabling

This paper is not the first attempt towards an integration of line planning and timetabling in railway scheduling. In Liebchen and Möhring [2007], some line planning decisions are included in the timetabling process. They assume that, for some parts (sequence of tracks) of the network, the number of lines serving each part is known beforehand. On these track sections they put an artificial station in the middle. Every line along this track section is then partitioned into two line segments, before and after the artificial station. They use the Periodic Event Scheduling Problem (PESP) which was introduced by Serafini and Ukovich [1989] to model the timetabling problem in which they add constraints such that a perfect matching between the arriving and the departing line segments is forced. This is achieved by matching arrival and departure times of the line segments in the artificial station which are assigned by this same model. Here one line corresponds to one train. This approach is deficient if, for some network parts, the number of passing trains is not known beforehand. This is often the case in real world networks. Furthermore, this approach can lead to a set of lines in which each line is a different combination of track sections which is not transparent for the passengers.

Kaspi and Raviv [2013] present a genetic algorithm that builds a line plan and timetable from scratch. They start from a given line pool and per line a fixed number of potential trains. A solution consists of three characteristics for each train: the value zero or one, which indicates if the train should be scheduled or not, an earliest start time and a stopping pattern. A member of the initial population is constructed by drawing values for each characteristic from separate Bernoulli distributions. The timetable and line plan are constructed by scheduling trains with value one for the first characteristic according to a fixed priority rule. If a train cannot be scheduled without one or more conflicts with other already scheduled trains, this train is omitted from the solution. For the resulting timetable, the passenger travel time and the operator cost are calculated. These performance results affect the distribution parameters of the Bernoulli distributions from which the next generation will be drawn. This approach uses the performance of the timetable as input for the line planning of the next iteration. This interaction between line planning and timetabling is also the case in our approach. But in contrast to the stochastic approach of Kaspi and Raviv [2013], we use information of the timetable to make some deterministic and tactical changes to the line planning. Also in Goerigk et al. [2013] timetable performance is used to evaluate line plans. However, they do not iterate between the construction phase of the line planning and the timetabling, and they do not use this information to improve the line planning. They only use it to compare different ways to construct a line plan.

Michaelis and Schöbel [2009] offer an integrated approach in which they re-order the classic sequence of line planning, timetabling and vehicle scheduling for bus planning. The different planning steps are, however, performed one after each other such that the approach is still sequential. Vehicle scheduling or rolling stock scheduling are not integrated in our approach, but we take turn restrictions in the end stations into account which significantly simplify the rolling stock scheduling. Taking turn restrictions into account is useful if terminal stations are not equipped with enough shunting space for efficient turning during daily operation. In fact, neglecting turn restrictions can lead to infeasible timetables. To the best of our knowledge, no other integrated approach for timetabling and line planning takes turn restrictions during daily operation into account. This is explained in the next section.

Very recently, Schöbel [2015] published a mixed integer linear program (MILP) in which line planning and timetabling are integrated for railway planning. This model is based on the PESP of Serafini and Ukovich [1989]. In the model, binary variables are introduced to indicate if a certain line is added to the line plan. There are also big M-constraints added to the PESP model in which these binary variables are used to push event times of lines which are not in the line plan to zero and also to switch off lower bounds of activities for unassigned lines. The objective function minimizes the planned travel time of the passengers. Transfer penalties are not taken into account, but they can easily be introduced as a weight in the objective function. No performance results of this model are presented yet.

An added value of our approach is that robustness is taken into account when constructing a line plan (and timetable). With our approach we want to shift the focus in current research on integration of line planning and timetabling to the creation of passenger robust line plans (and timetables). The algorithm that we propose constructs a line plan that minimizes planned passenger travel time and operator costs but also prevents unreliable travel times during daily operation in order to provide a short travel time in practice for all passengers.

As mentioned in the introduction, a *passenger robust* plan minimizes this total travel time in practice. In order to obtain short travel times in practice, the propagation of delays from one train to another train has to be avoided. This can be achieved if the line plan allows a timetable in which the buffer times between trains are above a certain threshold. Also in Kroon et al. [2008], Caimi et al. [2012], Salido et al. [2012], Dewilde et al. [2013], Sels et al. [2016] and Vansteenwegen et al. [2016] the (minimum) buffer times between train pairs are lengthened in order to reduce the propagation of delays.

Another added value of our approach is that trains with the same route are equally spread over the period of the cyclic timetable. Making the reasonable assumption that passengers arrive uniformly in a station of a high frequency network, a constant time interval between two trains with the same route minimizes the average waiting time of the passengers before boarding.

In our heuristic approach, a line planning and timetable module alternate, where each consists of an exact optimization model. We now motivate our choice for the timetable and line planning models that are used and briefly discuss related literature.

8.2.2 Timetabling

The goal of the timetabling module is to construct a passenger robust timetable. This avoids propagation of delays in case of small delays during daily operation in order to provide reliable travel times to the passengers and is achieved by maximizing the (minimum) buffer times between train pairs. Parbo et al. [2015] give an extensive overview of passenger perspectives in railway timetabling. The PESP model of Serafini and Ukovich [1989] is the foundation of many timetable models [Schrijver and Steenbeek, 1993, Nachtigall, 1996, Liebchen, 2006, Peeters, 2003] and [Großmann, 2011] and is also the framework of our timetabling model. The PESP model schedules events in a period of the cyclic timetable and takes precedence constraints and relations between events into account. Arrivals and departures of trains at stations or reservations and releases of track sections are events. If two events are related or can affect each other they form an activity. Examples of activities are the arrival and departure of the same train in a station or the reservation times of a shared switch or platform by two different trains. The PESP model constrains each activity time, which is the time between the two events that define the activity. The PESP is originally defined without an objective function, but several objective functions for PESP can be found in the literature. We add an objective function that maximizes the (minimum) buffer times between trains using the same part of the infrastructure, in order to achieve robustness. In our timetabling model, we also have ‘turning’ and ‘providing buffer time’ as activities between events besides the usual driving, waiting and transferring activities. Furthermore, we include extra constraints such that trains can be equally spread over the hour. A recent and elaborate discussion on timetable literature in general and PESP in specific can be found in Sels et al. [2016].

8.2.3 Line planning

Railway line planning is, generally, the construction of a set of lines to operate in a rail network. There are parallels to line planning problems in bus network design and network design for liner shipping. Line planning for rail takes the physical rail network as a fixed input, and provides a fixed input to subsequent timetabling and rolling stock planning. So when creating the line plan, assumptions can potentially be made about the form or characteristics of timetables, rolling stock and rolling stock planning. Schöbel [2012] gives an overview of different approaches to model and solve the line planning problem, broadly categorizing line planning approaches that are (operator) cost-oriented, and those that are passenger-oriented.

Goossens et al. [2006] focus on minimizing operator cost, for the less-studied case of line planning where lines may not stop at every station. Also in our research the stopping pattern of a line is decided in the line planning problem. The advantage of allowing lines to skip stations is the potential to combine fast lines which only stop at the stations with high demand and slow lines which also stop in stations with low demand (with the classification of stations not specified but decided during line planning). Using fast lines shortens the travel time of a lot of passengers and the slow lines assure a service in every station.

With a passenger focus, a common objective function is to maximize the number of direct travelers, i.e. the number of passengers who have a route from their origin to destination that does not require transfers. The simplest interpretation of this objective is to count the number of passengers for which there exists a line in the solution visiting both their origin and destination. This does not actually find passenger routes and does not guarantee that all counted passengers can actually *use* the line, as there may be insufficient capacity on some lines. Using this objective also has the risk in some networks that the passengers with no direct route may be faced with many transfers. Another disadvantage is that maximizing the number of direct travelers encourages long train lines and, critically in our case, does not favour skipped stations. Bussieck et al. [1997] is one example which uses this direct traveler objective, while ensuring that direct lines also have sufficient capacity to accommodate the passengers.

Another objective function with passenger focus is a travel time objective that takes into account the passenger's time traveling in trains and a penalty for switching trains (transfers). The calculation of this objective requires knowledge on the routing of passengers in the network taking into account travel time and switching. This routing of the passengers can be modelled in a graph, which can become very large due to the large number of passenger routes required (potentially one for every pair of stations). Schöbel and Scholl [2006] and Borndörfer et al. [2007] are examples where passengers between a pair of stations are routed by minimizing the sum of the travel time costs of the used paths. This passenger routing objective could be used as part of a weighted sum objective along with some operator cost [Borndörfer et al., 2007], or used alone but with an additional operator cost budget constraint [Schöbel and Scholl, 2006]. In some practical problems the inclusion of a budget can be very important when combined with a passenger-oriented objective, as without it, solutions can contain many lines to individually satisfy every type of passenger. Our line plan model uses also the passenger's travel time objective. In our case study, however, there are tight rate limits on the maximum number of trains turning at a terminal station and on the use of certain infrastructure. Thus even without an operator budget consideration we do not risk solutions having particularly many lines.

Operator focused or passenger focused is a first partitioning that can be made. Another partitioning is that a line planning model may be based on a predetermined set of lines (a line pool), or it may find new lines dynamically. An advantage of a predetermined line pool is that all lines in the pool can be guaranteed to be feasible in terms of line planning requirements, or advantageously for our case, in terms of timetabling requirements. This latter is explained in the next section. A predetermined pool also has the advantage of limiting the problem size in a useful and dynamic way (because the pool can be limited to be as diverse or as focused as desired). However, it has the disadvantage that the full set of possible lines may be very large and so enumerating them all could be intractable, while taking only a subset of all possible lines risks missing good solutions. Schöbel and Scholl [2006] present a model that takes as input a predetermined pool of lines. In contrast, Borndörfer et al. [2007] present a method where lines are generated dynamically in an infrastructure network as a pricing problem, finding maximum-weighted paths to introduce as lines to a restricted master problem. However, the master problem itself is formulated in terms of a known line pool.

With respect to decision variables, many approaches are similar in using either a binary decision for the presences of each line, or a non-negative or integral decision for the frequency of each line, where a frequency of zero means that the line is not in the solution. In our approach we may only select one of a set of frequencies defined individually for every line, so our model uses a binary decision variable indicating the presence of a (line, frequency)-pair.

Specifically related to the S-tog problem at DSB (which we will address as a case study), Rezanova [2015] solves the line planning problem with an operator focus, considering train driving time and a particular competing objective related to new regulations for drivers. The author notes the problem of finding line plan solutions that are not feasible for timetabling, and suggests that an integrated approach would be valuable.

8.3 Timetable-infeasibility

In this section we explain how limited shunt capacity and frequency combinations of lines that share a part of the network can lead to timetable-infeasibility of line plans. Our integrated approach uses these insights to construct line plans that allow conflict-free and passenger robust timetables.

8.3.1 Line infeasibility

Consider Figure 8.1, showing a single train line operating at six times per hour. The black dots on the time-axis show the scheduled departures from the beginning station for this line, which is once every ten minutes. We illustrate the first two time-distance graphs; the first departing from the beginning

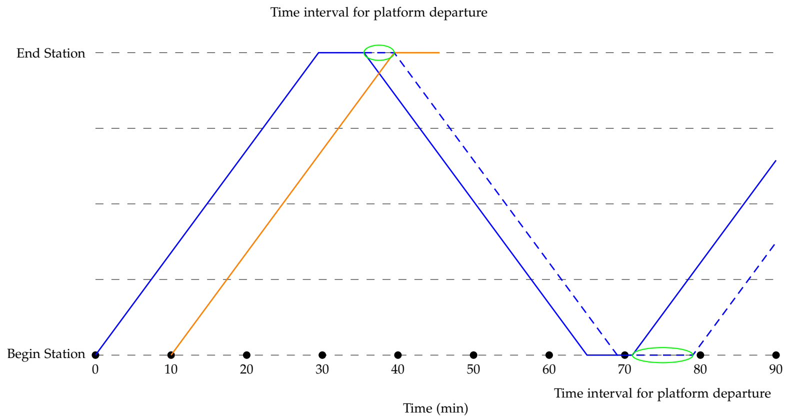


Figure 8.1: A line can be infeasible on its own

station at minute zero (solid blue line), and the subsequent train following with a departure at minute ten (red line). In this example, the travel time between the beginning and ending stations for the line is 29 minutes and we assume that the train has to turn on its platform due to restricted shunt capacity in its terminal stations. We define a minimum dwell time of seven minutes before a train can leave its platform in the other direction. This dwell time is the time required by passengers to leave the train, by the driver to change to the driver position at the other end of the train and by new passengers to board the train before a departure returning to the beginning station. Note that the subsequent train that departed ten minutes later is therefore arriving at the ending station ten minutes later as well, so the first train has a well-defined latest departure which is marked as a dashed blue line. The train then drives for 29 minutes back to its beginning station, arriving there between 65 minutes and 68 minutes after its first departure at minute zero. It can leave for the next round trip at 72 minutes after minute zero at the earliest (minute 65 arrival with seven minutes minimum required turn time) and 78 minutes at the latest (68 minute arrival with a maximum of ten minutes dwell time, assuming that the next train arrives ten minutes later on the same platform). However, no train is planned to leave the begin station in the interval of 72 to 78 minutes, which can be seen in Figure 8.1 as no black dot falls in the interval indicated with the green line. Therefore no feasible timetable can be found for this line. We will call this *line infeasibility*.

This insight can be mathematically formulated as: *In case of restricted shunt capacity in its terminal stations, a line is infeasible on its own if there exists no*

$k \in \mathbb{Z}^+$ such that

$$2 * dr_{l,s_{l,0},s_{l,e}} + \mathrm{ntt}_{s_{l,0}} + \mathrm{ntt}_{s_{l,e}} \leq \frac{P}{f_l} * k \leq 2 * dr_{l,s_{l,0},s_{l,e}} + 2 * \frac{P}{f_l},$$

where $dr_{l,s_{l,0},s_{l,e}}$ is the planned travel time between the start station $s_{l,0}$ and the end station $s_{l,e}$ of line l. Further $\mathrm{ntt}_{s_{l,0}}$ and $\mathrm{ntt}_{s_{l,e}}$ are respectively the necessary turn time for line l in its start and end station, f_l is the frequency of line l, P is the period of the cyclic timetable and trains of the same line are equally spread over the period and use the same platform in the terminal stations for passenger convenience.

8.3.2 Frequency combination infeasibility

Suppose that different lines share a part of the network and that trains of the same line are equally spread in the cyclic timetable. A second insight is that the frequencies of these lines affect the potential buffer time between these lines. Let $f_{l_1} \leq f_{l_2}$ be the frequencies of two lines l_1 and l_2 respectively. It is straightforward that the higher the frequencies the smaller the potential buffer time between trains of these lines. But we also claim that the buffer time between a line at a higher frequency and a lower frequency is no greater than between two lines at the higher frequency. For example, assume $f_{l_1} = 4$ and $f_{l_2} = 5$, then in one cycle of the timetable, there will be, at some point, two trains of line l_2 which are planned between two succeeding trains of l_1. The time between two trains of l_1 is 15 minutes and between two trains of l_2 is 12 minutes. This would lead to the situation in Figure 8.2. Because a is strictly smaller than three, the smallest buffer between a train of line l_1 and line l_2 is smaller than or equal to one-and-a-half minutes.

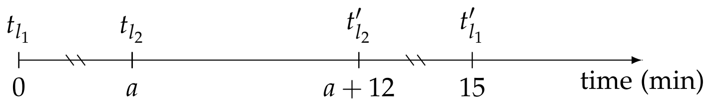


Figure 8.2: If lines l_1 and l_2 have frequencies $f_{l_1} = 4$ and $f_{l_2} = 5$ respectively, then once in 60 minutes two trains (t_{l_2} and t'_{l_2}) of line l_2 will pass in between two trains (t_{l_1} and t'_{l_1}) of line l_1. Without loss of generality we can assume that this happens in the first quarter. Here $a \in \mathbb{R}$ and $0 < a < 3$.

This can be mathematically generalized and formulated as: *The minimum time between a passage of a train of line l_1 and line l_2 with frequencies $f_{l_1} \leq f_{l_2}$ respectively* *is smaller than ($\leq$)*

$$\frac{\frac{P}{f_{l_1}} - (\lceil \frac{f_{l_2}}{f_{l_1}} \rceil - 1)\frac{P}{f_{l_2}}}{2} \tag{8.1}$$

where P is the period of the cyclic timetable and $\lceil a \rceil$ is the smallest integer b with $b \geq a$.

If this minimum buffer time is strictly smaller than the minimum necessary buffer time according to safety regulations in the network, then l_1 with frequency f_{l_1} and l_2 with frequency f_{l_2} are not feasible together. In the example, if the minimum necessary buffer time according to safety regulations is two minutes, then these lines l_1 and l_2 cannot be combined at frequencies $f_{l_1} = 4$ and $f_{l_2} = 5$. Note that formula (8.1) is maximal in case f_{l_2} equals f_{l_1} or is a multiple of f_{l_1} and then reduces to $\frac{P}{2f_{l_2}}$ which proves our claim.

8.4 Methodology

In this section, we propose an integrated approach that constructs a line plan from scratch that minimizes a weighted sum of operator and passenger cost and allows a feasible and robust timetable. First a timetable-feasible line plan is constructed. Then, iteratively and interactively, a line planning module produces a line plan, and for the plan, a timetable module produces a timetable that maximizes the (minimum) buffer times between train pairs. Each loop an analysis of the timetable indicates how the line plan could be adapted in order to allow a more robust timetable. This adaptation increases the flexibility of the line plan which is used, in the timetabling module, to increase the minimum buffer times between train pairs. The line plan module then calculates the new line plan that includes this adaptation while minimizing the weighted sum of operator and passenger costs. This feedback loop stops when there is no further improvement possible or if there is no improvement during a fixed number of iterations for the minimum buffer times between train pairs. We first discuss the line planning module and the timetabling module separately and then the integration of both. Both the timetable and the line planning module consist of an exact optimization model, though our combined approach, and the fact that we do not always solve the models to optimality, result in an overall heuristic method.

8.4.1 Line planning module

Constructing a line plan consists of selecting a set of lines which meet certain requirements from a pool of predetermined lines. The line pool is not exhaustive; there are many more possible lines than those considered, but the set is reduced to those that meet certain criteria as discussed with the rail operator. This also keeps the problem size small. The model performs three functions: (i) selecting the lines and frequencies and creating a valid plan, (ii) routing passengers between origin and destination stations and (iii) relating passenger routes to line selections.

Let us first define the set of all lines available: $\mathcal{L}$. For every line $l \in \mathcal{L}$ we define a set of valid frequencies for the line: $\mathcal{F}_l$. The operator must meet certain obligations for any valid line plan and must not exceed certain operational limits. These restrictions are referred to as service constraints. We define these all in terms of a set of resources $\mathcal{R}$, and define all limitations as either a minimum (rmin_r) or maximum (rmax_r) number of trains using that resource $r \in \mathcal{R}$ every hour. The subset of lines that make use of resource $r \in \mathcal{R}$ is defined as $\mathcal{L}_r$. Let $c_{l,f}$ be the cost to the operator for operating line l at frequency f.

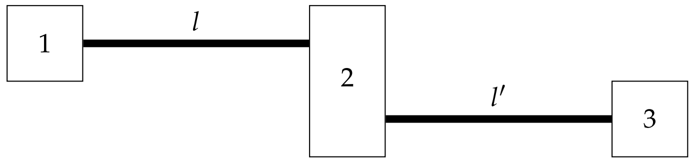
The line planning module starts from a known origin-destination (OD) matrix containing the passenger demand for travel between every origin and destination, where origins and destinations are simply stations in the rail network. Let $\mathcal{S}$ be the set of stations. For two stations $s_1, s_2 \in \mathcal{S}$ we know the demand d_{s_1,s_2}. We model passengers as a flow from each origin station to every relevant destination station in a graph constructed for the network and all lines in the pool. This graph captures the passenger cost in terms of travel time on lines and estimated waiting time between lines (estimated by frequency) in case a transfer is required. We refer to this graph as the passenger graph. The graph contains a *(line, frequency, station)* vertex for every line, frequency, and every station visited by that line. The edges of this graph represent travel possibilities, with the edge cost being the known train driving time or the estimated transfer time. Additionally, this graph contains source *(r_s)* and sink *(t_s)* vertices for every station s where passengers originate from or terminate their travel. These vertices are connected to the appropriate *(line, frequency, station)* vertices with edges representing boarding or alighting from a line. These edges have zero cost. Finally, for every station s we have a platform vertex *(p_s)* with edges from and to every *(line, frequency, s)* vertex, where the costs correspond to an estimate of perceived waiting time which consists of a fixed penalty component and a frequency-dependent component.

Let $\mathcal{V}$ and $\mathcal{E}$ be the set of all vertices and edges of this graph, respectively, and t_e be the cost to a single passenger of using edge $e \in \mathcal{E}$. In total there are five types of edges.

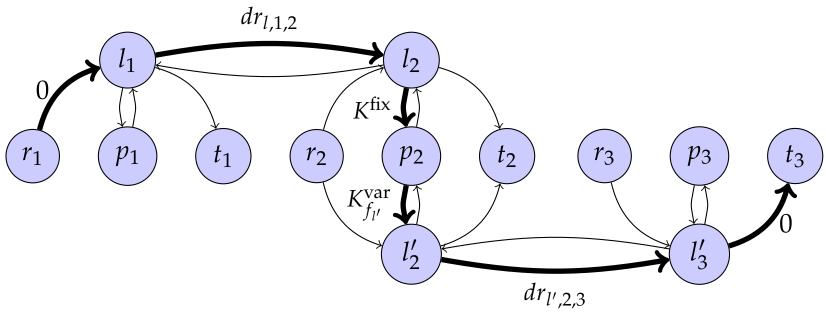
- Type 1. From (l, f, s) to (l, f, s') for all lines $l \in \mathcal{L}$, $f \in \mathcal{F}_l$ and s and s' two succeeding stations visited by line l.
- Type 2. From (l, f, s) to (p_s) for all lines $l \in \mathcal{L}$, $f \in \mathcal{F}_l$ and s a station visited by line l and (p_s) the platform vertex of station s.
- Type 3. From (p_s) to (l, f, s) for all lines $l \in \mathcal{L}$, $f \in \mathcal{F}_l$ and s a station visited by line l and (p_s) the platform vertex of station s.
- Type 4. From (r_s) to (l, f, s) for all lines $l \in \mathcal{L}$, $f \in \mathcal{F}_l$ and s a station visited by line l and (r_s) the source vertex of station s.
- Type 5. From (l, f, s) to (t_s) for all lines $l \in \mathcal{L}$, $f \in \mathcal{F}_l$ and s a station visited by line l and (t_s) the sink vertex of station s.

See Figure 8.3 for an example of a simple network with three stations, 1, 2 and 3, and two lines l and l' visiting two of the stations each (8.3a) and corresponding passenger graph (8.3b).

This graph is similar to the *change&go* graph of Schöbel and Scholl [2006], but distinguishes between line transfers that in our case happen to lines with discrete frequencies, with a frequency-dependent cost.



(a) Line structure (two lines, three stations)



(b) Corresponding passenger graph structure

Figure 8.3: The upper figure shows a simple network with three stations 1, 2 and 3, and two lines l and l'. Line l visits stations 1 and 2, and line l' visits stations 2 and 3. Each line operates at just a single frequency. The lower figure shows the subsequent passenger graph structure used for this network. Costs are labelled on the edges for a passenger travelling from station 1 to station 3, transferring lines at station 2, with used edges in bold. The costs to the passenger are $dr_{l,1,2}$, travelling (**dr**iving) on line l from station 1 to 2; fixed cost K^{fix} for a transfer and an additional $K^{\text{var}}_{f_{l'}}$ frequency dependent cost for transferring to line l'; and $dr_{l',2,3}$ travelling on line l' from station 2 to 3.

Let l_e be the line that edge e is related to and f_e be the frequency of the line that e is related to. This line and frequency of an edge are uniquely defined as the two vertices connected by edge e are either both related to the same line and frequency or only one of them is related to a line and frequency.

Let a^s_v be the demand for passengers originating from station s at vertex v in the passenger graph. For vertex types *(line, frequency, station)*, $a^s_v = 0$. For a given station s_1, there is a single source vertex v corresponding to s_1. For this vertex, and for passengers originating from that station, $a^{s_1}_v = -\sum_{s_2 \in \mathcal{S}} d_{s_1,s_2}$. For a given station s_2, there is also a single sink vertex corresponding to s_2. For every sink vertex v the demand of passengers to that vertex from station s_1 is the demand of passengers from s_1 to s_2: $a^{s_1}_v = d_{s_1,s_2}$.

For relating passengers to lines, let C_f be the passenger capacity of any line operating at frequency f. We are therefore assuming the same rolling stock unit type and sequence for every line, but a higher frequency provides more seats than a lower frequency. We require that no more passengers use a line as the line capacity permits for the frequency the line is operating at.

We use two classes of decision variables: $x_{l,f} \in \{0,1\}$ is a binary decision variable indicating whether or not line l is selected at frequency f, and y_s^e decides the number of passengers from origin station s that use edge e in the passenger graph.

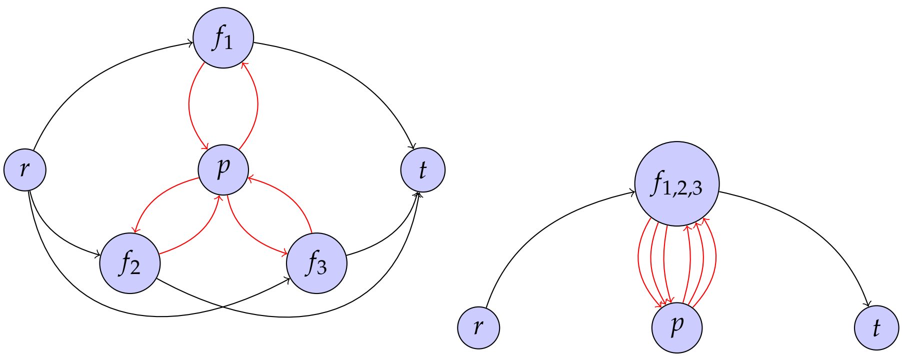
The line planning model is:

$$\begin{aligned}
\text{Minimize} \quad & \lambda \sum_{l \in \mathcal{L}} \sum_{f \in \mathcal{F}_l} c_{l,f} x_{l,f} + (1-\lambda) \sum_{e \in \mathcal{E}} \sum_{s \in \mathcal{S}} t_e y_s^e && && (8.2)\\
\text{s.t.} \quad & \sum_{f \in \mathcal{F}_l} x_{l,f} \leq 1 && \forall l \in \mathcal{L} && (8.3)\\
& \sum_{l \in \mathcal{L}_r} \sum_{f \in \mathcal{F}_l} f x_{l,f} \leq \mathrm{rmin}_r && \forall r \in \mathcal{R} && (8.4)\\
& \sum_{l \in \mathcal{L}_r} \sum_{f \in \mathcal{F}_l} f x_{l,f} \geq \mathrm{rmax}_r && \forall r \in \mathcal{R} && (8.5)\\
& \sum_{(u,v) \in \mathcal{E}} y_s^{(u,v)} - \sum_{(v,w) \in \mathcal{E}} y_s^{(v,w)} = a_v^s && \forall s \in \mathcal{S}, \forall v \in \mathcal{V} && (8.6)\\
& \sum_{s \in \mathcal{S}} y_s^e \leq C_f x_{l_e,f_e} && \forall e \in \mathcal{E} && (8.7)\\
& x_{l,f} \in \{0,1\} && \forall l \in \mathcal{L}, f \in \mathcal{F}_l && (8.8)\\
& y_s^e \in \mathbb{R}^+ && \forall s \in \mathcal{S}, e \in \mathcal{E} && (8.9)
\end{aligned}$$

The objective function (8.2) is a weighted sum of the operator cost and the passenger travel time (ride time and wait time), using a parameter $\lambda \in [0,1]$ to determine the importance of one component over the other.

Constraints (8.3) ensure that a line is chosen with at most one frequency (i.e. combinations of frequencies are not permitted, as if valid a discrete frequency would be present in the frequency set $\mathcal{F}_l$ for the line). Constraints (8.4) and (8.5) ensure that the obligatory and operational requirements are met for the line plan. Constraints (8.6) consist of the flow conservation constraints. The number of passengers leaving from an origin station must flow from that station with the appropriate number arriving at every destination station, such that flow is conserved. Constraints (8.7) link the flows of passengers to the line decisions. The presence of a positive passenger flow on an edge in the graph is dependent on some line being present in the plan. The maximum flow on that edge depends on the passenger capacity of the corresponding line at the appropriate frequency. Finally, constraints (8.8) and (8.9) restrict the line variables and flow variables to be binary variables and positive otherwise unrestricted variables, respectively.

The presented model is very large, as the passenger graph we construct is very large and there are therefore a large number of flow variables and corresponding constraints. However, we observe that many of the vertices and edges in the graph are very similar and differ only in line frequency. For lines with many possible frequencies there is significant duplication. For the edges related to switching lines at a station, frequency is required to determine the cost to the passenger. However for all other edges the frequency information is redundant. Indeed, the cost of travelling on a line between stations does not depend on the frequency of that line. A first simplification of the model is that for each line and its frequencies, we replace the edges (and vertices) which do not depend on frequency with an edge (and vertex) related only to *line* and *station* instead of *line*, *frequency* and *station*. This is shown in Figure 8.4. The capacity of the replacement edge (and resulting right hand side of constraints (8.7)), is given by $\sum_{f \in \mathcal{F}_l} C_f x_{l,f}$.



(a) Original graph structure **(b)** Reduced graph structure

Figure 8.4: The full and reduced graph structure for a single line with three frequencies at a single station.

Figure 8.4 shows the graph structure for a single station and a single line with three frequencies as originally described (Figure 8.4a) and with the explained reductions (Figure 8.4b). Nodes r and t are respectively the station source and sink vertices for passengers and p is the platform vertex for that station. The vertices f_1, f_2, f_3 are the *(line, frequency, station)* vertices for the three considered frequencies of the line, in that station. The red edges are the line switching edges (though no other lines are shown). Edges connecting these vertices f_1, f_2, and f_3 to corresponding vertices at other stations are not shown. Vertex $f_{1,2,3}$ is the combination of the vertices f_1, f_2, f_3. The edge between s and $f_{1,2,3}$, and between $f_{1,2,3}$ and t, is the combination of the edges between s and f_1, f_2 and f_3 in (Figure 8.4a), and f_1, f_2 and f_3 and t, respectively.

A second simplification of the model is that we consider line-switching edges only at a minimal set of switching stations. This set of stations is fixed beforehand and suffices to facilitate all optimal passenger flows, when every passenger origin-destination pair is considered individually. Any solution that is feasible for this restricted problem is feasible if switching is allowed at any station, but some solutions that are feasible if switching is permitted anywhere may not be feasible with the restriction (although we have not observed this). At stations where we do not permit switching we do not include switching edges, and this reduces the total number of edges in the graph by between 23% and 34% when tested for a range of line pools. Finally, we can determine that only a subset of all edges should be used for the flows from a given origin station; generally it is never true that in an optimal solution passengers will be assigned an edge that travels "towards" the station they originate from. This is a third measure to simplify the model.

By making these three alterations we find that the line planning problem is solvable directly as a MILP, though not to optimality in the time frame we require. For our tests, finding line plans with no other restriction, we use a time limit of one hour, or until a gap between the solution and best lower bound is below 0.5% (in most cases the gap limit is reached, but for some weightings of objectives, one hour is insufficient). However, for a reduced line pool that we use in the integrated approach described later, the problem becomes easier and is solvable to optimality in an acceptable time frame.

The formulation (8.2)–(8.9) defines the basic line planning model, ignoring some minor considerations not given here. However, when searching for line plans that only differ a little from a give line plan we may impose some additional restrictions. The simplest types are the following:

$$\sum_{l \in \mathcal{L}} \sum_{f \in \mathcal{F}_l} f \cdot x_{l,f} \geq k_1 \tag{8.10}$$

$$\sum_{l \in \mathcal{L}} \sum_{f \in \mathcal{F}_l} f \cdot x_{l,f} \leq k_2 \tag{8.11}$$

That is, we require that the total number of (one-directional) trains running in the network per hour is between some upper and lower bound. This may be, for example, to find solutions that do not differ too much from some original solution. We use this because, from the point of view of the timetable module, two solutions that differ only in line frequency but not in line routes can be very different. Without such constraints, when seeking a line plan that is similar but different to a given plan, a change of frequency would not maintain the similarities in timetabling that we seek. Now, suppose we are given a line plan or a partial line plan, in the form $\mathcal{X} = \{(l_1, f_1), (l_2, f_2), (l_3, f_3), ...\}$ where every (l, f) in $\mathcal{X}$ is a valid line and frequency combination, and that this (partial) line plan should not be in the solution. Then we may impose the following constraint for every such line plan:

$$\sum_{(l,f) \in \mathcal{X}} x_{l,f} \leq |\mathcal{X}| - 1. \tag{8.12}$$

Such constraints are used to forbid solutions we have already discovered and do not wish to find again, and also to forbid partial solutions which we already know are problematic for timetabling, i.e. they lead to timetable-infeasibility. Finally, and similarly, we may have some given line plan $\mathcal{X}$ and desire that the solution line plan contains at least k lines from the plan:

$$\sum_{(l,f) \in \mathcal{X}} x_{l,f} \geq k. \tag{8.13}$$

Such constraints ensure that a discovered line plan is similar to some previous line plan, while differing by some number of (unspecified) lines. If instead the lines that may differ are specified, we can fix the variables of the lines that may not differ and only permit those variables corresponding to the specified lines that may differ to change (along with variables corresponding to lines not in the plan). These extra restrictions are used in the integrated approach when looking for a similar line plan that is more flexible, i.e. allows a more robust timetable.

8.4.2 Timetabling module

The timetable problem is modelled as a PESP. We indicate our event-activity network as $(\mathscr{E}, \mathscr{A})$. The set of trains is indicated as T, the set of lines in the line plan (output of the line planning module) as $\mathcal{X}$, the line operated by train t is indicated as l_t, the set of stations is $\mathcal{S}$ and the set of stations on a line l is indicated as $\mathcal{S}_l$. As we assume a railway network with limited shunt capacity, our model assumes that all the trains can and must turn on their platform at end stations. The set T_{turn} contains the train couples (t, t') for which it holds that t becomes train t' after turning on the platform in its end station. Trains t and t' share the same rolling stock. Line l_t and $l_{t'}$ contain the same stations but in opposite direction. The set $T_{line\ spread}$ contains the train couples (t, t') where t and t' are two succeeding trains of the same line, i.e. no other train operating on the same line drives in between them.

The event set $\mathscr{E}$ of the event-activity network consists of the following events.

- The reservation of a station environment s (and a platform in this station) by a train t is a reservation event (t, s, res). We define $\mathscr{E}^{\text{res}}$ as $\{(t, s, \text{res}) \mid \forall t \in T, s \in S_{l_t}\}$.
- The release of a station environment s (and a platform in this station) by a train t is a release event (t, s, rel). We define $\mathscr{E}^{\text{rel}}$ as $\{(t, s, \text{rel}) \mid \forall t \in T, s \in S_{l_t}\}$.
- The reservation of a platform p_{s_e} in a terminal station s_e by a train t in order to turn is a platform reservation event $(t, p_{s_e,t}, \text{res})$. We define $\mathscr{E}^{\text{res},p}$ as $\{(t, p_{s_e,t}) \mid \forall t \in T\}$.
- The release of a platform p_{s_e} in a terminal station s_e by a train t in order to turn is a platform release event $(t, p_{s_e,t}, \text{rel})$. We define $\mathscr{E}^{\text{rel},p}$ as $\{(t, p_{s_e,t}) \mid \forall t \in T\}$.

The following inclusions hold $\mathscr{E}^{\text{res},p} \subset \mathscr{E}^{res} \subset \mathscr{E}$ and $\mathscr{E}^{\text{rel},p} \subset \mathscr{E}^{\text{rel}} \subset \mathscr{E}$ and $\mathscr{E} = \mathscr{E}^{\text{res}} \cup \mathscr{E}^{\text{rel}}$. So platform $p_{s_e,t}$ of train t in its terminal station can be interpreted as an extra station where the train arrives after arriving in its terminal station s_e.

The activity set $\mathscr{A}$ contains:

- *driving activities* between the release of a train in a station and the reservation of this train of the next station on its line. Let $\mathscr{A}^{drive} = \{((t, s, \text{rel}), (t, s', \text{res})) \in \mathscr{E}^{\text{rel}} \times \mathscr{E}^{\text{res}} \mid \forall t \in T$ and s and s' succeeding stations of $l_t\}$;

- *waiting activities* between the reservation and the release of a train in a station on its line. Let $\mathscr{A}^{wait} = \{((t,s,\text{res}),(t,s,\text{rel})) \in \mathscr{E}^{\text{res}} \times \mathscr{E}^{\text{rel}} \,|\, \forall t \in T, s \in S_{l_t}\}$;

- *buffer activities* between the release of one train and the reservation of another train on the same platform in the same station. Let $\mathscr{A}^{buffer} = \{((t,s,\text{rel}),(t',s,\text{res})) \in \mathscr{E}^{\text{rel}} \times \mathscr{E}^{\text{res}} \,|\, \forall t,t' \in T : t \neq t', s \in S_{l_t} \cap S_{l'_t}, p_{s,t} = p_{s,t'}\}$;

- *line spreading activities* between the reservations of two succeeding trains on the same line in the stations on their line. Let $\mathscr{A}^{line\ spread} = \{((t,s,\text{res}),(t',s,\text{res})) \in \mathscr{E}^{\text{res}} \times \mathscr{E}^{\text{res}} \,|\, \forall t,t' \in T : (t,t') \in T_{line\ spread}, s \in S_{l_t}\}$;

- *turning activities* between the release of a train of the platform in its end station and the release of the next train of the opposite line that leaves from that terminal station. Let $\mathscr{A}^{turn} = \{((t,p_{s_e,t},\text{rel}),(t',s_e,\text{rel})) \in \mathscr{E}^{\text{rel},p_{s_e,t}} \times \mathscr{E}^{\text{rel}} \,|\, \forall t,t' \in T : (t,t') \in T_{turn}\}$. This next train is the same physical train.

As mentioned in Section 8.2, we want to maximize the minimum buffer times between train pairs. In terms of the event-activity graph, we want to maximize the minimum activity time of the buffer activities which is the time between the release of an infrastructure part by one train and the reservation of that same infrastructure part by another train. Mathematically we have

$$\max \min_{a=(i,j)\in\mathscr{A}^{buffer}} (\pi_i - \pi_j + k_a P), \tag{8.14}$$

where π_i and π_j are the event times of event i and j respectively which define together a buffer activity. However, this objective function is not linear, but as it is a max-min objective function, it can easily be linearized. Therefore, we introduce an auxiliary variable $y \in [0,P]$, where P is the period of the cyclic timetable. We add the constraints

$$y \leq \pi_i - \pi_j + k_a P \qquad \forall a = (i,j) \in \mathscr{A}^{buffer} \tag{8.15}$$

and we change the objective function to the maximization of y: $\max y$. The complete model is then the following.

$$\max \quad y \tag{8.16}$$

$$y \leq \pi_i - \pi_j + k_a P \qquad \forall a = (i,j) \in \mathscr{A}^{buffer}$$

$$L_a \leq \pi_i - \pi_j + k_a P \leq U_a \qquad \forall a = (i,j) \in \mathscr{A} \tag{8.17}$$

$$0 \leq \pi_i \leq P \qquad \forall i \in \mathscr{E} \tag{8.18}$$

$$k_a \in \{0,1\} \qquad \forall a = (i,j) \in \mathscr{A} \tag{8.19}$$

Constraints (8.17) bound all activity times from below and above. The term $k_a P$ avoids negative activity times. To ensure a unique value for k_a, the value of U_a has to be smaller than the period P. The specific values of U_a and L_a are listed in Table 8.1 for all activities $a \in \mathscr{A}$. The driving activity times are bounded by the time that a train of line l needs between the release of a station s and the reservation of the next station s', indicated as $dr_{l,s,s'}$. The driving time between the terminal station of a train and the platform in its terminal station is zero minutes. The waiting activity times are bounded by the time that is necessary and provided for a line l to occupy a station s, indicated as $dw_{l,s}$. This is the time between the reservation and release time of that station. The bounds are different for waiting activities on platforms in terminal stations. Trains have to stay there for at least the necessary turn time in the terminal station s, which is indicated as ntt_s. Trains have to leave the platform before the next train arrives. This is ensured if they all get the same maximum time that they can stay on the platform which is equal to the period of the cyclic timetable divided by the number of trains that turn on platform p. The number of trains that turn on platform p is indicated as f_p. The buffer activities have to be positive and smaller than $P - dw_{l_{t'},s} - \epsilon$ to ensure that the timetable is feasible, i.e. occupation intervals may not overlap, independently of the order of both trains that will be assigned. On platforms in terminal stations the upper bound is smaller because trains occupy the platform for a longer time, i.e. the upper bound in our model is $P - \frac{P}{f_{p_{s_e,t}}} - \epsilon$. Before initializing the timetable module, a check is necessary to determine if too many trains are scheduled on one platform, i.e. if $\frac{P}{f_p} \leq ntt_s$. If so, the trains do not have enough time for turning and consequently the timetable will be infeasible. The value of ϵ depends on the time discretization. We use 0.1 minutes. In this model we equally distribute trains of a line over the period, and therefore the line spread activity times have to be equal to the period divided by the line frequency. The frequency of a line l is indicated as f_l. The turn activity times have to be equal to zero, ensuring that the 'turning' platform is freed if the next train leaves in the opposite direction.

Activity	L_a	U_a
$((t,s,\text{rel}),(t,s',\text{res})) \in A^{drive}$	$dr_{l_t,s,s'}$	$dr_{l_t,s,s'}$
$((t,s,\text{res}),(t,s,\text{rel})) \in A^{wait} : s \neq p_{s_e,t}$	$dw_{l_t,s}$	$dw_{l_t,s}$
$((t,s,\text{res}),(t,s,\text{rel})) \in A^{wait} : s = p_{s_e,t}$	ntt_s	$\frac{P}{f_{p_{s_e,t}}}$
$((t,s,\text{rel}),(t',s,\text{res})) \in A^{buffer}$	0	$P - dw_{l_{t'},s} - \epsilon$
$((t,s,\text{rel}),(t',s,\text{res})) \in A^{buffer} : s = p_{s_e,t} = p_{s_e,t'}$	0	$P - \frac{P}{f_{p_{s_e,t}}} - \epsilon$
$((t,s,\text{res}),(t',s,\text{res})) \in A^{line\ spread}$	$\frac{P}{f_l}$	$\frac{P}{f_l}$
$((t,p_{s_e,t},\text{rel}),(t',s_e,\text{rel})) \in A^{turn}$	0	0

Table 8.1: Lower and upper bounds for the PESP constraints (8.17)

8.4.3 Integrated approach

Here, we explain how the line planning and timetabling module can be integrated to construct a line plan and timetable that induce a low passenger and operator cost and maximize the buffer times between train pairs in order to provide a passenger robust railway schedule. The line planning and timetabling module work iteratively and interactively. The line planning module creates an initial line plan which is evaluated by the timetabling module. Based on the minimum buffer times between line pairs, a critical line in the line plan is identified. The line planning module then creates a new line plan with at least one different line, i.e. the time length of this critical line is changed. The goal is to create more flexibility in the line plan. This flexibility will be used by the timetabling module to improve its robustness. This heuristic approach which is divided into two parts is now further explained. In Figure 8.5, a visual overview of the algorithm is presented and in Section 8.6.2 we apply the approach to an example.

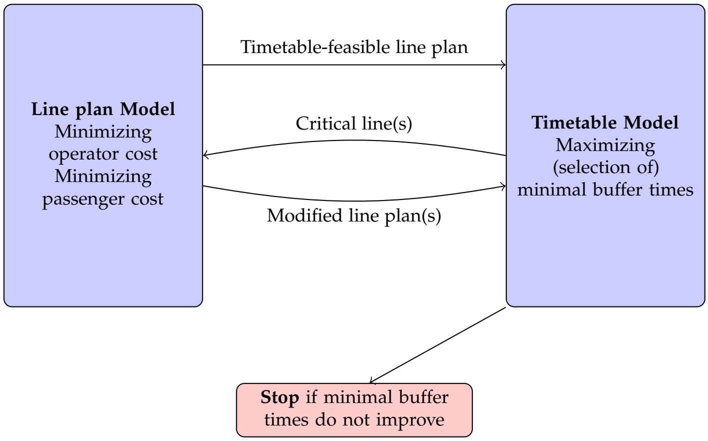


Figure 8.5: Overview of the integrated approach

Part 1: Initialization

Step 1: Construct an initial line plan

We construct a line plan that satisfies service constraints and optimizes a weighted sum of the passenger and operator cost with the line planning module. Beforehand, we check for infeasible lines in the line pool as discussed in Section 8.3. We check with the timetable module if a feasible timetable can be constructed for this line plan. A feasible timetable is a timetable in which no occupation intervals of trains overlap: if a station or platform is occupied by one train, no other train can occupy this station or platform until the first train leaves it. In case the constructed line plan is not timetable-feasible, different strategies can be applied. A straightforward strategy is to take the second best line plan for the weighted sum of the passenger and operator cost and if the second best is not timetable-feasible then the third best and so on. The disadvantage of this strategy is that it is possible that a lot of line plans are to be tested before a timetable-feasible line plan is found, because no insight in the problem is used. We propose another more effective strategy for a network with restricted shunt capacity as is assumed in this research. Due to the restricted shunt capacity in the terminal stations, the occupation of the terminal stations is critical in finding a timetable-feasible line plan. So an effective strategy for looking for a timetable-feasible line plan with a close to optimal objective value is by restricting the number of lines that share a terminal station. If a line using a shared terminal station also passes a different station that may be a terminal station, a close to optimal solution is a line plan in which this line is replaced by one that ends at this alternative terminal station. This decreases the number of lines sharing an end station and in some cases has minimal impact on operator and passenger costs. This new line plan is only feasible in case all service constraints remain fulfilled.

Part 2: Iterative steps

Step 2: Evaluate the line plan
Construct a timetable with the timetable module that maximizes the minimum buffer times between a selection, or between all the train pairs in the line plan. Calculate the minimum buffer times between all line pairs in the line plan, and the overall minimum buffer time. Refer to Section 8.6.2 for a concrete example. Test the following stop criteria:

- STOP if the minimum buffer time is closer than 5% to the desired minimum buffer time. The *desired minimum buffer time* can be found by identifying the station or track section which has the highest ratio of occupation time over free time and dividing the free time by the number of trains that pass by this section or station.
- STOP if the minimum buffer times do not improve the best encountered value during three succeeding iterations.

Otherwise, select the most critical line from the list. The most critical line is the line that is responsible for the highest number of buffers in the category of smallest buffers in the list. This is illustrated in the example in Section 8.6.2. In case of a tie, look at the next category of buffer times to identify the most critical line. If there is still a tie, let the decision be made by the line planning module in the next Step, based on the objective values there. The threshold to categorize the buffer times depends on the signaling system and the appearance of single tracks in the network. Go to Step 3.

Step 3: Adapt the line plan by changing the stopping pattern
Make a new line plan that alters the time length of the critical line by adding or removing a stop in a station on that line, such that this line becomes more flexible. This flexibility will be used to improve the buffer times in the timetabling module. This effect can be seen

in the results and the example presented in Section 8.6. There are three important considerations. Firstly, changing the time length can also make a line infeasible as discussed in Section 8.3 which has to be avoided. Secondly, an extra stop cannot be added to a line in cases where there are no skipped stations on the line. Thirdly, some stations cannot be skipped due to service constraints.

We potentially solve the line plan problem with three different line pools, sequentially, to attempt to find a feasible solution. If a feasible line plan is found, the line plan problem does not need to be solved for the other line pools in the sequence. The three line (and frequency) pools are as follows.

i. All lines of the solution of the previous iteration are fixed, including their frequency, except that of the critical line. We add all lines that differ by one stop from the critical line. For those lines we only allow the frequency of the critical line.
ii. All lines of the solution of the previous iteration are fixed, including their frequency, except that of the critical line. We add lines to the line pool that differ by one stop from the critical line, which we now allow at any frequency.
iii. Solution lines that share no stations with the critical line are fixed. We introduce lines that differ by one stop from the critical line and lines that differ from other non-fixed non-critical lines by one station, at any frequency.

Because the number of lines in the line pool and the number of feasible solutions is much more restricted, the run time for the line planning module is now much shorter. The objective function is the same as in Step 1. For the first line pool, if feasible, the best alternative line will be selected, i.e. the line that provides the lowest passenger and operator costs. For the second line pool, if feasible, one or more of these new lines will be selected, often with a frequency combination that sums to the frequency of the critical line. For the third line pool, one or more lines similar to the critical line will be selected, and other solution lines from the previous iteration may be replaced with one or more similar lines. A simple example of solution from the third line pool is where a stop at a certain station is shifted from the critical line to a line that first skipped this station. The time length of the critical line changed by removing a stop and the station that is now skipped by the critical line is still served, but by another line. Note that in this example, the length of the non-critical line is also changed. A composition resulting from the second line pool is captured in the example in Section 8.6.2.

In the case that a feasible solution is found, return to Step 2. In the case that no feasible solution is found, and if there is a second most critical line, solve the three line plan problems for the second most critical. Otherwise STOP.

End

The intuition behind the integrated approach is the following. Changing the number of stops of a line changes the time length of the line. This time length of a line affects the flexibility of that line. So we alter the stop pattern of a line to make the line more flexible in order to improve the spreading in the whole network. The station where the stop pattern is changed is decided by the line planning module, which takes a weighted sum of passenger and operator cost into account. These costs are not taken into account during timetable construction. We note here that in general we do not require that the lines created to modify a line plan are all in the original pool of lines specified for the original problem. This explains why the adapted line plan can have a better weighted sum of passenger and operator cost than the original one. For the second stop criterion we take three non-improving iterations, to both restrict the run time while still allowing improvements that require multiple lines to change before a resultant improvement in minimal buffer time is observed.

8.5 Case study

The railway system on which the approach is tested is the S-tog network in Copenhagen operated by Danish railways operator DSB. This is a cyclic high-frequency network with a one hour period of repetition and which transports 30 000 to 40 000 passengers per hour at peak times between 84 stations. The network is visualized in Figure 8.6. It contains a central corridor, indicated in red; five 'fingers', indicated in blue; and a circle track, indicated in yellow. During peak hour on weekdays, there is a service requirement of 30 trains per hour through the central corridor in each direction. Each train occupies a station area on its route, in the direction that the train is driving, for one minute. The minimum *desired buffer time* in the DSB S-tog network is therefore one minute, which is (60 min - 30 min)/30, where 60 minutes is the period of the cyclic timetable and 30 trains occupy a station in the central corridor each for one minute. With almost no exception, there are at least two tracks in between every two adjacent stations. Furthermore, given that the network is designed with bridges and tunnels, there are very few locations where trains in opposite directions have to cross each other during normal conditions. In this research we assume that trains in opposite directions only interact with each other in terminal stations. One requirement specified by the operator is that only lines at frequency three, six, nine or twelve are allowed in the weekday line plan. This restriction decreases the probability of frequency combination infeasibility (though that is not necessarily the intention for the requirement). In order to enable and maintain this high frequency in the central corridor, the spreading of the trains in this part of the network is crucial. Therefore the timetable module will be sequentially used twice with two different objective functions. First, the minimum buffer times in the central corridor are optimized. In a second optimization round the minimum buffer in the rest of the network is optimized while bounding the buffer times in the central corridor by the value found in the first optimization. We also considered one combined weighted objective function, but this proved computationally worse in our experiments, i.e. the run times were significantly higher.

We test our approach on ten line plans for this network. The approach can be applied to a pre-existing line plan, or applied to first create and then improve a line plan. The full approach is tested for five line plans created as described in Step 1 of the integrated approach, while the other five line plans come from the operator or are created by hand. The first two line plans (1-2) were recently in use for the S-tog network in Copenhagen. We have not considered the current line plan as it is only temporarily active and specifically developed for implementing the new signaling system in the central corridor of the network. The third line plan (3) is a night line plan for weekdays. As the demand during night time is lower, the frequencies of the lines in this line plan are also lower. All other line plans are line plans that are planned with the requirements for use during daytime on weekdays. So, the setting of this third line plan is different from the other ones. This third line plan is also not the current plan in the S-tog network as at the present time a temporary plan is in use. The fourth up to the eighth line plan (4-8) are created within our algorithm by solving the weighted sum line planning module, using a range of weights that give distinct line plans. For each of these weights, we solve the line planning model

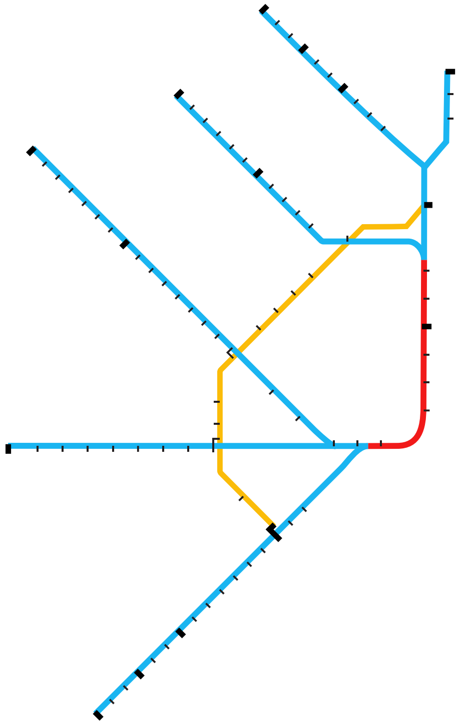


Figure 8.6: DSB S-tog network of Copenhagen

with a one hour time limit and to a 0.5% relative gap limit, and terminate when either is reached. We initially solve the line planning module finding distinct solutions with no consideration for the feasibility of timetables except for infeasible lines as explained in Section 8.3. Then we test whether or not these are timetable-feasible. We find for these considered weights that only a single line plan (4) is feasible for timetabling. This endorses the statement that the output of a previous level in railway planning is not necessarily adequate for the next planning level [Schöbel, 2015]. For those that are not feasible, we introduce restrictions on the use of terminal platforms, requiring only one line terminating at each terminal platform a station has. This is described in Step 3 of the integrated approach in Section 8.4. This is sometimes too conservative, since it can be possible for more than one line to share a single terminal platform. Conversely this alone does not guarantee that a feasible timetable is present for a line plan, but we observe that it is often a sufficient restriction. Applying this restriction we find four other distinct line plans (5-8). We note that, when considering the two line plan objectives of operator cost and passenger cost, none of the final four plans dominates any other. The ninth and the tenth line plan (9-10) are two special 'manually created' line plans, which are each based on one of the weighted-objective line plans (5 and 8 respectively). These paired plans only differ in stopping pattern from the plan they are manually adapted from, as we force every line to stop in every station it passes, while the original line plans contain many skipped stations. We want to investigate if each pair (5 and 9, 8 and 10) converges to a final line plan of similar quality when we modify stopping patterns of lines.

8.6 Results and discussion

In this section we show the results of the integrated approach for all ten line plans described in Section 8.5. Furthermore, we demonstrate the integrated approach for line plan 2.

8.6.1 Results for ten line plans

A first performance indicator is the estimated operator cost of a line plan. This cost is calculated by the line planning module. The total cost of a line plan is simply the sum of estimated operator costs for each line, which we take as given by the rail operator, here DSB. Each line in the pool has an operating cost associated with each frequency at which it could operate, and in calculating the total cost there are no additional considerations given to the combinations of lines.

A second performance indicator is the estimated passenger cost of a line plan. This cost is calculated by the line planning module. It is the sum of travel time of all passengers in the OD matrix. Because a timetable is not known (by the line planning module), the transfer time is estimated based on the frequency of the line as half of the time between two trains of that commuter line. For each passenger transfer an additional penalty of six minutes is added to the estimated passenger cost as transfers are perceived to be worse than direct connections.

The third performance indicator is the minimum buffer time between train pairs in the central corridor of the DSB S-tog network, optimized by the timetable module. The fourth performance indicator is the minimum buffer between train pairs everywhere in the network, while bounding the minimum buffer time in the central corridor first.

This fourth performance indicator is also optimized by the timetable module. The focus on the minimum buffer time first in the central corridor of the network and thereafter on the minimum buffer time overall is in consultation with DSB S-tog.

A fifth performance indicator is the sum of the inverse of the minimum buffer times between train pairs in each station that they have in common (and pass by in the same direction). We take the inverse minimum buffer times in order to give smaller buffers a higher weight than large buffers. As in Dewilde et al. [2013] a buffer time smaller than the time discretization ϵ (here 0.1 minute) has a contribution of 15 to the sum of the inverse buffer times. So the lower the sum of the inverse buffer times the better, because this means generally larger buffer times. The results are summarized in Table 8.2, Table 8.3 and Table 8.4.

Line plan		Min buffer in central corridor (min)			Min buffer overall (min)			Sum of inverse buffer times (1/min)		
		initial	final	impro	initial	final	impro	initial	final	impro
1	real	0.63	1.00	+58%	0.00	1.00	+∞%	2 639	2 189	-17%
2	real	0.73	1.00	+36%	0.00	1.00	+∞%	2 348	2 212	-6%
3	real	3.00	3.00	+0%	0.70	2.55	+264%	482	382	-21%
4	random	0.33	0.64	+93%	0.00	0.05	+∞%	3 293	3 323	+1%
5	random	0.17	0.83	+400%	0.00	0.20	+∞%	3 840	2 365	-38%
6	random	0.37	0.99	+170%	0.00	0.01	+∞%	3 211	2 929	-9%
7	random	0.23	0.23	+0%	0.00	0.00	+0%	4 324	4 324	-0%
8	random	0.23	0.23	+0%	0.00	0.00	+0%	4 357	4 348	-0%
9	special	1.00	1.00	+0%	0.70	1.00	+43%	2 318	2 203	-5%
10	special	0.92	1.00	+8%	0.00	0.00	+0%	3 179	3 362	+6%

Table 8.2: The integrated approach significantly improves the buffer times in eight out of ten of the studied line plans.

In Table 8.2 we observe that there is a significant improvement in the buffer times for eight out of the ten line plans. For three out of the ten line plans, the desired minimum buffer time is reached both in the central corridor and in the rest of the network. For three other line plans the desired minimum buffer time is reached in the central corridor but not in the rest of the network. Furthermore, we see that the sum of the inverse buffer times between train pairs in every station they have in common decreases, which means that the buffer times themselves increase as desired. Moreover, the results on the sum of the inverse buffer times are very similar to the minimum buffer time results in the central corridor and in the overall network. We note that a big absolute improvement of the minimum buffer time in the central corridor (or of the minimum buffer time overall) corresponds to a big improvement in the sum of the inverse buffer times, and vice versa. Unfortunately, for two out of the ten line plans (7 and 8) no improvement in minimum buffer time is achieved. To identify the critical line in Step 2 of the integrated algorithm, we categorize the buffers as zero, smaller than 30 seconds, smaller than one minute and bigger than one minute. We observe that for the two timetables corresponding to the initial line plans almost half of the minimum buffer times between line pairs are smaller than 30 seconds, while for the other line plans this is at most one third of the minimum buffer times. As a possible explanation, we note that for these line plans almost every line has a pairwise minimum buffer time below half a minute with some other line, and we may therefore expect that multiple lines must be modified to see an improvement. We typically change a single line in every iteration and in such cases, it may take more than three non-improving iterations before seeing an improvement, given that every line plan we consider has between six and ten lines.

The buffer times in Table 8.2 appear to be small. However, as discussed earlier, the minimum desired buffer time in a daytime week line planning is one minute and this is thus the value of the stopping criterion in Step 2 of the integrated approach. The maximal minimum buffer time everywhere in the network is also restricted by this maximal minimum buffer time in the central corridor. Moreover, even if the buffer time between two trains is zero the timetable is still feasible. A zero buffer time only means that the second train reserves and occupies a track section between two signals immediately after the first train leaves and releases this track section. However, a zero buffer time is undesirable and any delay of the first train is immediately propagated to the second. In the case study, a line plan performs best if it allows the desired buffer time of at least one minute between every two trains. As an exception, for line plan 3 the desired buffer time is three minutes. This desired buffer time is for example achieved for line plan 9.

One explanation for not reaching the desired value for some line plans could be no further improvement was made, because at each iteration the same line was identified as being critical. Either changing this line was no longer feasible or changing this line was feasible, but did not result in acceptable solutions. If changing the critical line is not feasible, then the second most critical line of the last found line plan is chosen. In the current algorithm, however, if the critical line itself does not give rise to good results, there is no backtracking to a previous iteration to try the second most critical line.

Line plan		Operator cost ($\times 10^5$)			Passenger cost ($\times 10^7$)		
		initial	final	change	initial	final	change
1	real	6.79	6.84	+0.74%	4.17	4.23	+1.47%
2	real	6.84	7.21	+5.40%	4.22	4.21	-0.12%
3	real	3.40	3.43	+0.64%	1.05	1.06	+1.08%
4	random	6.25	6.64	+6.23%	4.24	4.27	+0.87%
5	random	6.48	6.80	+4.94%	4.27	4.29	+0.36%
6	random	6.66	6.74	+1.13%	4.12	4.14	+0.51%
7	random	7.02	7.02	+0.00%	4.09	4.09	+0.00%
8	random	8.27	8.32	+0.71%	4.05	4.04	-0.22%
9	special	7.15	7.14	-0.17%	4.43	4.44	+0.32%
10	special	9.00	9.01	+0.20%	4.35	4.30	-1.06%

Table 8.3: The operator cost and passenger cost differ only slightly when applying the integrated approach for most plans.

In Table 8.3 the operator cost and the passenger cost for the initial and final line plan are presented. We observe that they differ only slightly for most plans. This is important because it illustrates that our improvement in terms of expected delay propagation, by modifying the line planning and timetable, does not necessarily require an increase in operator and passenger cost. Note in fact that in Table 8.3 some plans do improve for one measure but become worse for another, and although it is possible for both to improve (as we are in fact allowing lines that were not in the original line pool), we do not observe this here. Note also that for line plans 4 and 5 we do see a relatively large increase in operator cost (6.23% and 4.94%), combined with an increase in passenger cost which may be a relatively large cost to pay for timetable improvement. In contrast, for line plan 2 though we see a similarly large increase in operator cost but a reduction in passenger cost. Here the impact must be judged by the perceived relative importance of the two measures.

Line plan		Stop criterion	# iterations	# out-of-pool lines	Average run time timetabling (min)
1	real	DES	4	5	183.40
2	real	DES	3	5	4.88
3	real	BFV	2	3	0.50
4	random	BFV	6	5	75.71
5	random	BFV	7	7	385.563
6	random	BFV	7	5	167.19
7	random	BFV	3	0	126.75
8	random	BFV	3	1	9.25
9	special	DES	1	2	47.5
10	special	BFV	5	3	346.83

Table 8.4: Characteristics of the integrated approach

In Table 8.4 some characteristics of the integrated approach are presented together. We indicate there under which stop criterion the algorithm was terminated. If the algorithm terminated because the minimum buffer time between the selected train pairs is closer than 5% to the desired minimum buffer time, we indicate this as 'DES' from desired. If the algorithm ended because the minimum buffer time between the selected train pairs did not improve the best found value in three consecutive iterations, we indicate this as 'BFV', i.e. best found value. We see that three out of the ten line plans were within the desired minimum buffer time and for the remaining seven the algorithm ended with the best found value. The table also reports how many iterations the integrated approach passed through before a stop criterion was achieved. This value ranges between one and seven. We report the number of out-of-pool lines which are in the final solution, referring to lines that are in the final solution but do not come from the original, restricted, line pool but instead are similar to a line in the pool but with a modified stop pattern. We observe that the five line plans with the highest number of out-of-pool lines (line plan 1, 2, 4, 5 and 6) have the greatest relative improvement of the minimum buffer time in the central corridor; have the greatest increase in operator cost; and (with one exception) have the highest increase in passenger cost. Therefore, including new lines in the line pool has the potential to improve the minimum buffer times significantly, but may have a negative effect on passenger and operator costs.

The final characteristic in Table 8.4 is the run time for timetabling. The total run time for timetabling consists of the creation of the optimal timetable in Step 2 of the integrated approach for each iteration and for determining the initial timetable. As described in Section 8.5 the timetable is solved sequentially with two objective functions at each iteration. Firstly, the buffer time in the central corridor is maximized, and secondly the buffer time in the rest of the network is increased with a bound on the buffer time in the central corridor fixed by the first step.

For example the algorithm stops after three iterations for line plan 2. This means that eight timetables are calculated: two initially (for the two optimization criteria) and two at each of the three iteration steps. The average run time for timetabling for optimizing line plan 2 is 4.88 minutes, which means that the run time for each calculated timetable in the integrated approach on average is 4.88 minutes. All timetables are calculated with CPLEX 12.6 on an Intel Core i7-5600U CPU @ 2.60 GHz. We observe that there is a high variability in the average run times for the different line plans. Moreover, a high computation time may occur in case where there is both a big improvement (line plans 1 and 5) and also where there is either no improvement (line plan 7) or only a small improvement (line plan 10). Furthermore, the run time for timetabling can differ significantly from one iteration to the next. Even if two line plans are not dissimilar one can be intrinsically more difficult to solve. An explanation could be that due to changes in the stopping pattern, trains of different lines are more or less susceptible to catching up with each other in the fingers of the S-tog network, resulting in it being more complex to spread the trains optimally. The timetable module runs to optimality (relative gap smaller than 0.05%) for about 85% of the timetables. The average run time per timetable optimized within the time limit of 12 hours is 3801 seconds. For the other optimizations a time limit of 12 hours is imposed. The line planning module for the selected line pool determined by the critical line runs to optimality in all instances, taking at most up to ten minutes for cases where many lines are to be changed.

Finally, from Tables 8.2, 8.3 and 8.4, we deduce that line plans 5 and 9 did not converge to the same final line plan and line plans 8 and 10 did not either. We see that the final line planning and timetable for line plan 9 and 10 score better on robustness, i.e. minimum buffer time between line pairs is larger, while line plan 5 and 8 score better on operator and passenger cost. Based on these results, the final decision on which line plan is preferred, rests with the operator. In our opinion, the optimized version of line plan 1 will be the most passenger robust, without adding expense for the operator or passengers.

8.6.2 Illustration

In order to illustrate the integrated approach, we apply it to line plan 2. As this is an existing line plan, we skip Part 1 of the algorithm and only look at the iterative steps in Part 2. The estimated operator cost of this line plan is 6.84×10^5, the estimated total passenger travel time is 4.22×10^7. The optimal value for the minimum buffer time for this line plan in the central corridor of the network is 0.73 minutes. The optimal value for the minimum buffer time overall if the minimum buffer time in the central corridor is bounded below by 0.73 minutes is zero minutes. The minimum buffer times between the line pairs are present in minutes in Table 8.5. The smallest buffer time between line i and j is the same as the smallest buffer time between line j and i, so Table 8.5 is in fact symmetric, but we omitted here the superfluous information. If two lines do not share a part of the network, the minimum buffer time between these lines is indicated as 60 minutes, which is the period of the cyclic timetable. The smallest buffer time between two lines is zero minutes. This buffer time is between line 1 and itself. This means that the turn platform of line 1 in one of its terminal stations is permanently occupied by a train of the first line. Obviously, the critical line is line 1.

line	0	1	2	3	4	5	6	7
0	1.73	0.73	0.73	2.47	2.88	3.13	2.47	60
1	-	**0.00**	2.47	0.73	1.15	2.57	0.73	60
2	-	-	2.2	1.30	1.47	1.30	3.50	60
3	-	-	-	7.17	0.88	1.40	6.70	60
4	-	-	-	-	4.45	7.58	1.03	60
5	-	-	-	-	-	2.87	0.80	60
6	-	-	-	-	-	-	6.97	60
7	-	-	-	-	-	-	-	0.57

Table 8.5: The minimum buffer time overall is zero minutes, if the minimum buffer time in the central corridor is bounded below by 0.73 minutes

The line planning module adds a stop to line 1 by considering only the line pool that contains alternatives for line 1 of the same frequency. The new estimated operator cost increases to 6.99×10^5 and the new estimated total passenger travel time slightly increases to 4.23×10^7. The optimal value for the minimum buffer time for this line plan in the central corridor of the network is 1.00 minute. The optimal value for the minimum buffer time overall if the minimum buffer time in the central corridor is bounded below by 1.00 minute is still zero minutes. The minimum buffer times between the line pairs of the first modification of line plan 2 are present in minutes in Table 8.6. The smallest buffer time between two lines is still zero minutes. This buffer time is now only associated with line 6. Again, this means that the turn platform of line 6 in one of its terminal stations is permanently occupied by a train of line 6. The critical line is line 6.

line	0	1	2	3	4	5	6	7
0	6.17	1.01	2.99	60	3.01	1.00	3.99	2.99
1	-	2.14	2.99	60	1.00	1.99	1.00	1.00
2	-	-	0.29	60	1.00	1.00	1.01	1.00
3	-	-	-	0.67	60	60	60	60
4	-	-	-	-	1.82	2.99	2.99	2.99
5	-	-	-	-	-	7.23	6.99	6.99
6	-	-	-	-	-	-	**0.00**	1.00
7	-	-	-	-	-	-	-	3.02

Table 8.6: The minimum buffer time overall is zero minutes, if the minimum buffer time in the central corridor is bounded below by 1.00 minute

In Step 2, the line planning module first considers line pools that contain only alternatives for line 6 of the same frequency and of different frequencies, but they do not lead to a feasible line plan. We then consider the line pool that also contains alternative lines for different frequencies for the lines that share a part of the network with line 6. The result is a feasible line plan that does not include original line 6 and 7, each of frequency three, but contains a new line of frequency 6. The original line 6 stops at the same stations as the original line 7, but has some additional stops at one end of the line. The new line is similar to the original line 6 but skips one stop of that line. The new estimated operator cost is 7.22×10^5 and the new estimated total passenger travel time is 4.20×10^7. The optimal value for the minimum buffer time of this line plan in the central corridor of the network remains 1.00 minute. However, the optimal value for the minimum buffer time overall if the minimum buffer time in the central corridor is bounded below by 1.00 minute has now increased to 0.70 minutes. The minimum buffer times between the line pairs of the second modification of line plan 2 are present in minutes in Table 8.7. The smallest buffer time between two lines is now 0.7 minutes. This buffer time is now only associated with line 3, so the new critical line is line 3.

line	0	1	2	3	4	5	6
0	10.16	2.99	1.00	60	2.99	8.98	1.00
1	-	2.10	2.99	60	1.00	2.01	1.00
2	-	-	1.15	60	1.00	1.01	3.01
3	-	-	-	**0.70**	60	60	60
4	-	-	-	-	2.06	3.01	2.99
5	-	-	-	-	-	8.05	1.00
6	-	-	-	-	-	-	1.69

Table 8.7: The minimum buffer time overall is 0.70 minutes, if the minimum buffer time in the central corridor is bounded below by 1.00 minute

The line planning module skips a stop of line 3. The new estimated operator cost is 7.21×10^5 and the new estimated total passenger travel time is 4.21×10^7. The optimal value for the minimum buffer time of this line plan in the central corridor of the network is still 1.00 minute. The optimal value for the minimum buffer time overall if the minimum buffer time in the central corridor is bounded below by 1.00 minute is now 1.00 minute. The minimum buffer times between the line pairs of the second modification of line plan 2 are present in minutes in Table 8.8. This minimum buffer time overall is closer than five percent to the minimum desired buffer time of one minute, so this is the last iteration of the algorithm.

line	0	1	2	3	4	5	6
0	10.20	3.01	1.00	60	2.99	8.98	1.00
1	-	2.15	2.99	60	1.00	1.99	1.00
2	-	-	1.09	60	1.00	1.00	2.99
3	-	-	-	1.00	60	60	60
4	-	-	-	-	2.11	2.99	2.99
5	-	-	-	-	-	8.02	1.00
6	-	-	-	-	-	-	1.66

Table 8.8: The minimum buffer time overall is 1.00 minute, if the minimum buffer time in the central corridor is bounded below by 1.00 minute

8.7 Conclusion and further research

This paper presents a heuristic algorithm that builds a line plan from scratch resulting in a feasible and robust timetable. Our method iterates interactively, alternating between a line planning module and a timetabling module, improving the robustness of an initially built line plan. Both modules consist of an exact optimization model. The line planning module optimizes a weighted sum of passenger and operator costs, while the timetabling module focuses on improving minimum buffer times between line pairs. Appropriate and sufficiently large buffer times between train pairs are needed to reduce the risk of delays being propagated from one train to the next, thereby obtaining a robust railway schedule. The timetable module identifies a critical line based on the minimum buffer times between line pairs. The line planning module creates a new line plan in which the time length of the critical line is changed. Changing the time length of a line may create more flexibility in the schedule, which may result in improvements in robustness. The approach was tested for ten different line plans on the DSB S-tog network in Copenhagen. This is a high-frequency railway network with 84 stations, currently nine lines and restricted shunt capacity in the terminal stations. For eight out of ten initial line plans the robustness could be significantly improved, while the changes to the line plan generally did not result in significant changes in the weighted sum of operator and passenger cost. Ultimately the operator makes the final decision on the preferred criterion, considering the measures we have presented and others we have not captured.

An initial idea for future research is a smart extension of the integrated approach to overcome the situation where a certain line remains critical in each iteration, while keeping the computation time restricted. Another extension would be to allow different shunt characteristics in different terminal stations. In the presented research we had the very strict requirement present in the DSB S-tog system to have a schedule in which no train uses shunt capacity in a terminal station during daily operation. Furthermore, the development of a single integrated exact model that combines line planning and robust timetabling which is solvable in a reasonable amount of time for other real networks (similar to the DSB S-tog network) would be a next noteworthy step. A further idea for future research is to remove the requirement that trains of a line must operate exactly evenly timed (e.g. once every ten minutes for a six-per-hour line). Currently, this requirement is consistent with operation and ensures a regular service for customers. However it is potentially severely restrictive for the timetable given the tight spacing of trains in the central corridor. Relaxing this requirement could increase the complexity of the timetable model, both by expanding the solution space and by requiring new constraints and possibly an objective measure for the evenness of train timings.

8.8 Acknowledgements

This research is partly funded by a travel grant for a long stay abroad financed by Research Fund - Flanders (FWO). We also thank the Danish railway operator DSB for their help and commitment during the entire project.

8.9 Bibliography

Ralf Borndörfer, Martin Grötschel, and Marc E. Pfetsch. A column-generation approach to line planning in public transport. *Transportation Science*, 41(1):123–132. 2007.

Michael R. Bussieck, Peter Kreuzer, and Uwe T. Zimmermann. Optimal lines for railway systems. *European Journal of Operational Research*, 96(1):54–63. 1997.

Gabrio Caimi, Martin Fuchsberger, Marco Laumanns, and Marco Luethi. A model predictive control approach for discrete-time rescheduling in complex central railway station areas. *Computers & Operations Research*, 39:2578–2593. 2012.

Thijs Dewilde, Peter Sels, Dirk Cattrysse, and Pieter Vansteenwegen. Defining robustness of a railway timetable. In *Proceedings of the 4th International Seminar on Railway Operations Modelling and Analysis (RailRome), Rome, Italië*. 2011.

Thijs Dewilde, Peter Sels, Dirk Cattrysse, and Pieter Vansteenwegen. Robust railway station planning: An interaction between routing, timetabling and platforming. *Journal of Rail Transport Planning & Management*, 3(3):68–77. 2013.

Marc Goerigk, Michael Schachtebeck, and Anita Schöbel. Evaluating line concepts using travel times and robustness. *Public Transport*, 5(3):267–284. 2013.

Jan-Willem Goossens, Stan van Hoesel, and Leo Kroon. On solving multi-type railway line planning problems. *European Journal of Operational Research*, 168(2):403–424. 2006.

Peter Großmann. Polynomial reduction from PESP to SAT. Technical report, TU Dresden. 2011.

Mor Kaspi and Tal Raviv. Service-oriented line planning and timetabling for passenger trains. *Transportation Science*, 47(3):295–311. 2013.

Leo Kroon, Dennis Huisman, and Gábor Maróti. Optimisation models for railway timetabling. In Ingo A. Hansen and Jörn Pachl, editors, *Railway timetable & traffic: Analysis, modelling and simulation*, pages 135–154. Eurailpress, Hamburg. 2008.

Christian Liebchen. *Periodic Timetable Optimization in Public Transport*. Ph.D. thesis, Technische Universität Berlin. 2006.

Christian Liebchen and Rolf H. Möhring. The modeling power of the periodic event scheduling problem: railway timetables - and beyond. In *Algorithmic methods for railway optimization*, pages 3–40. Springer. 2007.

Richard M. Lusby, Jesper Larsen, Matthias Ehrgott, and David Ryan. Railway track allocation: Models and methods. *OR Spectrum*, 33(4):543–883. 2011.

Mathias Michaelis and Anita Schöbel. Integrating line planning, timetabling, and vehicle scheduling: a customer-oriented heuristic. *Public Transport*, 1(3):211–232. 2009.

Karl Nachtigall. Periodic network optimization with different arc frequencies. *Discrete Applied Mathematics*, 69:1–17. 1996.

Jens Parbo, Otto A. Nielsen, and Carlo G. Prato. Passenger perspectives in railway timetabling: A literature review. *Transport Reviews*, pages 1–27. 2015.

Leon W.P. Peeters. *Cyclic Railway Timetable Optimization*. Ph.D. thesis, Erasmus Research Institute of Management (ERIM). 2003.

Natalia J. Rezanova. Line planning optimization at DSB. In *13th Conference on Advanced Systems in Public Transport*. 2015.

Miguel A. Salido, Federico Barber, and Laura Ingolotti. Robustness for a single railway line: Analytical and simulation methods. *Expert Systems with Applications*, 39(18):13305 – 13327. 2012.

Anita Schöbel. Line planning in public transportation: models and methods. *OR spectrum*, 34(3):491–510. 2012.

Anita Schöbel. Integration of line planning and timetabling. In *Proceedings of the Conference on Advanced Systems in Public Transport*. 2015.

Anita Schöbel and Susanne Scholl. Line planning with minimal traveling time. In *ATMOS 2005-5th Workshop on Algorithmic Methods and Models for Optimization of Railways*. Internationales Begegnungs-und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl. 2006.

Alexander Schrijver and Adri Steenbeek. Spoorwegdienstregelingontwikkeling (Timetable construction). Technical report, CWI Center for Mathematics. 1993.

Peter Sels, Thijs Dewilde, Dirk Cattrysse, and Pieter Vansteenwegen. Reducing the passenger travel time in practice by the automated construction of a robust railway timetable. *Transportation Research Part B Methodological*, 84:124–156. 2016.

Paolo Serafini and Walter Ukovich. A mathematical model for periodic scheduling problems. *SIAM Journal on Discrete Mathematics*, 2(4):550–581. 1989.

Pieter Vansteenwegen, Thijs Dewilde, Sofie Burggraeve, and Cattrysse Dirk. An iterative approach for reducing the impact of infrastructure maintenance on the performance of railway systems. *European Journal of Operational Research*, 252(1):39–53. 2016.

Chapter 9

Exact Methods for Solving the Train Departure Matching Problem

With Jørgen Haahr.[1]

Submitted to Annals of Operations Research

[1]Technical University of Denmark. jhaa@dtu.dk

Abstract. *In this paper we consider the train departure matching problem which is an important subproblem of the Rolling Stock Unit Management on Railway Sites problem introduced in the ROADEF/EURO Challenge 2014. The subproblem entails matching arriving train units to scheduled departing trains at a railway site while respecting multiple physical and operational constraints. We formally define that subproblem, prove its NP-hardness, and present two exact method approaches for solving the problem. First, we present a compact Mixed Integer Program formulation which we solve using a MIP solver. Second, we present a formulation with an exponential number of variables which we solve using column generation. Our results show that both approaches have difficulties solving the ROADEF problem instances to optimality. Due to the complexity of solving the instances for this problem we have developed a heuristic based on both approaches. The column generation approach is able to generate good quality solutions within a few minutes in a heuristic setting.*

9.1 Introduction

Many railway planning problems have been studied in the literature for the last two decades. These range from long term high and level planning problems, such as line planing, to detailed and short term rolling stock and crew scheduling problems. At train stations, planning problems include platform assignment, routing, and shunting. These problems often have direct or indirect dependencies but due to the high complexity, or company organizational structure, they are often solved in isolation and in sequential order. The *ROADEF/EURO Challenge 2014* presents a problem where the goal is to adopt a holistic approach to the planning problems at railway stations. The problem combines several planning aspects that must be handled between arrivals and departures at terminal stations such as matching available trains to departures, routing trains in the station infrastructure (without any two trains occupying the same infrastructure without sufficient time between them), determining whether and when to perform maintenance, and when and where to do the couplings and decouplings of train convoys.

In this paper we present our contribution for the Departure Matching Problem (DMP) in this challenge. The DMP can be considered a pure subproblem of the problem presented in the *ROADEF/EURO Challenge 2014* document Ramond and Nicolas [2014], which we will refer to as the Rolling Stock Unit Management on Railway Sites (RSUM) problem.

The RSUM problem entails many different aspects but the performance of any solution approach will be greatly affected by how trains are matched to departures. The considered subproblem (DMP) is the problem of finding a good and feasible matching of trains to departures while respecting train compatibility and maintenance constraints. In contrast to the RSUM problem the DMP does not consider how to route, couple and de-couple train units in the station. For the purpose of this paper, we will assume that routing is done in a subsequent step. Importantly, many routing decisions are motivated by the matching of trains to departures; whether an arriving train should visit a maintenance facility or be parked in a yard depends on the train's subsequent departure.

In this paper we propose and benchmark two distinct optimal solution methods for solving the DMP. We will consider finding solutions that are either optimal or proven to be some percentage from the optimum. The proposed solution methods are however flexible and can be adjusted to find solutions in a heuristic manner. We investigate the potential of the methods in a heuristic setting.

9.1.1 Our Contributions

We present a definition for the DMP; a distinct, self contained subproblem of the RSUM. We prove the DMP to be NP-hard in the strong sense. Two optimization-based approaches for solving instances of the DMP are proposed. Firstly, we introduce a Mixed Integer Program (MIP) mathematical model and present results produced using a commercial MIP solver. Secondly, we propose an alternative but equivalent MIP mathematical model that is solved using column generation. Due to time limitations and the hardness of the instances we only solve the pricing problem at the root node. The results are presented in the benchmark section.

9.2 Problem definition

A full solution to the RSUM problem requires routing trains in the station infrastructure, respecting routing restrictions, headways, capacities, and many other constraints. Solutions are ranked using a weighted sum of multiple objectives such as making preferred matchings, allocating arrivals and departures to preferred platforms, avoiding unnecessary and platform dwell times. Solving the entire problem as a single optimization problem is intractable considering the given strict run-time requirements of the competition. The RSUM problem can be decomposed into several subproblems. The first subproblem, namely the DMP, is the scope of this paper.

A solution to the matching subproblem (DMP) can be used to build a solution to the overall RSUM problem, taking into account its other constraints and objectives. However the fixed matching provided from the DMP may lead to suboptimal solutions to the entire problem. Our approach for solving instances of the RSUM is to first solve a DMP instance, and then use an optimal approach to assigning platforms, before heuristically finding train routes. Our solution method is described in Haahr and Bull [2014]. The DMP contains components of the RSUM that are weakly linked to following subproblems in order to improve the tractability of the RSUM.

The DMP is a matching problem between arrivals and departures at some terminal station, with complicating dependencies between potential matches. Given a fixed planning horizon at the station, there is a set $\mathcal{A}$ of trains arriving and a set of pre-specified departures $\mathcal{D}$ that must be assigned some compatible train unit. Every arrival $a \in \mathcal{A}$ has an arrival time $arrTime_a$, and every departure $d \in \mathcal{D}$ has a departure time $depTime_d$. One arrival corresponds to a single train unit, and if several trains arrive together as a convoy then they are represented by multiple arrivals with the same arrival time. Similarly one departure exists for every train that departs in a convoy sharing the same departure time. A set $\mathcal{I}$ of initial trains may reside in the station infrastructure at the start of the planning horizon. These are all available from the start of the planning horizon h_0. The total set of trains is denoted by $\mathcal{T} = \mathcal{I} \cup \mathcal{A}$. We define the availability time of a train $t \in \mathcal{T}$ as $startTime_t$ where $startTime_t = arrTime_t$ if t is part of an arrival and $startTime_t = h_0$ if t is an initial train (i.e. in $\mathcal{I}$). Note, that h_0 denote the start of the planning horizon.

Some departures are the beginning of a tour that returns the train units to the terminal station as arrivals within the same planning horizon. If an arrival is linked to an earlier departure we call it a *linked* arrival. Likewise we call the earlier departure a *linked* departure. In order to avoid confusion, we note that the "linked" concept does *not* refer to physical train units that are coupled together to form a convoy. A linked arrival states that the arriving train is the same train that was assigned to the corresponding (earlier) linked departure. In contrast to a non-linked arrival, this means that the arriving train is the same physical train which was assigned to the linked departure. If the linked departure is canceled, then a new replacement train arrives instead. The linking between arrivals and departures is important because it means that the properties of (linked) arrivals are not known, without knowing what trains (if any) are matched to those arrivals' linked departures. We define the set $\mathcal{L} \subseteq \mathcal{A}$ as the set of arrivals which have an earlier linked departure, and define $\sigma(a)$ as the linked departure of the arrival train $a \in \mathcal{L}$.

Every train $t \in \mathcal{T}$ belongs to some category $cat_t \in \mathcal{C}$ that defines common characteristics such as train length, capacity and maintenance durations. Two trains of the same category are considered interchangeable when assigning trains to departures – with one exception. All trains in the system are subject to two types of maintenance constraints: Distance Before Maintenance (DBM) and Time Before Maintenance (TBM). Each train t has some initial *remaining DBM* $(remDBM_t)$ and *remaining TBM* $(remTBM_t)$. Every departure $d \in \mathcal{D}$ has a *required DBM* $(remDBM_d)$ and a *required TBM* $(remTBM_d)$ to perform the round-trip. If a train is to be matched to departure d, then it must have sufficient DBM and TBM.

A train may visit a maintenance facility at the station between arriving and departing, which in the RSUM problem takes a fixed amount of time and resets both the DBM and TBM to their maximum value for the train, and we include the decision of whether or not to perform maintenance in the DMP. The constants $maxDBM_t$ and $maxTBM_t$ indicate the level of DBM and TBM that is obtained if a maintenance operation is performed on train t. For trains of the same category $(i \in \mathcal{C})$ the constants have identical values, and we can therefore use the notation $maxDBM_i$ and $maxTBM_i$ without ambiguity. Due to a limited amount of manpower at the maintenance facilities the total number of maintenance operations per day is limited to a constant of $maxMaint \in \mathbf{Z}^+$. The imposed limit means that certain combinations of matches can not all be made: if there are $n > maxMaint$ otherwise independent matches that would all occur on the same day and all require maintenance, at most $maxMaint$ of them can be made. The remaining $(n - maxMaint)$ trains could not be maintained and would therefore not have sufficient DBM and TBM to be matched to the $(n - maxMaint)$ departures.

For those arrivals that have a linked departure, category, DBM and TBM are dependent on any matching made to that linked departure. An arrival $a \in \mathcal{L}$ is linked to a previous departure $d \in \mathcal{D}$, i.e., the train $t \in \mathcal{T}$ assigned to departure d is the same train arriving later in a. The train in a therefore inherits the remaining DBM and TBM and category of train t. In the case where no train is assigned to d then another new train arrives with its own specified remaining DBM, TBM, and category.

Every train $t \in \mathcal{T}$ can only be matched with a limited set of departures. We define $CompDep(t)$ as the set of departures that are compatible with t. Likewise we define $CompTr(d)$ as the set of trains that are compatible with departure $d \in \mathcal{D}$. These two sets are considered as parameters to the problem, and it is up to the end-user to specify which factors to consider. These factors could include for example a minimum routing time, a maximum time between arrival and departure, or a maximum number of potential matchings for any given train. For the sake of simplicity, we define only a few simple rules for possible matchings. Given a departure $d \in \mathcal{D}$ and a non-linked arrival train or initial train $t \in \mathcal{T}$ a matching is possible if the following conditions are satisfied:

$$\begin{gathered} cat_t \in compCatDep_d \\ startTime_t + maintenance_t > depTime_d \\ \max\{remDBM_t, allowDBM_t \cdot maxDBM_t\} \geq reqDBM_d \\ \max\{remTBM_t, allowTBM_t \cdot maxTBM_t\} \geq reqTBM_d \end{gathered}$$

where the binary parameters $allowDBM_t$ and $allowTBM_t$ indicate whether DBM and TBM are allowed to be performed. In some cases there is only time for one of the two operations but not both. The constant $maintenance_t$ indicates the time needed to perform the necessary maintenance operations for train t, or zero if no maintenance is required.

Given a departure $d \in \mathcal{D}$ and a linked arrival $t \in \mathcal{L}$ it is harder to limit the options beforehand. In the preprocessing it can only be restricted by $startTime_t > depTime_d$ as the category, remaining DBM and remaining TBM of the linked arrival t are unknown.

In practice it is in some cases expected that certain arrivals are matched with specific departures. If such a match is successful we call it a *train reuse*, otherwise it is a missed train reuse. We denote $\mathcal{U}$ as the set of train uses, where tr_u and dep_u define the train $t \in \mathcal{T}$ and departure $d \in \mathcal{D}$ for a reuse $u \in \mathcal{U}$.

Definition 9.1 (The DMP definition) We define the DMP as the problem of finding a feasible matching between trains and departures that respects the

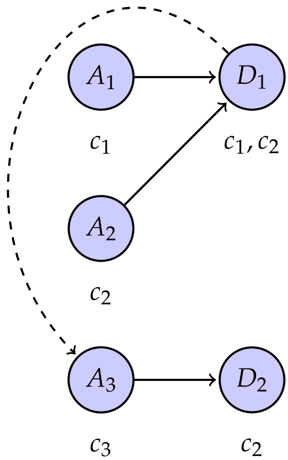


Figure 9.1: An illustration of the matching problem showing potential matches between three arrival trains (A_1, A_2, A_3) and two departures (D_1, D_2). Each arrival has a train category (c_1, c_2, c_3) and each departure has one or more acceptable train categories. Departure D_1 and arrival A_3 are linked; arrival A_3 has replacement category c_3 only if departure D_1 is unmatched, but instead has category c_1 or c_2 depending on which arrival is matched to D_1. The only matching of cardinality 2 is $\{(A_2, D_1), (A_3, D_2)\}$; matching A_1 with D_1 precludes matching A_3 with D_2 as the category of arrival A_1 is incompatible with departure D_2.

departure maintenance requirements, the departure train category compatibility, the time required to perform the needed maintenance, and the total number of maintenance operations per day. The objective of the DMP is to minimize the number of uncovered departures and to maximize the number of train reuses. In every instance of the problem there is a fixed penalty for every missed train reuse and a fixed penalty for every uncovered departure.

Figure 9.1 shows a small example with three arrival trains (A_1, A_2, A_3) and two departure trains (D_1, D_2), considering three different train categories having no maintenance requirements or other restrictions. In this example departure D_1 and arrival A_3 form a *linked arrival* pair: whatever is matched to departure

D_1 returns as arrival A_3. Arrival train A_3 is in $\mathcal{L}$, and its linked departure is D_1; that is, $\sigma(A_3) = D_1$. In the figure, arrival A_3 has category $c3$ marked as its *replacement train* category. That is, it *only* has category c_3 if no match is made to departure D_1, but if instead some match is made, the category of A_3 is inherited from that match. The match between arrival A_3 and departure D_2 is not independent of other matchings made; it is only feasible if a match is also made between arrival A_2 and departure D_1. If D_1 is unmatched, arrival A_3 will have the category of its replacement train, incompatible with departure D_2. Similarly, if D_1 is matched to arrival A_1 then arrival A_3 will inherit the category of arrival A_1, also incompatible with departure D_2.

9.3 Related problems

The RSUM as defined for the ROADEF competition is based on real station infrastructure and problems, with certain simplifications to make it appropriate for the competition, such as simplifications of the switching and yard infrastructure. Real train stations face similar problems, though those problems differ in specific details. A matching problem similar to the DMP subproblem (that we have identified) could also exist as a subproblem at stations, though it may not necessarily be treated as a self-contained subproblem.

If the linking between some departures and later arrivals is ignored or not present, and performing maintenance is ignored, then whether or not a match is possible would be pre-determinable. If the problem was just that of minimizing the number of uncovered departures, then it would be a relatively simple maximal bipartite matching problem, solvable in polynomial time Hopcroft and Karp [1973]. If minimizing a weighted sum of uncovered departures and missed reuses, the problem could be formulated as an assignment problem, also solvable in polynomial time Kuhn [1955].

The presence of linked arrivals inherently changes the structure of the problem. The matchings themselves are not independent because whether or not some train can be matched to some departure can depend on what *other* train is matched to some *other* departure. Similarly, the ability to perform maintenance changes which matches are possible, and the restriction of the maximum number of maintenance operations per day makes matches non-independent. The DMP combines the matching with components of the RSUM that we have identified as being closely related to and significantly interacting with the matching, without including so many aspects of the RSUM to make the problem intractable. For example, in the RSUM problem the restriction of the maximum number of maintenance operations per day means that some sets of potential matches can not all be included, and maintenance decisions should be included with the matching subproblem. If however there was no restriction on the maximum number of maintenance operations per day then perhaps a subproblem that ignores maintenance could be sufficient to provide feasible or optimal solutions, relying on it always being possible to perform maintenance if necessary. For some other similar station arrival problem, a similar but distinct problem to the DMP might instead be identified as a subproblem.

Freling et al. identify the subproblem of matching departures and arrivals as part of a shunting problem Freling et al. [2005]. The authors formulate a matching subproblem that considers the unattractiveness of producing matches that require breaking up trains into units matching different departures. In contrast, in the DMP we do not include any cost or penalty for matches which require coupling or decoupling. The authors do not describe anything similar to linked arrivals or maintenance decisions and daily restrictions, which are the features of our problem that make matches interdependent.

Kroon et al. identify a matching subproblem as part of a larger station shunting problem Kroon et al. [2008]. However there is no analogue to the linked arrivals of the DMP. The authors do not solve the matching problem in isolation but as part of a larger formulation that includes shunting features that are not part of the DMP or even necessarily part of the RSUM.

In the shunting literature there are many problems that share similarities with the RSUM problem and potentially have a subproblem that is very similar to the DMP. However shunting is not necessarily an important component of the ROADEF challenge because the station infrastructure has large and simplified “yard” resources. These are abstractions with only a maximum capacity, but train units can be parked in or removed from the yards without considering their ordering.

$$\max \sum_{i\in\{1,2,3\}} p_1 x_{i1} + p_2 x_{i2} + p_3 x_{i3} + p_4 x_{i4}$$

$$w_1 x_{11} + w_2 x_{12} + w_3 x_{13} + w_4 x_{14} \leq c_1$$
$$w_1 x_{21} + w_2 x_{22} + w_3 x_{23} + w_4 x_{24} \leq c_2$$
$$w_1 x_{31} + w_2 x_{32} + w_3 x_{33} + w_4 x_{34} \leq c_3$$

$$\sum_{i\in\{1,2,3\}} x_{ij} \leq 1 \quad \forall j \in \{1,2,3,4\}$$
$$x_{ij} \in \{0,1\}$$

Figure 9.2: Multiple Knapsack Problem instance with 4 (coloured) items and three knapsacks.

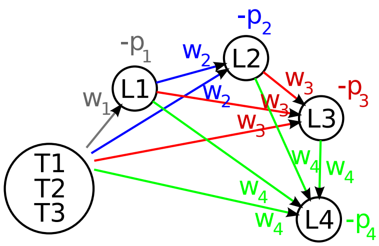


Figure 9.3: The constructed Train Matching Problem instance from the Multiple Knapsack Problem instance shown in Figure 9.2. Each train ($T1, T2, T3$) corresponds to a single knapsack, and each linked departure ($L1, L2, L3, L4$) corresponds to one item. For the sake of clarity, initial trains have been grouped together since the graph is identical for each train.

9.4 NP-hardness

It is relatively simple to prove that the DMP is NP-hard by reduction from the Knapsack Problem. However, since the classic version of the Knapsack Problem is NP-hard in the weak sense, we prove the same by reduction from the 0-1 Multiple Knapsack Problem (MKP). We adopt the definition of the MKP of Kellerer et al. [2004]:

Definition 9.2 Given a list of items to pack, each with a profit p_i and weight w_i, and one or more knapsacks of capacity c_j. The 0-1 Multiple Knapsack Problem is the problem of choosing a list of items for every knapsack such that the profit of the selected items is maximized while respecting the weight-restriction of every knapsack.

Note that the 0-1 variant only allows each item to be packed at most once. We show NP-completeness by reduction from a MKP instance to a DMP instance with a simple mapping from knapsacks to trains and items to departures. All physical train units have a DBM constraint that must be respected, this will be the map to the capacity of a knapsack. All departures assigned to a train consume some level of DBM and a profit/penalty is achieved/given, this corresponds to putting an item into the knapsack. The weight corresponds to the required level of DBM.

An illustrative example of the transformation is shown in Figure 9.2. The knapsack example contains four items and three knapsacks with individual knapsack capacities where every item can be packed at most once. The resulting DMP instance is shown in Figure 9.3.

Consider a MKP instance with N items and M knapsacks, let the weights and profits respectively be defined as w_i and p_i for every item i, and finally the capacities as c_j for every knapsack j.

We construct a DMP instance with M initial (or arrival) trains and N linked departures. Each individual train has remaining DBM corresponding to one of the capacities c_j of a knapsack. We set the remaining TBM to some high value such that this constraint is never binding. The departure and arrival times are set such that any of the four trains could be matched to any of the four linked departures, provided that it has sufficient remaining DBM. Any train can also visit any subset of the departures. The cost of matching every respectable (linked) departure is set to the negative profit of the item corresponding to each departure in the MKP instance. The cost of cancelling a departure is set to zero, and w.l.o.g. no maintenance is allowed. We can ensure that no maintenance will be performed by enforcing zero allowed maintenance operations, or by setting a high cost on maintenance, or by setting the replenished value to zero after maintenance. No train-reuse costs are defined. The initial trains are of the same single category, and all departures (of the linked departures) are compatible with only that train type. The replacement train of the linked arrivals have a different (incompatible) train type, and therefore do not influence the solution or quality in any way. Thus it is only possible for the initial trains to be matched to any of the linked departures. The optimal solution of the DMP instance will therefore only match the linked departures with the physical trains starting as the specified initial trains. Further, the matching-sequences are restricted by the given capacities of the MKP instance, where every matching/item consumes the specified knapsack weights. Due to the departure constraint, every linked departure can be matched at most once, thus every item is at most assigned to one knapsack. If any train is matched to linked departure i, $-p_i$ is added to the objective and inflicts a consumption ($reqDBM_i$) of w_i to the DBM of the train. As no costs, but the profits of the MKP instances are present, the optimal solution value will correspond to the optimal value of the MKP instance, only with a negative sign. Alternatively, in order to more closely follow the DMP objective function we adopt in later sections, the profit can be reformulated in terms of minimizing the cost of cancellations. A item of the MKP instance is only picked in a particular knapsack iff the corresponding linked departure is matched in the DMP instance to a particular train (initially, or as a returning linked arrival of that train). In conclusion, by transforming (in polynomial time and resources) an instance of the MKP is solved by solving the constructed DMP instance. The transformation graph is polynomial in the number of vertices and edges. The number of vertices is $1 + M$. The graph is acyclic and every node only connects forward in time (with respect to arrival and departure times) which results in a total of $\sum_{k=1}^{N} k = 1/2(N^2 + N)$ edges.

Theorem 9.3 *The DMP is NP-hard*

Proof. It is straightforward to verify whether a solution to the DMP is feasible or not. This is verified by checking that the DBM and TBM constraints are valid, that the train types are compatible, and that the matched arrival and departures times are respected by the train units. Likewise, it is simple to calculate the objective cost of any given feasible solution. The number of operations and resources used to construct the DMP instance is polynomial. The number of vertices and edges created are polynomial in the MKP instance size. By reduction from the MKP, as described in the example, we have argued that there is a one-to-one correspondence between the solution of the constructed DMP instance and the MKP instance. Any feasible train matching solution to the DMP instance is a feasible item selection for the MKP instance, and vice versa.

9.5 Mixed integer program model

In this section we present a MIP mathematical model for the DMP. The size of the proposed MIP model for the DMP is polynomial in the number of input trains and departures, and can be solved using a commercial solver.

The model contains six types of variables. A set of binary variables m_t^d determine whether train $t \in \mathcal{T}$ is matched to departure $d \in \mathcal{D}$. Matches that are not present in the compatibility set ($CompDep(t)$ or $CompTr(d)$) are omitted or fixed to zero. We also introduce a set of binary variables cat_t^i that indicate if train $t \in \mathcal{T}$ is of category $i \in \mathcal{C}$. For all initial trains and non-linked arrival trains the category is known and the corresponding variable can be fixed (or omitted). The continuous variables dbm_t and tbm_t determine the DBM and TBM of train $t \in \mathcal{T}$ at the time before departure, which is the initially available DBM/TBM or $maxDBM_t/maxTBM_t$ if maintenance is performed on the train. Finally we introduce the binary variables f_t^d and g_t^d that determine whether a train $t \in \mathcal{T}$ matched to $d \in \mathcal{D}$ is being maintained on DBM or TBM. In the following we will assume that maintenance of DBM and TBM can only be done on one specific known day, for a particular matching. We introduce a binary parameter $\omega_{t,day}^d$ that is 1 if a match between t and d would perform maintenance on day (if it performs it at all), and 0 otherwise. The objective is formulated as a minimization of the number of unmatched departures and the cost of missed train reuse:

$$\begin{aligned} \min \sum_{d \in D} & cancellationCost_d \cdot c_d \\ & - \sum_{u \in \mathcal{U}} reuseCost \cdot m_{tr_u}^{dep_u} \\ & + reuseCost \cdot |\mathcal{U}| \end{aligned}$$

We minimize the number of cancelled trains and maximize the number of reuses. Note that the final term is constant and can be left out. The constraints of the model are the following:

$$\sum_{t \in \mathcal{T}} m_t^d \geq 1 - c_d \qquad d \in \mathcal{D} \tag{9.1}$$

$$\sum_{t \in \mathcal{T}} m_t^d \leq 1 \qquad d \in \mathcal{D} \tag{9.2}$$

$$\sum_{d \in \mathcal{D}} m_t^d \leq 1 \qquad t \in \mathcal{T} \tag{9.3}$$

$$\sum_{t \in \mathcal{T}} \sum_{d \in \mathcal{D}} \left(\omega_{t,day}^d f_t^d + \omega_{t,day}^d g_t^d \right) \leq maxMaint \qquad day \in \mathcal{H} \tag{9.4}$$

$$f_t^d + g_t^d \leq 2 m_t^d \qquad t \in \mathcal{T}, d \in \mathcal{D} \tag{9.5}$$

Constraints (9.1) ensure that every departure is assigned (to some train), unless there is a cancellation. Constraints (9.2) ensure that at most one train is assigned to every departure. Constraints (9.3) ensure that each train is assigned at most once. Constraints (9.4) ensure that the total number of maintenance operations (every day) is respected. Constraints (9.5) prohibit maintenance usage on a match if that match is not made.

$$\sum_{i \in C} cat_t^i = 1 \qquad t \in T \tag{9.6}$$

$$m_t^d \leq \sum_{i \in compCatDep_d} cat_t^i \qquad t \in \mathcal{T}, d \in \mathcal{D} \tag{9.7}$$

$$cat_t^i \geq cat_{t'}^i + m_{t'}^{\sigma(t)} - 1 \qquad t \in \mathcal{T}, t' \in \mathcal{T} \setminus \{t\}, i \in \mathcal{C} \tag{9.8}$$

$$cat_t^{i_t} \geq c_{\sigma(t)} \qquad t \in \mathcal{T} \tag{9.9}$$

Constraints (9.6) ensure that every train is assigned exactly one category. Again, the constraint for initial trains and non-linked arrival trains can be omitted, if the correct value is fixed. Constraints (9.7) ensure that trains cannot be assigned to departures where the category is not compatible. Constraints (9.8) ensure that if train t' is matched to the linked departure $\sigma(t)$ of train t, then train t inherits the category i of train t'. Finally, constraints (9.9) ensure that, if the linked departure $\sigma(t)$ of train t is not covered, then train t has the category i_t of its replacement train.

$$\begin{aligned} dbm_t \leq & \sum_{t' \in \mathcal{T} \setminus \{t\}} \left(dbm_{t'}^{\sigma(t)} - reqDBM_{\sigma(t)} \right) \cdot m_{t'}^{\sigma(t)} \qquad t \in \mathcal{T} \\ & + c_{\sigma(t)} \cdot remDBM_t \end{aligned} \tag{9.10}$$

$$+ \sum_{d \in \mathcal{D}} f_t^d \cdot M$$

$$dbm_t \leq \sum_{i \in \mathcal{C}} maxDBM_i \cdot cat_t^i \qquad t \in \mathcal{T} \qquad (9.11)$$

$$0 \leq dbm_t - m_t^d \cdot reqDBM_d \qquad t \in \mathcal{T}, d \in \mathcal{D} \qquad (9.12)$$

Constraints (9.10) ensure that the available DBM for a train t is correct. The right hand side consists of three terms. The first term counts the contribution from trains that are linked to t. If the train is cancelled the term sum is zero, and the second term will contribute with $remDBM_t$ which is the *new* train inserted in case of a cancellation of departure $\sigma(t)$. The third term makes the constraint non-binding if maintenance is performed; the constant M is a big number that is no less than the maximum of $maxDBM_i$ for all $i \in \mathcal{C}$. For the sake of clarity the constraints have been presented using a non-linearity in the first term. In order to maintain a linear model we replace the constraints with linear constraints described in section 9.5.1. Since the train category i of a train t might be unknown we further constraint the DBM of the train to respect the $maxDBM_i$ in Constraints (9.11). Constraints (9.12) make sure that enough DBM is available for a matching. For all matchings that are not active the constraint is just requiring dbm_t to be non-negative.

We will not present the corresponding constraints for TBM since they are analogous to Constraints (9.10), (9.11) and (9.12).

9.5.1 Reformulating the nonlinear constraints

The first term on the right hand side of Constraints (9.10) is non-linear and can be remodeled as a linear constraint by adding one more group of continuous variables and additional constraints. We introduce one new auxiliary variable κ_t^{dbm} for each train t that measures the DBM contribution of the train linked to departure $\sigma(t)$. We replace Constraints (9.10) with the following set of constraints:

$$dbm_t \leq \kappa_t^{dbm} + c_{\sigma(t)} \cdot remDBM_t + \sum_{d \in \mathcal{D}} f_t^d \cdot M \qquad t \in \mathcal{T} \qquad (9.13)$$

$$\kappa_t^{dbm} \leq dbm_{t'} - m_{t'}^{\sigma(t)} \cdot reqDBM_{\sigma(t)} \qquad t \in T, t' \in \mathcal{T} \setminus \{t\} \qquad (9.14)$$

$$+ \sum_{d \in \mathcal{D} \setminus \sigma(t)} m_{t'}^d \cdot M + \sum_{t'' \in \mathcal{T} \setminus t'} m_{t''}^{\sigma(t)} \cdot M$$

$$\kappa_t^{dbm} \leq M \cdot (1 - c_{\sigma(t)}) \qquad t \in T \qquad (9.15)$$

The three terms of Constraints (9.13) are analogous to the three terms of Constraints (9.10), except the first non-linear term is replaced with the new contribution term κ_t^{dbm}. Constraints (9.14) have four terms defining an upper bound on the linked contribution for train t. The first two terms are relevant if train t' is matched to the linked departure $\sigma(t)$ and ensure the contribution is no greater than the difference between the DBM for t' and the required DBM for departure $\sigma(t)$. The third term loosens any bound on κ_t^{dbm} related to t' if t' is matched to some departure other than $\sigma(t)$. The fourth term loosens the bound on κ_t^{dbm} related to t' if some other train t'' is matched to $\sigma(t)$. Finally, Constraints (9.15) make sure that the contribution is zero if departure $\sigma(t)$ is canceled.

9.5.2 Reducing the model

In order to simplify (and streamline) the model description we treated every train as a linked arrival train. However, some arrival trains are not linked, and some trains are initial trains already in the system. For both types of trains the constraints can be simplified, and for an instance with few linked arrivals the model may be reduced substantially. The variables cat_t^i decide the category for a train t. For initial trains and non-linked arrivals the category is set as a problem parameter, and so for such trains Constraints (9.6) are not necessary. Constraints (9.7) can be removed and replaced by setting the upper bound to zero for any departure for which the known category of t is incompatible. Constraints (9.8)-(9.9) are unnecessary.

A non-linked arrival or an initial train t has no DBM or TBM contribution from any previous train. Constraints (9.14) can have the κ_t^{dbm} term removed, and we can set $c_{\sigma(t)} = 1$ (because instead train t will always take its own remaining DBM). Constraints (9.14)-(9.15) are unnecessary in such cases.

9.6 Column generation model

The formulation presented in section 9.5 models every initial and arrival train individually even if some of them constitute a sequence of linked arrivals and departures and therefore essentially represents a single physical train unit. In this section we present a formulation that models each physical train unit using a single variable. The number of possible variables grows more than exponentially by the number of present linked arrivals. Initially every train can be matched to any linked departure (i.e. departure with a linked arrival), and continue to any other linked departure, before it finally reaches its final departure. The number of potential paths is bounded by $O(|T| \cdot |L|! \cdot |D|)$, where $|L|$ is the number of linked arrivals/departures. We therefore propose a column generation approach for solving the model. Column generation is a well-described technique used with success for solving MIP problems, e.g. the Vehicle Routing Problem with Time Windows (VRPTW) Barnhart et al. [1998], Kallehauge et al. [2005], Lübbecke and Desrosiers [2005]. It is assumed that the reader is familiar with column generation solution methods.

The model contains two types of variables. For every departure $d \in \mathcal{D}$ we introduce a binary variable c_d that indicates whether d is canceled or not. For every possible train unit pattern $p \in \mathcal{P}$ we have a binary variable λ_p that indicates whether pattern p is used or not. A pattern represents a sequence of linked arrivals and departures. The objective is formulated as a minimization of the number of unmatched departures and cost of missed train reuses :

$$\begin{aligned}\min \sum_{d \in D} & cancellationCost \cdot c_d \\ & - \sum_{u \in \mathcal{U}} \sum_{p \in \mathcal{P}} reuseCost \cdot \alpha_p^{dep_u} \cdot \beta_p^{arr_u} \cdot \lambda_p \\ & + reuseCost \cdot |\mathcal{U}|\end{aligned} \tag{9.16}$$

$$\sum_{p \in \mathcal{P}} \alpha_p^d \lambda_p \geq 1 - c_d \qquad d \in \mathcal{D} \tag{9.17}$$

$$\sum_{p \in \mathcal{P}} \alpha_p^d \lambda_p + \sum_{p \in \mathcal{P}} \phi_p^d \lambda_p \leq 1 \qquad d \in \mathcal{D} \tag{9.18}$$

$$\sum_{p \in \mathcal{P}} \beta_p^t \lambda_p + \sum_{p \in \mathcal{P}} \varphi_p^t \lambda_p \leq 1 \qquad t \in \mathcal{T} \tag{9.19}$$

$$\sum_{p \in \mathcal{P}} maint_p^{day} \cdot \lambda_p \leq maxMaint \qquad day \in \mathcal{H} \tag{9.20}$$

$$\lambda_p \in \{0,1\} \qquad p \in \mathcal{P} \tag{9.21}$$

$$c_d \in \{0,1\} \qquad d \in \mathcal{D} \tag{9.22}$$

The α_p^d is a binary coefficient that indicates whether departure d is covered by pattern p. The β_p^t is a binary coefficient that indicates whether train t is used by patten p. Therefore $\alpha_p^{dep_u}$ indicates whether departure $dep_u \in \mathcal{D}$ is covered by pattern p, and $\beta_p^{arr_u}$ whether train $dep_u \in \mathcal{T}$ is used by p. Finally, ϕ_p^d and φ_p^t are binary coefficients that indicate whether a departure d or train t is blocked as a result of pattern p. A train is blocked by a pattern if the final matching of the pattern ends (or terminates) on a departure that is linked to some arrival. No other pattern may use this arrival-train, as it would mean that two patterns are using the same physical train without keeping proper score on train type, DBM and TBM. A departure is likewise blocked by a pattern if it starts using a train of a linked arrival. This corresponds to using one of the replacement trains which assumes (or requires) that the linked departures was canceled. Essentially this means that the departure must be blocks if such a pattern is used. Finally, the coefficient $maint_p^{day} \in \mathbf{Z}^+$ indicates the number of maintenance operations performed on day day using pattern p.

Constraints (9.17) ensure that every departure is assigned (to some train), unless there is a cancellation. Constraints (9.18) ensure that at most one train is assigned to every departure, and also blocks departures for trains that assume that the departure is cancelled. Constraints (9.19) ensure that each train is assigned at most once. Trains are blocked by patterns if they use the corresponding linked departure. Constraints(9.20) ensure that the total number of maintenance operations (every day) is respected. For comparability it is here assumed that, given a matching (t,d), the day that a maintenance operation is performed is fixed. The day of maintenance operations can however be made a choice in the subproblem without difficulty. Finally, Constraints (9.21) and (9.22) show the variable domains. Note that Constraints (9.22) can be relaxed as Constraint (9.17) ensures that the variables are *naturally* binary in all feasible solutions.

The number of variables in the model is exponential by the number of linked departures, however compared to the MIP model present earlier the number of constraints is reduced to $O(|\mathcal{D}| + |\mathcal{T}| + |\mathcal{H}|)$.

For obtaining the optimal *integer* solution to this problem (using column generation) a Branch and Price (B&P) framework can be adopted. In our solution method we will however only add columns in the root node and use Branch and Bound (B&B) for finding the best solution using the generated columns. In general this produces solutions of high quality, but no guarantee of optimality can be given in general. The gap between the found solution and the LP solution (of the root node) gives a bound on the optimality gap. It is left as future research to develop a full B&P framework.

9.6.1 Column generation subproblems

We distinguish between generating two families of columns. The first family of variables consists of patterns containing only one train-to-departure matching. It is relatively easy to enumerate all choices in this family and produce columns with the most negative reduced costs. In our implementation we pre-generate all columns of this family. Therefore, for the next family of variables, assume that the following patterns consist of at least two train-to-departure matches.

The subproblem can be split into one subproblem per train category. This transformation will both simplify and reduce each individual problem as the number of compatible departures is lower. The complexity of a labeling algorithm is also reduced since it is no longer needed to keep track of the train category when extending arcs. An additional advantage is that all subproblems can then be solved in parallel.

The subproblem consists of solving a Resource Constrained Shortest Path Problem (RSCPP). The underlying graph consists of one node per linked arrival in addition to one source and one sink node, see Figure 9.4. The arcs constitute matching choices. Three types of arcs are added. First, arcs originating from the source to every node in the graph represent compatible trains that are matched to the linked departure of the node. Second, arcs are added between nodes that represent compatible linked continuations, i.e., linked arrival/departures that connect to another linked arrival/departure. An example: in the (s, a, d, t) path the departure of the initial train matching (represented by arc (s, a)) is linked to an arrival (node a). The departure of the next matching (arc (a, d)) is connected to another linked arrival (node d). The departure of the last matching is not linked to any arrival, and thus the sequence ends. As described earlier, some patterns (represented by the path) may block other trains or departures.

More formally, we construct a directed non-cyclic graph $G(V, A)$ for every category $c \in \mathcal{C}$. Let $v_\pi \in V$ denote the *source* vertex, and $v_\omega \in V$ denote the *sink* vertex. Finally, we construct one vertex $v_{la} \in V$ per compatible linked arrival $la \in \{a \in \mathcal{L} | c \in compCatDep_a\}$. The arcs in the graph represents a train and departure match. For every train t of category c ($\{t \in \mathcal{T} | cat_t = c\}$) an arc $a_{(\pi,la)} \in A$ connects v_π and to v_{la} if the distance between availability time of t and departure time of departure $\sigma(la)$ is sufficient[2]. Likewise for

[2]This could depend on multiple factor such as expected/minimum routing time, maintenance

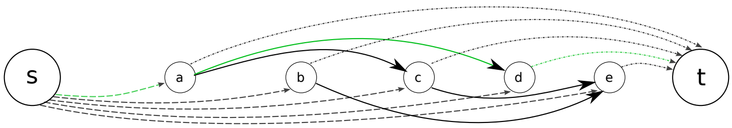


Figure 9.4: An illustration of the underlying graph for the matching subproblem. Every arc represents a match between a train and a departure. A feasible path (or linked sequence pattern) starts in the source s and ends in the sink/target t traversing a set of arcs. The feasible path (s, a, d, t) is shown in green. A linked sequence can start (or terminate) with many different trains (or departures) thus the dashed arcs illustrate where multiple arcs exist with the same origin and destination vertex.

every departure d compatible with category c ($\{d \in \mathcal{D} | c \in compCatDep_d\}$) an arc $a_{(ld,\omega)} \in A$ connects vertex v_{la} with v_ω, again only if the necessary time is available. Finally for every pair of linked arrivals (la_1, la_2) an arc $a_{(la_1,la_2)}$ connects the train of la_1 to departure $\sigma(la_2)$ if it respects the required time. No edge directly connects the v_π to v_ω.

Theorem 9.4 *Any path originating in π and terminating in ω represents a feasible matching.*

Proof. Every path consists of an initial arc and a terminating arc, and optionally multiple intermediate arcs. All arcs represent matchings that are possible to make. Since we only have one category per subproblem, the construction of the graph inherently ensures that the train category is compatible with the assigned departures. Since we assume that maintenance can be performed on any matching, the required DBM and TBM can be fulfilled.

Theorem 9.5 *All possible matchings of at least two departures are valid paths in the constructed graph, originating in π and terminating in ω.*

Proof. All feasible matchings, w.r.t. category and time, are present in the graph. All initial matchings are represented as arcs originating in the source π. All intermediate matchings are present as arcs connecting the non-terminal nodes. Finally, all possible terminations of linked sequences are represented as arcs terminating in the sink ω.

---

time and dwell time. For the sake of simplicity and comparability reasons we only require that the train availability time is less (or equal) to the departure time.

The subproblem can now be defined as the problem of finding the minimum cost path originating from v_π and to v_ω, given some edge costs, maintenance costs and restrictions. Every edge has a primal cost corresponding to the objective in the master problem, i.e., zero if no train reuses are satisfied or $-(n \cdot reuseCost)$ cost if n reuses have been satisfied by the path. A dual cost also appears on every edge that depends on the dual values given after solving the master problem in each iteration. The dual value of (9.17) and (9.18) are added to all arcs that include the corresponding departure. The dual of (9.19) is added to arcs including the corresponding train. The maintenance restrictions relate to the DBM and TBM restrictions. These values must be positive at all times, and all edges either increase or decrease these values. A path starts with values of 0 for DBM and TBM. All arcs originating from the source increase the values as indicated by the $remDBM$ and $remTBM$ on the train (of the arc). Other edges only decrease the values as indicated by $reqDBM$ and $reqTBM$ of the departures. Before extending an arc maintenance can be performed, which replenishes the DBM and/or TBM levels, but this comes at a cost of the corresponding dual of (9.20).

The problem is a RSCPP due to the maintenance constraints. In addition to the objective coefficients, the duals from Constraints (9.17) and (9.18) are added to arcs of the corresponding departure, and the duals from Constraint (9.19) are added to arcs of the corresponding trains. The appropriate dual from Constraints (9.20) is added every time a maintenance operation is scheduled.

The subproblem can be formulated as a mathematical problem:

$$min \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{(i,j)\in A} \beta_{ij}y_{ij} + \sum_{(i,j)\in A} \alpha_{ij}z_{ij} \tag{9.23}$$

$$s.t. \sum_{(i,j)\in\delta^+(\pi)} x_{ij} = 1 \tag{9.24}$$

$$\sum_{(i,j)\in\delta^-(\omega)} x_{ij} = 1 \tag{9.25}$$

$$\sum_{(i,j)\in\delta^+(v)} x_{ij} = \sum_{(i,j)\in\delta^-(v)} x_{ij} \qquad \forall v \in V \setminus \{\pi, \omega\} \tag{9.26}$$

$$\sum_{(i,j)\in\Pi_v} reqDBM_{ij} \cdot x_{ij} + maxDBM \cdot y_{ij} \geq 0 \qquad \forall v \in V \tag{9.27}$$

$$\sum_{(i,j)\in\Pi_v} reqTBM_{ij} \cdot x_{ij} + maxTBM \cdot z_{ij} \geq 0 \qquad \forall v \in V \tag{9.28}$$

$$x_{ij} \in \{0,1\} \quad y_{ij} \in \{0,1\} \quad z_{ij} \in \{0,1\}$$

Where $\delta^+(v)$ and $\delta^-(v)$ respectively denote the outgoing and ingoing edges of vertex v. The binary variables x_{ij} define whether flow is used on edge (i,j) in shortest path, and y_{ij} and z_{ij} respectively determine whether DBM and/or TBM is performed on arc (i,j). Constraints (9.24)-(9.26) constitute the flow-conservation in a shortest path formulation. Constraints (9.27) and (9.28) ensure that the sufficient DBM and TBM is available - these levels can never be negative. Note that the graph is acyclic which means that we know all possible edges that can appear before reaching any vertex v - we denote Π_v as the set of all such edges. The constraints thus ensure that if a matching is made, then the DBM/TBM levels on any edge (leading up to v) must be non-negative.

9.6.1.1 Labeling algorithm

As an alternative to solving the mathematical model of the subproblem directly, we propose a dynamic programming approach for finding the optimal paths for the resource constrained shortest path. We refer to Irnich Irnich [2008] for a more in-depth description of a this topic.

The labeling algorithm is similar to a shortest path algorithm that uses full enumeration, e.g. using a Breath First Search (BFS) strategy, to find the minimum cost path. In addition, we also need to respect some side-constraints. In our method, a label is a partial path from the source to some intermediate vertex that also keeps track of the total reduced cost, remaining DBM, remaining TBM and performed maintenance. Every arch has a primal and dual cost and a required level of DBM and TBM.

Initially, we generate the empty label at the source. In every iteration of the labeling algorithm, we pick one label at some vertex v and extend it. When extending we generate new labels from every outgoing edges from v. Due to the possibility of performing maintenance, we generate multiple labels for every outgoing edge: one that does not perform any maintenance, one that only performs DBM maintenance, one that only perform TBM maintenance and one that performs both DBM and TBM maintenance. Labels are not extended if either DBM or TBM becomes negative when subtracting the required level of DBM and TBM since these represent prefixes of infeasible paths.

In order to reduce computational time, we introduce dominance rules. A label a is said to dominate another label b, if we can safely remove b without loosing optimality. By removing a label b we omit searching all paths that follow the matching pattern of b.

Theorem 9.6 *A label ($cost_a$, $remDBM_a$, $remTBM_a$) at vertex v dominates ($cost_b$, $remDBM_b$, $remTBM_b$) at vertex v if $cost_a \geq cost_b$, $remDBM_a \geq remDBM_b$ and $remTBM_a \geq remTBM_b$.*

PROOF. The first label has lower cost and more DBM and TBM remaining. The second label cannot have any advantages in terms of future matchings or maintenance costs. Thus, no matter how the path continues from v, the first label will always be at least as good as the second label.

Finding the optimal path in the subproblem can be both time and space consuming. Since it is sufficient to find any one path with negative reduced cost, we will rely on generating heuristic columns initially. Only when no heuristic columns (with negative reduced cost) can be found, we solve using the exact labeling algorithm.

We adopt two variations of the full labeling algorithm to find heuristic negative reduced cost paths. The first variant allows no DBM nor TBM maintenance which efficiently limits the number of possible matchings. The second variant only allows at most one DBM and at most one TBM maintenance. This is motivated by the fact that one maintenance is likely sufficient when considering short planning periods, c.f., the provided data-set.

9.7 Benchmarks

The main characteristics of the tested instances are shown in Table 9.1. They correspond to the instances of the final phase of the ROADEF Challenge 2014. All tests are run on a dedicated machine with dual 2.66 GHz Intel Xeon E5345 processors and 24 GB of memory. Both processors have 4 physical cores supporting 2 threads per core. The IBM ILOG CPLEX version 12.5 optimization software is used for solving all Linear Programs (LPs) and MIPs. All solutions are verified to be correct (with respect to feasibility and solution cost) using a modified version of the provided *solution checker* Ramond and Nicolas [2014].

Instance	Arrivals	Linked	Departures	Reuses
B1	1235	475	1235	804
B2	1235	475	1235	0
B3	1235	0	1235	0
B4	1780	722	1780	1089
B5	2153	720	2153	1089
B6	1780	722	1780	1089
B7	304	144	304	187
B8	304	144	304	187
B9	1967	860	1967	1226
B10	196	89	196	123
B11	1122	486	1122	726
B12	570	263	570	377

Table 9.1: A description of the tested instances. The columns show instance names, number of train arrivals, number of linked train arrival/departures, number of train departures and number of specified train reuses.

In the modified version all constraints unrelated to the matching have been omitted.

The timelimit set in the ROADEF Challenge 2014 was 10 minutes. Within this limit the submitted algorithms had to perform both matching and routing within the station. The provided instances cover up to 7 consecutive days of arrivals and departures. The provided timelimit for the challenge seems a bit restricting given a planning horizon of several days. Due to the difficulty of the problem we target a time limit of 10-30 minutes.

An initial benchmark shows that both solution methods are unable to solve all instances to optimality within a few hours. Table 9.2 shows the details of the instances that were solved to optimality using the Column Generation Method (CGM). A few instances run out of memory before being able to solve the root relaxation to optimality. Given a high time-limit the MIP method is able to solve the same instanced except B12. We discuss details of the column generation later in this section.

Solving the root relaxation of the instances seems to be a hard in general. Given a time-limit of 30 minutes only four instances are solved, see Table 9.3.

					Runtime in seconds			
Instance	Obj	Generated	Cons	Vars	Sub	LP	MIP	Total
B3	0	0	3 749	229 841	0.0	3.3	21.0	25.0
B7	4 400	15 629	946	19 048	33.4	28.2	20.3	83.0
B8	4 400	17 070	946	19 027	27.4	25.9	21.1	76.0
B10	2 100	3 413	620	7 050	0.7	2.4	0.2	3.0
B12	14 700	25 017	1 753	47 258	22 912.6	81.0	52.3	23 115.0

Table 9.2: Instances solved using column generation. The columns show instance name, objective of found solution, number of generated column, number of constraints and variables in final program, and finally total runtime of subproblem, LP solving, MIP solving and total. The omitted instances (B1, B2, B4, B5, B6, B9, B11) were not solved.

			Runtime(s)	
Instance	LP Relaxation	Columns	Subproblem	LP
B1	74 278	363 378	255.9	1 484.1
B2	32 596	633 486	214.9	1 532.0
B3	0	0	0.0	3.6
B4	134 582	619 406	264.2	1 485.9
B5	132 582	429 643	191.9	1 567.7
B6	133 095	652 439	270.3	1 477.5
B7	3 400	28 980	10.6	52.4
B8	3 400	30 242	12.0	55.7
B9	222 156	399 799	253.4	1 508.2
B10	1 800	6 168	1.1	3.4
B11	34 735	285 340	393.4	1 368.3
B12	13 503	71 922	1 533.2	173.6

Table 9.3: Root relaxations results given a time limit of 30 minutes. The columns show the instance name, objective of final relaxation, number of generated columns, and time spent generating columns and solving the LP relaxations. Optimal results were found for B3, B7, B8 and B10.

9.7.1 Heuristic results

Solving the problem instances using the exact methods proves to be very difficult and we therefore also investigate the performance of the methods in a heuristic context.

The next benchmark shows results obtained using CGM with a 20 minute time limit. We allocate 10 minutes for generating columns, followed by a 10 minute MIP solve (using CPLEX) of the master problem. Table 9.4 summarizes the results. The results are compared to no column generation, i.e., not allowing any linked sequences. It is observed that the result of disabling column generation seems to have a drastic effect on solution quality. This argues that ignoring linked sequences altogether is undesirable as it will drastically penalize the achievable objective. Solutions are now found for all instances in contrast to the exact approach, were only 4 instances where solved within the same timelimit. It is observed that the time is mostly spent solving LP relaxations, except for one case. Heuristic columns were used to speed up the subproblem process instead of solving the exact optimization problem in every iteration. Whenever no columns with negative reduced cost are found the method solves the exact problem. Preliminary results show that this approach is favorable as solving the exact subproblem is in many cases extremely time-consuming. Many columns are generated during the execution of CGM, and these are gradually increasing the master problem size. In future work, it might be interesting to remove unnecessary columns after a few iterations.

In our final benchmark we run the instances using the MIP Method (MIPM). As before we set a runtime limit of 20 minutes – the results are shown in Table 9.5. We note that this approach was unable to provide any solution for most instances. However, for instances B7, B8 and B10 MIPM was able to find slightly better results than CGM. A worse objective was found for the B12 instance. MIPM was unable to find solutions to all instances due to insufficient memory.

9.8 Conclusions

We have described and investigated the Departure Matching Problem which is identified as a crucial subproblem of the RSUM problem in the ROADEF/EURO

	Using Column Generation			No Column Generation	
Instance	Columns	Time	Objective	Time	Objective
B1	278 229	1 207	93 200	1 213	254 600
B2	383 937	1 200	61 000	1 201	228 000
B3	0	26	0	5	0
B4	300 143	1 162	209 300	333	379 000
B5	248 919	1 203	221 100	87	370 900
B6	285 498	1 201	211 800	341	379 000
B7	30 245	78	4 300	81	73 200
B8	27 942	86	4 400	81	73 200
B9	277 485	1 207	252 900	1 215	456 700
B10	5 908	5	2 000	26	42 400
B11	171 118	1 215	80 600	1 210	258 000
B12	71 442	1 206	14 300	1 205	135 600

Table 9.4: Solutions found using the column generation approach in 20 minutes. Results are compared to using no column generation, i.e., allowing no linked sequences.

Instance	Objective	Constraints	Variables	Time
B3	0	1 151 899	706 138	1199.4
B7	3 400	119 708	42 400	31.3
B8	3 400	119 708	42 400	31.0
B10	1 000	72 160	20 440	56.0
B12	43 900	571 526	131 360	1204.5

Table 9.5: Solutions found using the MIP approach. This approach was unable to produce any feasible solution to the omitted (B1, B2, B4, B5, B6, B9, B11) instances.

Challenge 2014. Without explicitly considering the matching problem, too many departures will be uncovered.

We prove in section 9.4 that the DMP is NP-hard in the strong sense by reduction from the 0-1 Multiple Knapsack Problem.

We have proposed two methods for solving the DMP. We first presented a pure MIP formulation of the problem which could act as a reference point for future studies. The model is however large in terms of variables and constraints and the benchmarks show that this model is unable to solve most of the proposed instances. Memory usage is one significant drawback of this method.

A second solution method based on column generation has also been presented. This model is simple and can without much difficulty be extended even further to handle more constraints. It is shown how the subproblem can be split into several independent problems thereby reducing complexity and enabling parallelism. The benchmarks for this approach show that we can find good solutions fast, if the method is used with time limits. However, even solving the root node relaxation is shown to be difficult in multiple cases. We expect that the method could be improved if embedded in a B&P framework with more efficient handing of the columns. Furthermore, the performance of this method could be further improved if a good initial solution can be provided as a *hotstart* to the CGM, e.g., the result of a heuristic method.

The considered data instances proved to be surprisingly hard to solve. The final results of the ROADEF challenge winners suggests that it is not possible to obtain a satisfactory solution to the overall problem, RSUM. There may not even exists a solution without cancellations.

For the sake of simplicity it has been assumed that all matches are possible where the departure time occurs after the arrival time. In reality it might be more realistic to remove matching options where the time between arrival and departure is either insufficient. Trains may arrival late, some time is required to perform routing inside the station and time is required to perform maintenance. Further, it may not be necessary to consider long matchings, e.g., an arrival on day 1 with a departure on day 7. Reducing the number of possible matchings will reduce the number of decision variables and constraints and likely improve the success-rate of solution methods. This may be a viable practical approach to ensure feasibility.

For future work there are a multiple matters worth considering. There are several column generation techniques that can be investigated to improve performance, e.g., dual stabilization, branch-and-price and reduced-cost fixing. The potential of heuristic methods for DMP is unknown. Such methods can potentially find good solutions fast, or even be used to speed up an exact approach. It would be interesting to consider new data instances where it is know that there exists at least one feasible solution without any cancellations.

9.9 Bibliography

Cynthia Barnhart, Ellis L Johnson, George L Nemhauser, Martin WP Savelsbergh, and Pamela H Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329. 1998.

Richard Freling, Ramon M Lentink, Leo G Kroon, and Dennis Huisman. Shunting of passenger train units in a railway station. *Transportation Science*, 39(2):261–272. 2005.

Jørgen Haahr and Simon Bull. A Math-Heuristic Framework for the ROADEF/EURO Challenge 2014. Technical report, The Technical University of Denmark. 2014.

John E Hopcroft and Richard M Karp. An n^5/2 algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231. 1973.

Stefan Irnich. Resource extension functions: properties, inversion, and generalization to segments. *OR SPECTRUM*, 30(1):113–148. 2008.

Brian Kallehauge, Jesper Larsen, Oli B.G. Madsen, and Marius M. Solomon. Vehicle routing problem with time windows. In *Column Generation*, pages 67–98. Springer US. 2005.

Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack problems*. Springer. 2004.

Leo G Kroon, Ramon M Lentink, and Alexander Schrijver. Shunting of passenger train units: an integrated approach. *Transportation Science*, 42(4):436–449. 2008.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97. 1955.

Marco E. Lübbecke and Jacques Desrosiers. Selected topics in column generation. *Operations Research*, 53(6):1007–1023. URL `http://dx.doi.org/10.1287/opre.1050.0234`. 2005.

François Ramond and Marcos Nicolas. *Trains don't vanish! ROADEF EURO 2014 Challenge Problem Description*. SNCF - Innovation & Research Department. 2014.

Part III

Appendices

Appendix A

A Math-Heuristic Framework for the ROADEF/EURO Challenge 2014

With Jørgen Haahr.[1]

[1]Technical University of Denmark. jhaa@dtu.dk

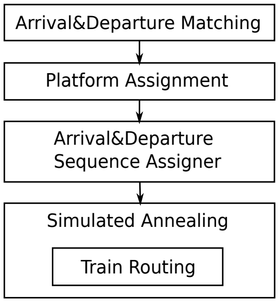


Figure A.1: An overview of the solution framework flow

A.1 Overview

The proposed solution framework can be classified as a *math heuristic* as it combines exact with heuristic methods. The heart of the solution framework is an Simulated Annealing (SA) framework that iteratively destroys and builds random train routes. However, in order to improve convergence (and runtime) a few smaller Mixed Integer Program (MIP) problems are solved in advance in order to avoid resource use conflicts and improve resource utilization. Figure A.1 illustrates the flow (and main components) of the solution framework. The full Rolling Stock Unit Management on Railway Sites (RSUM) problem is decomposed into four sequential steps which will be described in the following sections of this document. In the first step arrivals and departures are matched in order to get the best possible matching, such that e.g. the number of cancellations is minimized. Next, a platform slot is reserved for all arrivals and departures such that as many as possible are assigned to preferred platforms. Thirdly, a track group usage pattern is chosen for all arrival/departure sequences, such that no pairs of patterns are in conflict and such that no pattern is in conflict with the pre-specified imposed resource usages. Fourthly, non-overlapping facility usage slots are reserved for all maintenance activities (these are generated as a results of the matching). Finally, an SA approach iteratively removes and reroutes a group of *related* (see section A.4) trains as specified by the found matching.

A.2 Matcher

The first subproblem in the solution framework tries to match departures with compatible trains, i.e., initial trains or arrivals. The matching is formulated as a mathematical model consisting of linear constraints (and an objective) and solved using column generation, due to the large number of variables. The primary goal is to minimize the number of uncovered departures while a secondary goal is to maximize train re-uses.

The objective is formulated as a minimization of the number of unmatched departures and cost of non-satisfied train reuse :

$$\begin{aligned} \min \sum_{d \in D} & cancellationCost_d \cdot c_d \\ & + \sum_{d \in D} \sum_{t \in Comp(d)} reuseCost_t^d \cdot m_t^d \\ & + \sum_{p \in P} reuseCost_p \cdot \lambda_p \end{aligned}$$

A few heuristic artificial heuristic costs are also be added to improve the ability to perform the routing afterwards. The constraints are:

$$\sum_{t \in Comp(d)} m_t^d + \sum_{p \in P} \alpha_p^d \lambda_p \geq 1 - c_d \qquad \forall d \in D \tag{A.1}$$

$$\sum_{t \in Comp(d)} m_t^d + \sum_{p \in P} \alpha_p^d \lambda_p + \sum_{t \in Comp(d)} Block_d^t \cdot m_t^d + \sum_{p \in P} Block_p^d \cdot \lambda_p \leq 1 \qquad \forall d \in D \tag{A.2}$$

$$\sum_{d \in Comp(t)} m_t^d + \sum_{p \in P} \beta_p^t \lambda_p \leq 1 \qquad \forall t \in T \tag{A.3}$$

$$\sum_{d \in D} \sum_{t \in Comp(d)} Maint_{t,d}^{day} \cdot m_t^d + \sum_{p \in P} Maint_p^{day} \lambda_p \leq MaintLimit_{day} \qquad \forall day \in H \tag{A.4}$$

Where m_t^d is a binary variable indicating whether train t is matched to departure d. The binary variables c_d indicate whether departure d is canceled or not. The binary variables λ_p indicate whether a linked-departure pattern p is chosen. Since it is not trivial to model linked departures the m_t^d variables only indicate choices for non-linked matches while all linked arrival and departures are modeled using patterns. A pattern is the arrival/departure trajectory of a *real* train, i.e., a pattern is a sequence of matches where all (except possibly the last) have non-linked departures. A full enumeration of these patterns is intractable which is why column generation is used, the column generation process is describe in subsection A.2.1.

The cancellation cost also includes the cost of a non-satisfied reuse, if such is present. The sets D, T and H respectively represent all departures, all trains and all days of the planning horizon. The $Maint_{t,d}^{day}$ and $Maint_{p}^{day}$ coefficients denote how many maintenance operations are needed at day $day \in H$. The $Block_d^t$ and $Block_p^d$ indicate whether the assignment blocks departure d. The coefficients α_p^d and β_p^t respectively denote whether a pattern contains departure d or train t. The set $Comp(d)$ is the set of trains which are compatible with d, likewise the $Comp(t)$ is the set of departures that are compatible with train t. These two sets are generated in a preprocessing step. Constraints (A.1) ensure that every departure is assigned to some train, unless there is a cancellation. Constraints (A.2) ensure that at most one train is assigned to every departure, and also block departures for trains that assume that the departure is cancelled. Note that where t is a linked train m_t^d assumes that the linked departure d' is cancelled. Constraints (A.3) ensure that each train is assigned at most once. Finally, Constraints(A.4) ensure that the total number of maintenance operations (every day) is respected. For simplicity it is here assumed that, given a matching (t,d), the day that a maintenance operation is performed is fixed. With the additions of more variables, it is possible to make the day of maintenance operations a choice.

Instead of enumerating all possible train and departure matches the sets $Comp(d)$ and $Comp(t)$ are computed. First, any pairs with a train t arriving after the departure d are removed as such clearly cannot be matched. Second, all pairs where the train is not compatible with the departure are removed. Third, pairs are removed by inspecting the timespan between train and departure and comparing this to the minimum time required for routing and required maintenance appointments. Finally, all arrivals (or departures) are removed where there exists no feasible arrival (or departure) sequence, due to imposed resources. Since runtime is scarce a number of heuristic choices can also be employed here. In order to reduce the MIP size the solution framework furthermore removes matches longer than a certain timespan. Due to the team size and time all joint arrivals and departures are removed.

A model (solved in a full branch-and-price framework) can be used to generate a true lower bound on uncovered departures, or even a lower bound on the minimum cost. Such a lower bound could prove useful for heuristic methods.

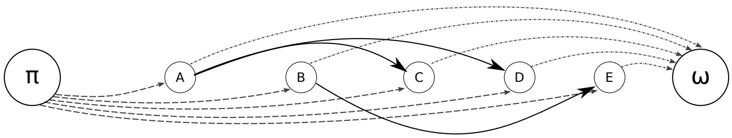


Figure A.2: An illustration of the underlying graph for the matching subproblem

A.2.1 Column generation subproblem

Column generation is a well-described technique used with success for solving MIP problems, e.g. the Vehicle Routing Problem with Time Windows (VRPTW). It is assumed that the reader is familiar with column generation solution methods. The subproblem can be solved as a Resource Constrained Shortest Path Problem (RSCPP). The underlying graph consists of one node per linked arrival and departure pair in addition to one source and one sink node, see Figure A.2. The edges constitute matching choices. Three families of edges are added. First, edges originating from the source to every node in the graph represent compatible arrivals that are matched to the linked departure of the node. Second, edges are added between nodes that represent compatible linked continuations, i.e., linked arrival/departures that connect to another linked arrival/departure. An example: In the (π, A, D, ω) path the departure of the initial train matching (represented by edge (π, A)) is linked to one arrival (node A). The departure of the next matching ((A, D)) is to another linked arrival (node D). The departure of the last matching is not linked to any arrival, and thus the sequence ends.

The problem is a RSCPP due to the maintenance constraints. In addition to the objective coefficients, the duals from Constraints (A.1) and (A.2) are added to edges of the corresponding departure, and the duals from Constraint (A.3) are added to edges of the corresponding trains. The appropriate dual from Constraints (A.4) is added every time a maintenance operation is scheduled. Labels in the Shortest Path enumeration method keep track of total cost, remaining DBM and remaining TBM. Domination is possible if a label has lower cost ($\leq$), and at least equal remaining DBM and TBM ($\geq$). Labels originate from the source vertex and are extended by traversing available arcs and deciding whether to perform maintenance (DBM or TBM or both).

A.3 Platform assigner

Platforms must be assigned to all covered arrivals and departures. Once an arrival/departure matching is known, a platform assignment can be performed. In a highly utilized network the arrivals and departures may be competing for the same platforms, which motivates an exact solution approach. In the solution framework a MIP is formulated that assigns one compatible platform to every covered arrival and departure:

$$\max -\sum_{a\in A} cancellationCost \cdot s_a - \sum_{d\in D} cancellationCost \cdot s_d + \sum_{(i,j)\in NC} c \cdot d_{i,j}$$

The constraints are:

$$\sum_{p\in Comp(a)} x_a^p \geq 1 - s_a \qquad \forall a \in A \qquad (A.5)$$

$$\sum_{p\in Comp(d)} x_d^p \geq 1 - s_d \qquad \forall d \in D \qquad (A.6)$$

$$x_i^p + x_j^p \leq 1 \qquad \forall p \in P, (i,j) \in C \qquad (A.7)$$

$$end_i + M(1 - x_i^p) + d_{i,j} \leq begin_j + M(1 - x_j^p) \qquad \forall p \in P, (i,j) \in NC \qquad (A.8)$$

$$(A.9)$$

Two sets of binary variables are used x_a^p and x_d^p respectively indicating whether arrival a or departure d is assigned to platform p. A set of continuous variables $d_{i,j}$ exists for measuring the (approximate) slack between two consecutive platform usages. The ideal objective is to maximize the smallest slack variable. However, for performance reasons the objective is changed to maximize the slack sum (i.e. average). A coefficient $c \in \mathbf{R}$ is used for scaling the distance. C is the set of all usage pairs that are overlapping in time. NC denotes all pairs of consecutive (in time) usages. Constraints (A.5)-(A.6) ensure one assignment (or cancellation). Constraints (A.7) ensures that two assignments (arrival or departure) cannot be assigned to the same platform if they overlap in time. Constraints (A.8) measure the slack between two consecutive events if they appear on the same platform. The $start_j \in \mathbf{R}$ and $end_j \in \mathbf{R}$ are determined by the minimal usage required for the corresponding arrival or departure usage.

Note that an arrival or a departure is not limited to using the assigned platform, however the found platform assignment will ensure a minimal cancellation due to lack of a available platforms.

A.4 Simulated annealing

In the final step a SA approach is used to search for a good solution. The initial solution is an empty solution and in every iteration a train is selected for routing, and a randomized path is generated for the train (based on the matching and allocated resource allocations). Before routing the neighborhood of the selected train is also removed (if the path is blocked). The neighborhood is the set of other trains in the current solution that intersect usages on the selected path. After adding the generated path for the selected train the neighborhood is re-inserted (if possible) in random order. The new solution is then accepted or rejected depending on new solution cost and the current temperature. An overview is shown in Algorithm 1.

Algorithm 1 Simulated Annealing

$current \leftarrow \texttt{GenerateEmptySolution}()$
$T \leftarrow T_{init}$
while $T_{term} < T$ **do**
    for $i \in \{0, \ldots, iterations\}$ **do**
        $t \leftarrow \texttt{RandomTrain}()$
        $p \leftarrow \texttt{RandomPath}(t)$
        $n \leftarrow \texttt{FindNeighborhood}(p)$
        $solution' \leftarrow current$
        $solution' \leftarrow \texttt{Destroy}(solution', p)$
        $solution' \leftarrow \texttt{Destroy}(solutoin', n)$
        $solution' \leftarrow \texttt{Route}(solution', p)$
        $solution' \leftarrow \texttt{Route}(solution', n)$
        $\delta \leftarrow \texttt{Cost}(solution') - \texttt{Cost}(current)$
        if $\delta < 0 \vee \texttt{Random}(0,1) < e^{\frac{-\delta}{T}}$ **then**
            $current \leftarrow solution'$
    $T \leftarrow T \cdot \alpha$

A.4.1 Router

Given a existing resources usages and a valid resource path (with unspecified entrances, exits and gates) through the infrastructure the *Router* aims to find non-conflicting usages for that path. Multiple feasible solutions may exist, but ties are broken by assigning an artificial cost to dwell times at each resource. The *Router* recursively explores all available usage windows on a single resources. If compatible windows (earliest entrance and latest exit) are found for all resources on the path then the lowest cost assignment is made (using the artificial costs). The optimal solution can be sought by continuing the search and pruning unexplored paths whenever possible.

A.5 Final remarks

The problem has proven to be very difficult. Implementing the solution framework has been time-consuming and the work is still not completed. Routing of joint arrivals and departures has been omitted due to time constraints.

Appendix B

Train Turn Restrictions and Line Plan Performance

With Sofie Burggraeve.[1]

[1]University of Leuven. sofie.burggraeve@kuleuven.be

B.1 Introduction

In this paper, we study the impact of the 'turn conditions' in end stations on the performance of a line plan. If trains have to turn on their platform in an end station, they occupy the platform for several minutes. A more preferred option, from a timetabling point of view, would be that a train disappears from the platform in its end station after dwelling and only appears again when departing for a subsequent trip. In this case, the train will not interfere with other trains that dwell on the platform during the time between these events. However, this option is only possible if the train can stay in a flexible and large enough shunt. Starting from a given line plan, we compare two timetables, one where trains have to turn on their platform and one where trains can turn in a shunt. We evaluate the impact on the performance of the line plan by its feasibility for timetabling, the minimum overall buffer time between trains, the sum of the buffer times and the buffer times in individual stations. A case study on the DSB S-tog in Copenhagen (Denmark) is performed.

B.2 State of the art

Railway network characteristics affect the performance of a railway service, for example, the use of tracks or junctions by trains in opposite directions, the number of platforms in the stations, the existence of alternative routes, the layout of the shunt in end stations, etc. In [Marín et al., 2009], an iterative approach is presented to integrate robust network design and line planning. They focus on the construction of connection links that offer an alternative in case of link failure. However, we focus on the design of the terminal stations. Line plan characteristics also influence the performance of a railway system, for example, the number of lines that use the same infrastructure, the frequency of a line and the combination of frequencies, the trip lengths (catching up due to skipping stations), etc. Goerigk et al. [2013] analyse different line planning models by comparing typical characteristics of line plans, but also by comparing their impact on timetables and their robustness against delays. Therefore, they calculate a timetable for each line plan. In this paper, we also calculate timetables in order to evaluate line plans, but we focus on the impact of an infrastructure change and we don't compare line plans mutually.

B.3 Methodology

We create a set of diverse line plans. For each of these line plans we test if it is possible to generate a feasible timetable with and without turning on the platform. The majority of the initial line plans are not timetable feasible with turning on the platform. Then, we update the line planning model by including information on why timetables are infeasible. If the line plan is timetable feasible with and without turning on the platform, we find optimal (or near optimal) solutions to both timetable problems. We then compare the performance of both. The line planning module, its updates and the timetable model are now briefly explained.

B.3.1 Line planning

Railway line planning is a long term planning problem which consists of the determination of the routes, stopping patterns, and hourly frequencies that should be operated in the railway network. We use a mixed integer linear program (MILP) to create line plans. Passengers are simultaneously routed, each picking a route with best estimated travel time, waiting time (estimated from headway times) and additional transfer penalties, but subject to capacity constraints. Operator line cost is estimated based on the rolling stock unit requirements and running times for the plan. A feasible line plan satisfies operational requirements and contractual service levels, satisfies conditions for feasible timetabling (though no all-embracing set of conditions is known), and provides capacity for all expected passengers. Line plans are either found by minimizing total passenger travel time or running cost, with a constraint on the other, or a weighted sum of each. We report average passenger travel time, which is similar across line plans as the majority of passenger demand is between stations that are well-served by all “reasonable” line plans but the differences are important as they are large differences for a minority of passengers. To find timetable feasible plans, additional constraints are introduced and tightened iteratively, concerned with the shared use of end stations. Additionally, specific line combinations meeting certain unfavourable conditions for timetabling, e.g. one line that would certainly catch up with another one, are forbidden from appearing together.

B.3.2 Timetable model

Railway timetabling is a mid-long term planning problem, which consists of the determination of train arrival and departure times in stations. We use a MILP to build a cyclic timetable with a period of one hour starting from a line plan. The trains of a line are equally spread over the period. The goal function of this timetable model maximizes the minimal buffer time between every two trains. Reserve and release times of station areas are taken into account in the calculation of the buffer times between the trains. There is only one type of constraint that is necessary to allow for feasible turnarounds in the end stations of the lines. A train reserves the platform in its end station until the next train in the opposite direction departs from that platform (because that is the same physical train). This constraint is necessary to connect the schedule of trains in opposite directions, but it shortens the buffer time between trains of the same line in the same direction.

B.4 Case study

The DSB S-tog is a high frequency railway service that transports 30 000 to 40 000 passengers per hour at peak times between 84 stations. An illustration can be found on `http://www.dsb.dk/Global/STog/S%20kort%20udk%202015%233%20(3).pdf`. This network has seven terminal stations with two platforms and eight intermediate stations in which one platform is constructed as a terminal. Every train turns on the platform in its end station and thus occupies this platform for several minutes. A train's occupation time of this platform is bounded below by the minimum time needed to turn and bounded from above by the time until the arrival of the next train. The DSB S-tog network is designed so that trains that cross each other in opposite directions (almost) only affect each other in the end station. We assume that trains occupy every station area on their trip for 60 seconds. Trains of the same line are equally spread over the period of one hour.

The results of the case study can be found in Table B.1. The first line plan was previously used by DSB S-tog. The second line plan is designed to be better for the passengers than line plan 1. The third line plan is designed to have a better operator cost than line plan 1 and the fourth line plan outperforms line plan 1 on both characteristics. The upper part of the table shows some general information on the line plans. The passenger and operator cost are estimated by the line plan model. The number of interactions are the number of train pairs that share a platform in at least one station. In the lower part, we report four indicators on the spreading of the trains in time. All the timetables are constructed with a calculation time limit of 5400 seconds. We see that the minimal overall buffer time (Min buf time) and the sum of the minimal buffer times ($\sum$ buf times) improve if turning on the platform is not required. The values of the number of stations with a buffer time bigger than two minutes or equal to the minimum buffer time, show no clear trend, but they show that the turn restrictions affect the buffer times in the whole network. Optimal values give railway companies the information on how much the performance could be increased if flexible large enough shunts would be built in the terminal stations.

Line plan	1			2		3		4	
# lines	9			8		8		7	
# interactions	1434			1362		1434		1434	
passenger cost	1172			1158		1198		1171	
line cost	679			829		653		665	
Turn restrictions	DSB	yes	no	yes	no	yes	no	yes	no
Min buf time (min)	-1 [2]	0,178	0,667	0,250	0,5	0	0,250	0,355	0,667
(upper bound)		(0,667)	(0,667)	(0,583)	(1)	(0,250)	(0,250)	(0,467)	(0,667)
#stations with min buf time	1	12	52	34	29	24	10	19	19
#stations with buf time ≥ 2	64	53	30	21	31	43	46	50	47
$\sum$ buf times (min)	18 450	18 969,6	20 196	17 985,2	19 086	19 085,8	20 166	19 180,9	20 308
(upper bound)		(21 066)	(21 786)	(19 602)	(20 298)	(20 580)	(21 306)	(20 214)	(20 946)

Table B.1: The minimal buffer times improve if the infrastructure allows to turn in a shunt.

[2] One station has overlapping occupation times. This allows for bigger buffer times in other stations.

B.5 Conclusion and future research

We analysed the effect of train turn restrictions on railway service performance. We conclude that for different kinds of line plans, the probability of propagation of delays can significantly be improved if the terminal stations have an appropriate shunt so trains don't need to turn on (and occupy) their platform. Moreover, with our approach we can calculate the magnitude of this performance improvement. Furthermore, we found that information on trip lengths and frequency combinations in end stations is useful for finding line plans that are feasible for timetabling with train turn restrictions. Future research consists of further integration of line planning and robust timetabling.

B.6 Bibliography

M. Goerigk, M. Schachtebeck, and A. Schöbel. Evaluating line concepts using travel times and robustness. *Public Transportation*, 5:267 – 284, 2013.

Á. Marín, J.A. Mesa and F. Perea. Integrating Robust Railway Network Design and Line Planning under Failures. In R. K. Ahuja, R.H. Möhring, C.D. Zaroliagis, *Robust an Online Large-Scale Optimization*, volume 5868 of Lecture Notes in Computer Science, pp. 273-292, Springer-Verlag Berlin Heidelberg, 2009.

Appendix C

Algorithm for Enumerating Many Near-Efficient Solutions

C.1 Method description

Here, we briefly discuss the method used to generate multiple solutions in Chapter 7. Recall the two objectives: one related to passenger flows and the other to operator cost.

The operator cost objective and has the form:

$$\text{minimize} \quad \sum_i c_i x_i$$

where $x_i \in \{0, 1\}$.

Algorithm 2 Enumerative solution discovery

$model \leftarrow \texttt{FormulateProblem}()$
$solutions \leftarrow \emptyset$
for $i \in \{0, \ldots, iterations\}$ **do**
    $S \leftarrow \texttt{GetEfficientSolutions}(model)$
    for $s \in S$ **do**
        $model \leftarrow \texttt{ApplySolutionCut}(model, s)$
    $solutions \leftarrow solutions \cup S$

Algorithm 2 explains the method for finding a variety of solutions. In short, we simply use a method to find efficient solutions; apply a constraint to forbid each found solution; and continue in the reduced problem. If continued with no other stop criteria, eventually every feasible solution would be enumerated. In contrast, if only one applied for one iteration then only efficient solutions would be discovered. Applying the method for a few iterations only finds solutions that are efficient and close to being efficient.

In finding the efficient solutions, we tested using a weighted sum scalarization method, and an ϵ-constraint method (see [Ehrgott, 2006]). In the case of the ϵ-constraint method, we turn the cost objective into a knapsack constraint which for some right hand side values proved to cause large degradation in solve time. Instead we primarily used a weighted sum method.

C.2 Bibliography

Matthias Ehrgott. A discussion of scalarization techniques for multiple objective integer programming. *Annals of Operations Research*, 147(1):343–360. 2006.